NAME

INSTALL – Installation procedure for NetBSD/i386.

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 1.5 on the *i386* 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 thirty-one different system architectures featuring twelve distinct families of CPUs, and is being ported to more. The NetBSD 1.5 release contains complete binary releases for fifteen different machine types. (The sixteen remaining are not fully supported at this time and are thus not part of the binary distribution. For information on them, please see the NetBSD web site at **http://www.netbsd.org/**)

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 Since The Last Release

The NetBSD 1.5 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 userland 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 over one year of development that went into the NetBSD 1.5 release. Some highlights include:

Kernel

- Ports to new platforms including: arc, cobalt, hpcmips, news68k, sgimips, and sparc64.
- Improved performance and stability of the UVM virtual memory subsystem.
- Implementation of generic kernel locking code, as well as a restructure and re-tuning of the scheduler, to be used by the future symmetric multi-processing (SMP) implementation.
- Improved compatibility support for Linux, OSF1, and SVR4 programs.

- New compatibility support for Win32 programs.
- Support for dynamically loaded ELF kernel modules.
- Kernel process tracing using `ktruss`(1).
- Deletion of swap devices using `swapctl`(8).
- Easier hot-pluggability of keyboards and mice using a new wscons device - wsmux.
- Improved PCMCIA and Cardbus support, including support for detaching of devices and cards, resulting in better support for notebooks and PDA devices.
- Numerous hardware improvements, including areas such as: audio, UDMA/66 support for ATA drives, USB, and wireless networking.

Networking

- Addition of IP version 6 (IPv6) and IPsec to the networking stack, from the KAME project. This includes addition of kernel code for IPv6/IPsec, IPv4/v6 dual-stack user applications and supporting libraries. Due to this, the shlib major version for `pcap`(3) is incremented and you may need to recompile userland tools. The KAME IPv6 part includes results from the unified-ipv6 effort.

File system

- Significant Fast file system (FFS) performance enhancements via integration of Kirk McKusick's soft updates and trickle sync code.
- Support for the Windows NT 'NTFS' file system (read-only at this stage).
- Support for revision 1 of the Linux 'ext2fs' file system.
- Enhanced stability and usability of LFS (the BSD log-structured file system).
- Various RAIDframe enhancements including: auto-detection of RAID components and auto-configuration of RAID sets, and the ability to configure the root file system (`/`) on a RAID set.
- Support for Microsoft Joliet extensions to the ISO9660 CD file system.
- Improved file system vnode locking mechanisms, thus resolving a source of several panics in the past.
- Support for NFS and RPC over IPv6.
- Server part of NFS locking (implemented by `rpc.lockd`(8)) now works.

Security

- Strong cryptographic libraries and applications integrated, including the AES cipher Rijndael, the OpenSSL library, more complete Kerberos IV and Kerberos V support, and an SSH server and client.
- `sysctl`(3) interfaces to various elements of process and system information, allowing programs such as `ps`(1), `dmesg`(1) and the like to operate without recompilation after kernel upgrades, and remove the necessity to run setgid kmem (thus improving system security).
- Disable various services by default, and set the default options for disabled daemons to a higher level of logging.
- Several code audits were performed. One audit replaced string routines that were used without bound checking, and another one to identify and disable places where format strings were used in an unsafe way, allowing arbitrary data entered by (possibly) malicious users to overwrite application code, and leading from Denial of Service attacks to compromised system.

System administration and user tools

- Conversion of the `rc`(8) system startup and shutdown scripts to an 'rc.d' mechanism, with separate control scripts for each service, and appropriate dependency ordering provided by `rcorder`(8).
- `postfix`(1) provided as alternative mail transport agent to `sendmail`(8).
- User management tools `useradd`(8), `usermod`(8), `userdel`(8), `groupadd`(8), `groupmod`(8), and `groupdel`(8) added to the system.
- Incorporation of a login class capability database (`/etc/login.conf`) from BSD/OS.
- Improved support for usernames longer than eight characters in programs such as `at`(1) and `w`(1).
- Many enhancements to `ftpd`(8) providing features found in larger and less secure FTP daemons, such as user classes, connection limits, improved support for virtual hosting, transfer statistics, transfer rate throttling, and support for various IETF ftpext working group extensions.
- The `ftp`(1) client has been improved even further, including transfer rate throttling, improved URL support, command line uploads. See the man page for details.

Miscellaneous

- Updates to the NetBSD source code style code (located in `/usr/share/misc/style`) to use ANSI C only (instead of K&R) and reflect current (best) practice, and begin migrating the NetBSD source code to follow it.
- Implementation of many SUSv2 features to the `curses`(3) library, including support for color.
- Updates of most third party packages that are shipped in the base system, including `file`(1), `ipfilter`(4), ppp(4), and `sendmail`(8) to the latest stable release.
- Many new packages in the *pkgsrc* system, including standard desktops like KDE and GNOME as well as latest Tcl/Tk and perl and many of the components of the Java Enterprise platform. The package framework itself now has full wildcard dependency support.

As has been noted, there have also been innumerable bug fixes.

Kernel interfaces have continued to be refined, and more subsystems and device drivers are shared among the different ports. You can look for this trend to continue.

This is the eighth major release of NetBSD for the i386.

As is usual between releases, the i386 port has had many improvements made to it - too many to detail all of them here.

Numerous new drivers have been added. See the supported hardware list for details.

Some (but not all!) notable i386-specific improvements include:

- Migration to the ELF binary format from a.out. You can still use your pre-1.5 a.out applications, though.
- Preliminary support for MCA (MicroChannel Architecture)

NetBSD 1.5 on i386 is, as usual, also fully backward compatible with old NetBSD/i386 binaries, so you don't need to recompile all your local programs provided you set the appropriate binary compatibility options in your kernel configuration.

The Future of NetBSD

The NetBSD Foundation has been incorporated as a non-profit organization. Its purpose is to encourage, foster and promote the free exchange of computer software, namely the NetBSD Operating System. The founda-

tion will allow for many things to be handled more smoothly than could be done with our previous informal organization. In particular, it provides the framework to deal with other parties that wish to become involved in the NetBSD Project.

The NetBSD Foundation will help improve the quality of NetBSD by:

- providing better organization to keep track of development efforts, including co-ordination with groups working in related fields.
- providing a framework to receive donations of goods and services and to own the resources necessary to run the NetBSD Project.
- providing a better position from which to undertake promotional activities.
- periodically organizing workshops for developers and other interested people to discuss ongoing work.

We intend to begin narrowing the time delay between releases. Our ambition is to provide a full release every six to eight months.

We hope to support even *more* hardware in the future, and we have a rather large number of other ideas about what can be done to improve NetBSD.

We intend to continue our current practice of making the NetBSD-current development source available on a daily basis.

We intend to integrate free, positive changes from whatever sources submit them, providing that they are well thought-out and increase the usability of the system.

Above all, we hope to create a stable and accessible system, and to be responsive to the needs and desires of NetBSD users, because it is for and because of them that NetBSD exists.

Sources of NetBSD

Refer to **http://www.netbsd.org/Sites/net.html**.

NetBSD 1.5 Release Contents

The root directory of the NetBSD 1.5 release is organized as follows:

`.../NetBSD-1.5/`

`CHANGES` Changes since earlier NetBSD releases.

`LAST_MINUTE` Last minute changes.

`MIRRORS` A list of sites that mirror the NetBSD 1.5 distribution.

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

`TODO` NetBSD 's todo list (also somewhat incomplete and out of date).

`patches/` Post-release source code patches.

`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 1.5 has a binary distribution. There are also `README.export-control` files sprinkled liberally throughout the distribution tree, which point out that there are some portions of the distribution that may be subject to export regulations of the United States, e.g. code under `src/crypto` and `src/sys/crypto`. It is your responsibility to determine whether or not it is legal for you to export these portions and to act accordingly.

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.
22.3 MB gzipped, 98.8 MB uncompressed

pkgsrc This set contains the ''pkgsrc'' sources, which contain the infrastructure to build third-party packages.
5.6 MB gzipped, 57.0 MB uncompressed

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.
3.3 MB gzipped, 13.2 MB uncompressed

src This set contains all of the base NetBSD 1.5 sources which are not in **gnusrc**, **sharesrc**, or **syssrc**.
24.2 MB gzipped, 120.6 MB uncompressed

syssrc This set contains the sources to the NetBSD 1.5 kernel for all architectures, `config`(8), and `dbsym`(8).
17.6 MB gzipped, 88.6 MB uncompressed

xsrc This set contains the sources to the X Window System.
35.2 MB gzipped, 176.8 MB uncompressed

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. They may be unpacked into `/usr/src` with the command:

```
# ( cd / ; tar -zxpf - ) < set_name.tgz
```

The `sets/Split/` subdirectory contains split versions of the source sets for those users who need to load the source sets from floppy or otherwise need a split distribution. The split sets are named `set_name.`*xx* where `set_name` is the distribution set name, and *xx* is the sequence number of the file, starting with ''aa'' for the first file in the distribution set, then ''ab'' for the next, and so on. All of these files except the last one of each set should be exactly 240,640 bytes long. (The last file is just long enough to contain the remainder of the data for that distribution set.)

The split distributions may be reassembled and extracted with **cat** as follows:

```
# cat set_name.?? | ( cd / ; tar -zxpf - )
```

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

`BSDSUM` Historic BSD checksums for the various files in that directory, in the format produced by the command: **cksum -o 1** *file*

`CKSUM` POSIX checksums for the various files in that directory, in the format produced by the command: **cksum** *file*.

`MD5` MD5 digests for the various files in that directory, in the format produced by the command: **cksum -m** *file*.

`SYSVSUM` Historic AT&T System V UNIX checksums for the various files in that directory, in the format produced by the command: **cksum -o -2** *file*.

The MD5 digest is the safest checksum, followed by the POSIX checksum. The other two checksums are provided only to ensure that the widest possible range of system can check the integrity of the release files.

NetBSD/i386 subdirectory structure

The i386-specific portion of the NetBSD 1.5 release is found in the `i386` subdirectory of the distribution:

`.../NetBSD-1.5/i386/`

`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.
		`netbsd.GENERIC_LAPTOP.gz`	A version of `GENERIC` that has USB, PCMCIA and Cardbus enabled to allow installing on Laptop machine.
		`netbsd.GENERIC_TINY.gz`	A version of `GENERIC` intended to run on machines with less than 8 MB.
		`netbsd.INSTALL.gz`	The install kernel.
		`netbsd.INSTALL_TINY.gz`	A version of `INSTALL` intended to run on machines with less than 8 MB.
		`netbsd.INSTALL_SMALL.gz`	A version of `INSTALL` intended to fit on a 5.25"/1.2 MB diskette.
		`netbsd.INSTALL_LAPTOP.gz`	A version of `INSTALL` that has USB, PCMCIA and Cardbus enabled to allow installing on Laptop machine.
	`sets/`	i386 binary distribution sets; see below.	
`installation/`			
	`floppy/`	i386 boot and installation floppies; see below.	
	`misc/`	Miscellaneous i386 installation utilities; see installation section, below.	

Binary distribution sets

The NetBSD i386 binary distribution sets contain the binaries which comprise the NetBSD 1.5 release for the i386. There are eight binary distribution sets. The binary distribution sets can be found in the `i386/binary/sets` subdirectory of the NetBSD 1.5 distribution tree, and are as follows:

base The NetBSD 1.5 i386 **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. It includes shared library support, and excludes everything described below.
16.4 MB gzipped, 37.9 MB uncompressed

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.
12.4 MB gzipped, 44.8 MB uncompressed

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. (If you are upgrading, it's recommended that you get a copy of this set and *carefully* upgrade your configuration files by hand.)
0.1 MB gzipped, 0.6 MB uncompressed

games This set includes the games and their manual pages.
2.9 MB gzipped, 7.0 MB uncompressed

kern This set contains a NetBSD/i386 1.5 `GENERIC` kernel, named `/netbsd`. You *must* install this distribution set.
2.2 MB gzipped, 4.7 MB uncompressed

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.
5.0 MB gzipped, 19.9 MB uncompressed

misc This set includes the (rather large) system dictionaries, the typesettable document set, and other files from `/usr/share`.
2.6 MB gzipped, 10.1 MB uncompressed

text This set includes NetBSD 's text processing tools, including `groff`(1), all related programs, and their manual pages.
1.2 MB gzipped, 4.2 MB uncompressed

NetBSD maintains its own set of sources for the X Window System in order to assure tight integration and compatibility. These sources are based on XFree86, and tightly track XFree86 releases. They are currently equivalent to XFree86 3.3.6. Binary sets for the X Window system are distributed with NetBSD. The sets are:

xbase The basic files needed for a complete X client environment. This does not include the X servers.
2.7 MB gzipped, 7.9 MB uncompressed

xcomp The extra libraries and include files needed to compile X source code.
1.8 MB gzipped, 7.9 MB uncompressed

xcontrib Programs that were contributed to X.
0.2 MB gzipped, 0.5 MB uncompressed

xfont Fonts needed by X.
6.2 MB gzipped, 7.5 MB uncompressed

xserver All XFree86 X servers. Because all of them are included, this set is large. However, you will only need one of the servers provided in this set. (Typically, **`XF86_SVGA`**.)
14.9 MB gzipped, 35.2 MB uncompressed

The i386 binary distribution sets are distributed as gzipped tar files named with the extension **.tgz**, e.g. `base.tgz`. They are also available in split form – catted together, the members of a split set form a gzipped tar file.

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 files are `/`-relative and therefore are extracted *below* the current directory. That is, if you want to extract the binaries into your system, i.e. replace the system binaries with them, you have to run the **`tar -xpf`** command from `/`.

Note: Each directory in the i386 binary distribution also has its own checksum files, just as the source distribution does.

NetBSD/i386 System Requirements and Supported Devices

NetBSD 1.5 runs on ISA (AT-Bus), EISA, PCI, and VL-bus systems with 386-family processors, with or without math coprocessors. Support for MCA systems (such as some IBM PS/2 systems) is present, but still very experimental and needs special setup. The minimal configuration is said to require 4 MB of RAM and 50 MB of disk space, though we do not know of anyone running with a system quite this minimal today. To install the entire system requires much more disk space (the unpacked binary distribution, without sources, requires at least 65 MB without counting space needed for swap space, etc), and to run X or compile the system, more RAM is recommended. (4 MB of RAM will actually allow you to run X and/or compile, but it won't be speedy. Note that until you have around 16 MB of RAM, getting more RAM is more important than getting a faster CPU.)

Supported devices

- Floppy controllers.
- MFM, ESDI, IDE, and RLL hard disk controllers.
  There is complete support (including IDE DMA or Ultra-DMA) for the following PCI controllers
  - Acer labs M5229 IDE Controller
  - CMD Tech PCI0643, 0646, 0648 and 0649 IDE Controllers
  - Contaq Microsystems/Cypress CY82C693 IDE Controller
  - HighPoint HPT366 and HPT370 (in Ultra/66 mode only)
  - Intel PIIX, PIIX3 and PIIX4 IDE Controllers
  - Intel 82801 (ICH/ICH0) IDE Controllers
  - Promise PDC20246 (Ultra/33), PDC20262 (Ultra/66) and Ultra/100 (in Ultra/66 mode only)
  - Silicon Integrated System 5597/5598 IDE controller
  - VIA Technologies VT82C586 and VT82C586A IDE Controllers

  Most of these controllers are only available in multifunction PCI chips. Other PCI IDE controllers are supported, but performance may not be optimal. ISA, ISA plug and play and PCMCIA IDE controllers are supported as well.
- SCSI host adapters
  - Adaptec AHA-154xA, -B, -C, and -CF
  - Adaptec AHA-1640 cards (MCA variant of AHA-1540) [m]
  - Adaptec AHA-174x
  - Adaptec AIC-6260 and AIC-6360 based boards, including the Adaptec AHA-152x, Adaptec APA-1460 (PCMCIA) and APA-1480 (CardBus), and the SoundBlaster SCSI host adapter.

    Note: You cannot boot from these boards if they do not have a boot ROM; consequently only the AHA-152x and motherboards using this chip are likely to be bootable.
  - Adaptec AHA-2910, 2915, 2920, and 2930C adapters.
  - Adaptec AHA-2x4x[U][2][W] cards and onboard PCI designs using the AIC-7770, AIC-7850, AIC-7860, AIC-7870, AIC-7880 and AIC-789x chipsets.
  - Adaptec AHA-394x[U][W] cards [b]
  - Adaptec AHA-3950U2 cards
  - Adaptec AHA-3960, 19160 and 29160 Ultra-160 adapters
  - AdvanSys ABP-9x0[U][A] cards
  - AdvanSys ABP-940UW[68], ABP-970UW[68], ASB3940UW-00 cards

- AMD PCscsi-PCI (Am53c974) based SCSI adapters, including Tekram DC-390
- BusLogic 54x (Adaptec AHA-154x clones)
- BusLogic 445, 74x, 9xx (But not the new 'FlashPoint' series of BusLogic SCSI adapters)
- Qlogic ISP [12]0x0 SCSI/FibreChannel boards
- Seagate/Future Domain ISA SCSI adapter cards
  - ST01/02
  - Future Domain TMC-885
  - Future Domain TMC-950
- Symbios Logic (NCR) 53C8xx-based PCI SCSI host adapters
  - Acculogic PCIpport
  - ASUS SC-200 (requires NCR BIOS on motherboard to boot from disks)
  - ASUS SC-875
  - ASUS SP3[G] motherboard onboard SCSI
  - DEC Celebris XL/590 onboard SCSI
  - Diamond FirePort 40
  - Lomas Data SCSI adapters
  - NCR/SYM 8125 (and its many clones; be careful, some of these cards have a jumper to set the PCI interrupt; leave it on INT A!)
  - Promise DC540 (a particularly common OEM model of the SYM 8125)
  - Tekram DC-390U/F
  - Tyan Yorktown
- Symbios Logic (NCR) 5380/53C400-based ISA SCSI host adapters [*]
- Ultrastor 14f, 34f, and (possibly) 24f
- Western Digital WD7000 SCSI and TMC-7000 host adapters (ISA cards only)

- MDA, CGA, VGA, SVGA, and HGC Display Adapters.

  Note: Not all of the display adapters NetBSD/i386 can work with are supported by X. See the XFree86 FAQ for more information. [m]

- Serial ports
  - 8250/16450-based ports [m]
  - 16550/16650/16750-based ports [m]
  - AST-style 4-port serial cards [*]
  - BOCA 8-port serial cards [*]
  - BOCA 6-port (ioat) serial cards [*]
  - IBM PC-RT 4-port serial cards [*]
  - Single-port Hayes ESP serial cards [*]
  - Cyclades Cyclom-Y serial cards [*] [+]
  - Addonics FlexPort 8S 8-port serial cards [*]
  - PCI universal communication cards

- Parallel ports. [*] [+] [m]

- Ethernet adapters
  - AMD LANCE and PCnet-based ISA Ethernet adapters [*]
    - Novell NE1500T
    - Novell NE2100
    - Kingston 21xx
    - Digital EtherWORKS II ISA adapters (DE200/DE201/DE202)
  - AMD PCnet-based PCI Ethernet adapters
    - Addtron AE-350

    - BOCALANcard/PCI
    - SVEC FD0455
    - X/Lan Add-On Adapter
    - IBM #13H9237 PCI Ethernet Adapter
- AT&T StarLAN 10, EN100, and StarLAN Fiber
- 3COM 3c501
- 3COM 3c503
- 3COM 3c505 [*]
- 3COM 3c507
- 3COM 3c509, 3c579, 3c589, and 3c59X
- 3COM 3c90X (including 3c905B)
- Digital DC21x4x-based PCI Ethernet adapters
    - ASUS PCI-DEC100TX+
    - Cogent EM1X0, EM960 (a.k.a. Adaptec ANA-69XX)
    - Cogent EM964 [b]
    - Cogent EM4XX [b]
    - Compex Readylink PCI
    - DANPEX EN-9400P3
    - Digital Celebris GL, GLST on-board ethernet
    - DEC (Digital) PCI Ethernet/Fast Ethernet adapters (all)
    - DLINK DFE500-TX
    - JCIS Condor JC1260
    - Linksys PCI Fast Ethernet
    - SMC EtherPower 10, 10/100 (PCI only!)
    - SMC EtherPower^2 [b]
    - SVEC PN0455
    - SVEC FD1000-TP
    - Znyx ZX34X
- Digital EtherWORKS III ISA adapters (DE203/DE204/DE205) [*]
- Digital DEPCM-BA (PCMCIA) and DE305 (ISA) NE2000-compat. cards
- BICC Isolan [* and not recently tested]
- Efficient Networks EN-155 and Adaptec AIC-590x ATM interfaces
- Essential Communications Hippi (800 Mbit/s)
- Fujitsu MB86960A/MB86965A based cards
    - Fujitsu FMV-180 series
    - Allied-Telesis AT1700 series
    - Allied-Telesis RE2000 series
- Intel EtherExpress 16
- Intel EtherExpress PRO/10
- Intel EtherExpress 100 Fast Ethernet adapters
- Novell NE1000, NE2000 (ISA, PCI, PCMCIA, ISA PnP)
- RealTek 8129/8139 based boards
- SMC/WD 8003, 8013, and the SMC 'Elite16' ISA boards
- SMC/WD 8216 (the SMC 'Elite16 Ultra' ISA boards)
- SMC 91C9x-based boards (ISA and PCMCIA)
- SMC EPIC/100 Fast Ethernet boards
    - SMC Etherpower-II
- Texas Instruments ThunderLAN based ethernet boards
    - Compaq Netelligent 10/100 TX

    - Compaq ProLiant Integrated Netelligent 10/100 TX
    - Compaq Netelligent 10 T (untested)
    - Compaq Integrated NetFlex 3/P
    - Compaq NetFlex 3/P in baseboard variant (the PCI variant doesn't use the same chip !)
    - Compaq Dual Port Netelligent 10/100 TX
    - Compaq Deskpro 4000 5233MMX (untested)
    - Texas Instruments TravelMate 5000 series laptop docking station Ethernet board
  - VIA VT3043(Rhine) and VT86C100A(Rhine-II) based ethernet boards
    - D-Link DFE530TX

- FDDI adapters
  - Digital DEFPA PCI FDDI adapters [*] [+]
  - Digital DEFEA EISA FDDI adapters [*] [+]

- Token-Ring adapters
  - IBM Token-Ring Network PC Adapter [+]
  - IBM Token-Ring Network PC Adapter II [+]
  - IBM Token-Ring Network Adapter/A [+]
  - IBM Token-Ring Network 16/4 Adapter [+]
  - IBM Token-Ring Network 16/4 Adapter/A [+] [m]
  - IBM 16/4 ISA Adapter [+]
  - IBM Auto 16/4 Token-Ring ISA Adapter [+]
  - 3COM 3C619 TokenLink [+]
  - 3COM 3C319 TokenLink Velocity [+]

- Wireless network adapters
  - AT&T/Lucent WaveLan IEEE (802.11) PCMCIA cards
  - BayStack 650 802.11FH PCMIA cards [*] [+]
  - NetWave AirSurfer PCMCIA cards [*] [+]
  - Melco AIR CONNECT WLI-PCM-L11 cards [*] [+]
  - DEC/Cabletron RoamAbout 802.11 DS High Rate cards [*] [+]
  - Corega Wireless LAN PCC-11 cards [*] [+]
  - ELSA AirLancer MC-11 card [*] [+]

- High Speed Serial
  - LAN Media Corporation SSI/LMC10000 (up to 10 Mbps) [*] [+]
  - LAN Media Corporation HSSI/LMC5200 [*] [+]
  - LAN Media Corporation DS3/LMC5245 [*] [+]

- Tape drives
  - Most SCSI tape drives
  - QIC-02 and QIC-36 format (Archive- and Wangtek- compatible) tape drives [*] [+]

- CD-ROM drives
  - Non-IDE Mitsumi CD-ROM drives [*] [+]

    Note: The Mitsumi driver device probe is known to cause trouble with several devices!

  - Most SCSI CD-ROM drives
  - Most ATAPI CD-ROM drives.

    Note: Some low-priced IDE CD-ROM drives are known for being not or not fully ATAPI compliant, and thus require some hack (generally an entry to a quirk table) to work with NetBSD.

- Mice
  - ''Logitech'' -style bus mice [*] [+]
  - Microsoft-style bus mice [*] [+]
  - ''PS/2'' -style mice [*] [+] [m]
  - Serial mice (no kernel support necessary)
- Sound Cards
  - SoundBlaster, SoundBlaster Pro, SoundBlaster 16 [*] [+]
  - Gravis Ultrasound and Ultrasound Max [*] [+]
  - Windows Sound System [*] [+]

  The following drivers have not been extensively tested
  - Personal Sound System [*] [+]
  - ProAudio Spectrum [*] [+]
  - Gravis Ultrasound Plug&Play [*] [+]
  - Ensoniq AudioPCI [*] [+]
  - Yamaha OPL3-SA3 [*] [+]
  - Aria based sound cards [*]
  - S3 SonicVibes [*] [+]
  - ESS Technology ES1777/1868/1869/1887/1888/888 and Solo-1 ES1938/1946 audio [*] [+]
- Game Ports (Joysticks) [*] [+]
- Miscellaneous
  - Advanced power management (APM) [*]
- Universal Serial Bus (USB)
  - UHCI host controllers [*] [+]
  - OHCI host controllers [*] [+]
  - Hubs [*] [+]
  - Keyboards using the boot protocol [*] [+]
  - Mice [*] [+]
  - Printers [*] [+]
  - Modems using Abstract Control Model [*] [+]
  - Generic support for HID devices [*] [+]
  - Ethernet adapters [*] [+]
  - Audio devices [*] [+]
  - driver for FTDI based serial adapters [*] [+]
  - Mass storage devices such as disks, ZIP drives and digital cameras [*] [+]
  - driver for the Prolific host-to-host adapter [*] [+]
  - Handspring Visor driver [*] [+]
- PCMCIA Controllers.
  ISA, PCI, and ISA Plug&Play attachments
  - Intel 82365 PCIC, rev 0 and rev 1
  - Cirrus PD6710
  - Cirrus PD672X

  Note: This will work with most laptops as well as with ISA cards which provide PCMCIA slots for desktops.
- RAID Controllers
  - 3ware Escalade family of controllers

- Compaq Integrated Array (PCI) [b]
- Compaq IAES (EISA)
- Compaq IDA, IDA-2 (EISA)
- Compaq Smart Array 221, 3100ES, 3200, 4200, 4250ES (PCI) [b]
- Compaq Smart Array 431, RAID LC2 [b]
- Compaq SMART 2, 2/E (EISA)
- Compaq SMART 2/E, 2/P, 2DH, 2SL (PCI) [b]
- DPT SCSI RAID boards (ISA [*], EISA and PCI)
  - SmartCache III
  - SmartCache IV
  - SmartRAID III
  - SmartRAID IV

Specific driver footnotes:

[*] Drivers are *not* present in kernels on the distribution floppies. Except as noted above, all drivers are present on all disks. Also, at the present time, the distributed kernels support only one SCSI host adapter per machine. NetBSD normally allows more, though, so if you have more than one, you can use all of them by compiling a custom kernel once NetBSD is installed.

[+] Support *is* included in the `GENERIC` kernels, although it is not in the kernels which are on the distribution floppies.

[b] Devices require BIOS support for PCI-PCI bridging on your motherboard. Most reasonably modern Pentium motherboards have this support, or can acquire it via a BIOS upgrade.

[m] Devices are also supported by MCA-enabled kernels, such as one compiled from the `PS2` configuration file. Support for MCA is *not* in the `GENERIC` or installation kernels at this moment.

Hardware the we do *not* currently support, but get many questions about:

- Multiprocessor systems. NetBSD will run, but only use one processor.
- PCI WD-7000 SCSI host adapters.
- QIC-40 and QIC-80 tape drives. (Drives that connect to the floppy disk controller.)

We are planning future support for many of these devices.

To be detected by the distributed kernels, the devices must be configured as follows:

```
Device          Name    Port    IRQ     DRQ     Misc
------          ----    ----    ---     ---     ----
Serial ports    com0    0x3f8   4               [8250/16450/16550/clones]
                com1    0x2f8   3               [8250/16450/16550/clones]
                com2    0x3e8   5               [8250/16450/16550/clones]

Parallel ports lpt0     0x378   7               [interrupt-driven or polling]
                lpt1    0x278                   [polling only]
                lpt2    0x3bc                   [polling only]

Floppy controller
                fdc0    0x3f0   6       2       [supports two disks]

AHA-154x, AHA-174x (in compatibility mode), or BT-54x SCSI host adapters
                aha0    0x330   any     any
                aha1    0x334   any     any
```

```
AHA-174x SCSI host adapters (in enhanced mode)
               ahb0    any     any     any

AHA-152x, AIC-6260- or AIC-6360-based SCSI host adapters
               aic0    0x340   11      6

AHA-2X4X or AIC-7xxx-based SCSI host adapters       [precise list: see NetBSD
               ahc0    any     any     any      System Requirements and
                                                 Supported Devices]

AdvanSys ABP-9x0[U][A] SCSI host adapters
               adv0    any     any     any

AdvanSys ABP-940UW[68], ABP-970UW[68], ASB3940UW-00 SCSI host adapters
               adw0    any     any     any

AMD PCscsi-PCI based SCSI host adapters
               pcscp0  any     any     any

BusLogic BT445, BT74x, or BT9xx SCSI host adapters
               bha0    0x330   any     any
               bha1    0x334   any     any

Seagate/Future Domain SCSI
               sea0    any     5       any     iomem 0xd8000

Symbios Logic/NCR 53C8xx based PCI SCSI host adapters
               ncr0    any     any     any

Ultrastor 14f, 24f (if it works), or 34f SCSI host adapters
               uha0    0x330   any     any
               uha1    0x340   any     any

Western Digital WD7000 based ISA SCSI host adapters
               wds0    0x350   15      6
               wds1    0x358   11      5

PCI IDE hard disk controllers
               pciide0 any     any     any     [supports four devices]

MFM/ESDI/IDE/RLL hard disk controllers
               wdc0    0x1f0   14              [supports two devices]
               wdc1    0x170   15              [supports two devices]

ATA disks      wd0, wd1, ...
SCSI and ATAPI disks sd0, sd1, ...
SCSI tapes     st0, st1, ...
SCSI and ATAPI CD-ROMs cd0, cd1, ...
       For each SCSI and IDE controller found, the SCSI or ATA(PI) devices
       present on the bus are probed in increasing ID order for SCSI and
       master/slave order for ATA(PI). So the first SCSI drive found will
       be called sd0, the second sd1, and so on ...
```

```
StarLAN cards
               ai0     0x360   7       any     iomem 0xd0000

FMV-180 series cards
               fmv0    0x2a0   any

AT1700 cards
               ate0    0x2a0   any

Intel EtherExpress/16 cards
               ix0     0x300   10

Intel EtherExpress PRO 10 ISA cards
               iy0     0x360   any

CS8900 Ethernet cards
               cs0     0x300   any     any

3Com 3c501 Ethernet cards
               el0     0x300   9

3Com 3c503 Ethernet cards
               ec0     0x250   9               iomem 0xd8000

3Com 3c505 Ethernet cards
               eg0     0x280   9

3Com 3c507 Ethernet cards
               ef0     0x360   7               iomem 0xd0000

Novell NE1000, or NE2000 Ethernet boards
               ne0     0x280   9
               ne1     0x300   10

Novell NE2100 Ethernet boards
               ne2     0x320   9       7

BICC IsoLan cards
               ne3     0x320   10      7

SMC/WD 8003, 8013, Elite16, and Elite16 Ultra Ethernet boards
               we0     0x280   9               iomem 0xd0000
               we1     0x300   10              iomem 0xcc000

3COM 3c509 or 3COM 3c579 Ethernet boards
               ep0     any     any

3COM 3x59X PCI Ethernet boards
               ep0     any     any     [you must assign an interrupt in your
                                        PCI BIOS, or let it do so for you]
```

```
3COM 3x90X PCI Ethernet boards
                ex0     any     any     [you must assign an interrupt in your
                                         PCI BIOS, or let it do so for you]

Intel EtherExpress PRO 10 ISA
                iy0             0x360   any

Intel EtherExpress 100 Fast Ethernet adapters
                fxp0    any     any     [you must assign an interrupt in your
                                         PCI BIOS, or let it do so for you]

SMC91C9x based Ethernet cards
                sm0     0x300   10

PCnet-PCI based Ethernet boards; see above for partial list
                le0     any     any     [you must assign an interrupt in your
                                         PCI BIOS, or let it do so for you]

DC21x4x based Ethernet boards; see above for partial list
                de0     any     any     [you must assign an interrupt in your
                                         PCI BIOS, or let it do so for you]

Digital EtherWORKS III (DE203/DE204/DE205) LEMAC
                lc0     0x320   any

Qlogic ISP [12]0x0 SCSI/FibreChannel boards
                isp0    any     any

Efficient Networks EN-155 and Adaptec AIC-590x ATM interfaces
                en0     any     any

SMC EPIC/100 Fast Ethernet boards
                epic0   any     any

Texas Instruments ThunderLAN based ethernet boards
                tl0     any     any

VIA VT3043(Rhine) and VT86C100A(Rhine-II) based ethernet boards
                vr0     any     any

IBM TROPIC based Token-Ring cards
                tr0     0xa20   any             iomem 0xd8000
                tr1     0xa24   any             iomem 0xd0000
                tr2     any     any
```

Getting the NetBSD System on to Useful Media

If you are not booting off a CD-ROM, you will need to have some floppy disks to boot off; either two 1.44 MB floppies or one 1.2 MB floppy.

For laptops that have cardbus slots, you should use the `bootlap1.fs` and `bootlap2.fs` floppy images.

For older machines with little RAM, use `boot-tiny.fs`. This image is tailored towards old, small-memory systems, and thus does not contain any PCI or SCSI support. It should work on systems with 4M of RAM. Note that this means 4M available to NetBSD; systems that are said to have 4M may have 640k of base memory and 3072k of extended memory, which currently will not work, as this is a total of 3712k.

For old machines that may have EISA, SCSI and more RAM, but only have an 1.2M floppy drive, use `boot-small.fs`. For all other systems, use `boot1.fs` and `boot2.fs`

For the 2-floppy sets (and the CD boot image), utilities to repair a badly crashed systems are included. The -small and -tiny images have seperate rescue floppy images because of lack of space.

If you are using a UNIX-like system to write the floppy images to disks, you should use the `dd` command to copy the file system image(s) (.fs file) directly to the raw floppy disk. It is suggested that you read the `dd`(1) manual page or ask your system administrator to determine the correct set of arguments to use; it will be slightly different from system to system, and a comprehensive list of the possibilities is beyond the scope of this document.

If you are using MS-DOS to write the floppy image(s) to floppy disk, you should use the **`rawrite`** utility, provided in the `i386/utilities` directory of the NetBSD distribution. It will write a file system image (.fs file) to a floppy disk. A **`rawrite32`** is also available that runs under MS Windows.

Installation is supported from several media types, including:

- CD-ROM
- MS-DOS floppy
- FTP
- Remote NFS partition
- Tape
- Existing NetBSD partitions, if performing an upgrade

The steps necessary to prepare the distribution sets for installation depend upon which installation medium you choose. The steps for the various media are outlined below.

CD-ROM

Find out where the distribution set files are on the CD-ROM.

Proceed to the instruction on installation.

MS-DOS floppy

Count the number of `set_name.`*`xx`* files that make up the distribution sets you want to install or upgrade. You will need one fifth that number of 1.2 MB floppies, or one sixth that number of 1.44 MB floppies. You should only use one size of floppy for the install or upgrade procedure; you can't use some 1.2 MB floppies and some 1.44 MB floppies.

Format all of the floppies with MS-DOS. Do *not* make any of them bootable MS-DOS floppies, i.e. don't use `format /s` to format them. (If the floppies are bootable, then the MS-DOS system files that make them bootable will take up some space, and you won't be able to fit the distribution set parts on the disks.) If you're using floppies that are formatted for MS-DOS by their manufacturers, they probably aren't bootable, and you can use them out of the box.

Place all of the `set_name.`*xx* files on the MS-DOS disks.

Once you have the files on MS-DOS disks, you can proceed to the next step in the installation or upgrade process. If you're installing NetBSD from scratch, go to the section on preparing your hard disk, below. If you're upgrading an existing installation, go directly to the section on upgrading.

FTP

The preparations for this installation/upgrade method are easy; all you need to do is make sure that there's an 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. If you don't have access to a functioning nameserver during installation, the IP address of **ftp.netbsd.org** is `204.152.184.75` (as of October, 2000).

Once you have this information, you can proceed to the next step in the installation or upgrade process. If you're installing NetBSD from scratch, go to the section on preparing your hard disk, below. If you're upgrading an existing installation, go directly to the section on upgrading.

Note: This method of installation is recommended only for those already familiar with using BSD network configuration and management commands. If you aren't, this documentation should help, but is not intended to be all-encompassing.

NFS

Place the NetBSD distribution sets you wish to install into a directory on an NFS server, and make that directory mountable by the machine on which you are installing or upgrading NetBSD. This will probably require modifying the `/etc/exports` file on of the NFS server and resetting its mount daemon (mountd). (Both of these actions will probably require superuser privileges on the server.)

You need to know the numeric IP address of the NFS server, and, if the server is 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.

Once the NFS server is set up properly and you have the information mentioned above, you can proceed to the next step in the installation or upgrade process. If you're installing NetBSD from scratch, go to the section on preparing your hard disk, below. If you're upgrading an existing installation, go directly to the section on upgrading.

Note: This method of installation is recommended only for those already familiar with using BSD network configuration and management commands. If you aren't, this documentation should help, but is not intended to be all-encompassing.

Upgrade

If you are upgrading NetBSD, you also have the option of installing NetBSD by putting the new distribution sets somewhere in your existing file system, and using them from there. To do that, you must do the following:

Place the distribution sets you wish to upgrade somewhere in your current file system tree. Please note that the `/dev` on the floppy used for upgrades only knows about `wd0`, `wd1`, `sd0`, `sd1`, and `sd2`. If you have more than two IDE drives or more than three SCSI drives, you should take care not to place the sets on the high numbered drives.

At a bare minimum, you must upgrade the **base** and **kern** binary distributions, and so must put the **base** and **kern** sets somewhere in your file system. If you wish, you can do the other sets, as well, but you should *not* upgrade the **etc** distribution; it contains contains system configuration files that you should review and update by hand.

Once you have done this, you can proceed to the next step in the upgrade process, actually upgrading your system.

Preparing your System for NetBSD installation

First and foremost, before beginning the installation process, *make sure you have a reliable backup* of any data on your hard disk that you wish to keep. Mistakes in partitioning your hard disk may lead to data loss.

Before you begin, you should be aware of the geometry issues that may arise in relation to your hard disk. First of all, you should know about sector size. You can count on this to be 512 bytes; other sizes are rare (and currently not supported). Of particular interest are the number of sectors per track, the number of tracks per cylinder (also known as the number of heads), and the number of cylinders. Together they describe the disk geometry.

The BIOS has a limit of 1024 cylinders and 63 sectors per track for doing BIOS I/O. This is because of the old programming interface to the BIOS that restricts these values. Most of the big disks currently being used have more than 1024 real cylinders. Some have more than 63 sectors per track. Therefore, the BIOS can be instructed to use a fake geometry that accesses most of the disk and the fake geometry has less than or equal to 1024 cylinders and less than or equal to 63 sectors. This is possible because the disks can be addressed in a way that is not restricted to these values, and the BIOS can internally perform a translation. This can be activated in most modern BIOSes by using *Large* or *LBA* mode for the disk.

NetBSD does not have the mentioned limitations with regard to the geometry. However, since the BIOS has to be used during startup, it is important to know about the geometry the BIOS uses. The NetBSD kernel should be on a part of the disk where it can be loaded using the BIOS, within the limitations of the BIOS geometry. The install program will check this for you, and will give you a chance to correct this if this is not the case.

If you have not yet installed any other systems on the hard disk that you plan to install NetBSD on, or if you plan to use the disk entirely for NetBSD, you may wish to check your BIOS settings for the 'Large' or 'LBA' modes, and activate them for the hard disk in question. While they are not needed by NetBSD as such, doing so will remove the limitations mentioned above, and will avoid hassle should you wish to share the disk with other systems. Do *not* change these settings if you already have data on the disk that you want to preserve!

In any case, it is wise to check your the BIOS settings for the hard disk geometry before beginning the installation, and write them down. While this should usually not be needed, it enables you to verify that the install program determines these values correctly.

The geometry that the BIOS uses will be referred to as the *BIOS geometry*, the geometry that NetBSD uses is the *real geometry*.

`sysinst` will try to discover both the real geometry and BIOS geometry.

It is *important* that `sysinst` know the proper BIOS geometry to be able to get NetBSD to boot, regardless of where on your disk you put it. It is less of a concern if the disk is going to be used entirely for NetBSD. If you intend to have several OSes on your disk, this becomes a much larger issue.

Installing the NetBSD System

Running the sysinst installation program

1. *Introduction*

   Using **sysinst**, installing NetBSD is a relatively easy process. You still should read this document and have it in hand when doing the installation process. This document tries to be a good guideline for the installation and as such covers many details to be completed. Do not let this discourage you, the install program is not hard to use.

2. *Possible* PCMCIA *issues*

   Machines with PCMCIA slots may have problems during installation. With the improvements of the PCMCIA code in this release, this will not happen very frequently. If you do not have PCMCIA on your machine (PCMCIA is only really used on laptop machines), you can skip this section, and ignore the ''[PCMCIA]'' notes. If you do have PCMCIA in your machine, you can safely ignore this section and the ''[PCMCIA]'' the first time, as you are likely to not have problems. Should troubles occur during floppy boot, they may be PCMCIA specific. You should then re-read this section and try again, following the instructions in the ''[PCMCIA]'' notes.

   This section explains how to work around the installation problem.

   The kernel keeps careful track of what interrupts and I/O ports are in use during autoconfiguration. It then allows the PCMCIA devices to pick unused interrupts and I/O ports. Unfortunately, the INSTALL kernel may not detect all devices in your system. This may be because the INSTALL kernel only supports the minimum set of devices to install NetBSD on your system, or it may be that NetBSD does not have support for the device causing the conflict.

   For example, suppose your laptop has a soundblaster device built in; the INSTALL kernel has no sound support. The PCMCIA code might allocate your soundblaster's IRQ and I/O ports to PCMCIA devices, causing them not to work, or to lock up the system. This is especially bad if one of the devices in question is your ethernet card.

   As of NetBSD 1.5, the kernel attempts to probe for available interrupts that are usable by the PCIC (PCMCIA interrupt controller). Assuming that this functions correctly, it should alleviate interrupt conflicts; however, I/O port conflicts are still possible.

   This problem will impact some, but not all, users of PCMCIA. If this problem is affecting you, watch the ''[PCMCIA]'' notes that will appear in this document.

   It can be difficult to distinguish an interrupt conflict from an I/O space conflict. There are no hard-and-fast rules, but interrupt conflicts are more likely to lock up the machine, and I/O space conflicts are more likely to result in misbehavior (e.g. a network card that cannot send or receive packets).

   The kernel selects a free interrupt according to a mask of allowable interrupts, stored in the kernel global variable *pcic_isa_intr_alloc_mask*. This mask is a logical-or of power-of-2s of allowable interrupts:

IRQ	Val	IRQ	Val	IRQ	Val	IRQ	Val
0	0x0001	4	0x0010	8	0x0100	12	0x1000
1	0x0002	5	0x0020	9	0x0200	13	0x2000
2	0x0004	6	0x0040	10	0x0400	14	0x4000
3	0x0008	7	0x0080	11	0x0800	15	0x8000

   For example, 0x0a00 allows both IRQ 9 and IRQ 11. By default, the INSTALL kernel permits all IRQs other than IRQs 5 and 7, so the corresponding mask is 0xff5f. The GENERIC kernel, however, allows all IRQs. (The presumption here is that IRQ 10 may be assigned to a device that the GENERIC kernel supports, but that the INSTALL does not.) Because of support for interrupt probing, it is no longer nec-

essary to exclude IRQs 3 and 5 explicitly; if they are in use, they should not be assigned to PCMCIA.

The kernel selects IO space by assigning cards IO space within a predefined range. The range is specified as a base and size, specified by the kernel global variables *pcic_isa_alloc_iobase* and *pcic_isa_alloc_iosize*. For systems with 12-bit addressing (most systems), the kernel defaults to a base of 0x400 and a size of 0xbff (a range of 0x400-0xfff). For systems with 10-bit addressing, the kernel defaults to a base of 0x300 and a size of 0xff (range of 0x300-0x3ff).

Unfortunately, these ranges may conflict with some devices. In the event of a conflict, try a base of 0x330 with a size of 0x0bf (range of 0x330-0x3ff).

In order to work around this at installation time, you may boot the `INSTALL` kernel with **`boot -d`**, in order to enter ddb(4) (the in-kernel debugger), and then use the **`write`** command to alter the variable values:

```
db> write pcic_isa_intr_alloc_mask 0x0a00
pcic_isa_intr_alloc_mask            0xff5f = 0xa00
db> write pcic_isa_alloc_iobase 0x330
pcic_isa_alloc_iobase         0x400 = 0x330
db> write pcic_isa_alloc_iosize 0x0bf
pcic_isa_alloc_iosize         0xbff = 0xbf
db> continue
```

Note that, since some floppy images may not have symbol information in the kernel, you may have to consult the matching `.symbols` file in the `binary/kernel` directory in the installation tree. Find the pcic_ symbols used above, look at the hexadecimal value in the first column, and write, for example (if *pcic_isa_intr_alloc_mask* is equal to c0513e3c):

```
db> write 0xc0513e3c 0x0a00
```

After installation, this value can be permanently written to the kernel image directly with:

```
# cp /netbsd /netbsd.bak
# gdb --write /netbsd
(gdb) set pcic_isa_intr_alloc_mask=0x0a00
(gdb) set pcic_isa_alloc_iobase=0x330
(gdb) set pcic_isa_alloc_iosize=0x0bf
(gdb) quit
#
```

or you could specify these value when configuring your kernel, e.g.:

```
options PCIC_ISA_INTR_ALLOC_MASK=0x0a00
options PCIC_ISA_ALLOC_IOBASE=0x330
options PCIC_ISA_ALLOC_IOSIZE=0x0bf
```

If you can get your PCMCIA card to work using this hack, you may also ignore the [PCMCIA] notes later in this document.

We hope to provide a more elegant solution to this problem in a future NetBSD release.

3. *General*

The following is a walk-through of the steps you will take while getting NetBSD installed on your hard disk. **`sysinst`** is a menu driven installation system that allows for some freedom in doing the installation. Sometimes, questions will be asked and in many cases the default answer will be displayed in brackets ("[ ]") after the question. If you wish to stop the installation, you may press `CONTROL-C` at any time, but if you do, you'll have to begin the installation process again from scratch.

4. *Quick install*

   First, let's describe a quick install. The other sections of this document go into the installation procedure in more detail, but you may find that you do not need this. If you want detailed instructions, skip to section 3. This section describes a basic installation, using a CD-ROM install as an example.

   - What you need.
     - The distribution sets (in this example, they are on CD).
     - Two 1.44 MB 3.5" floppy disks. One 1.44 MB 3.5" floppy.
     - A PC with a 386 or newer processor.
     - A CD-ROM drive (SCSI or ATAPI), a harddisk and a minimum of 4 MB of memory installed.
     - The harddisk should have at least 70 + *n* megabytes of space free, where *n* is the number of megabytes of main memory in your system. If you wish to install the X window system as well, you will need at least 60 MB more.
   - Creating the boot floppies. You can create the floppies needed for installation under MS-DOS or Windows. Supposing your 1.44 MB floppy drive is drive A:, and your CD is drive E: do the following from an MS-DOS command prompt:

     ```
     e:
     cd \NetBSD-1.5\i386\installation\misc
     rawrite
     ```

     When asked for a source filename, answer

     ```
     ..\floppy\boot1.fs
     ```

     for the first diskette and

     ```
     ..\floppy\boot2.fs
     ```

     for the second diskette.

     When asked for a destination drive answer '**a**'.
   - To create a bootfloppy under NetBSD or other UNIX-like system, you would type something like:

     ```
     # dd if=.../boot1.fs bs=18k of=/dev/rfd0a
     ```
   - The Quick Installation
     - Insert the first boot floppy you just created. Restart the computer. When prompted, insert the second boot floppy. The main menu will be displayed.
     - If you wish, you can configure some network settings immediately by choosing the **utilities** menu and then **configure network**. It isn't actually required at this point, but it may be more convenient. Go back to the main menu.
     - Choose **install**
     - You will be guided through some steps regarding the setup of your disk, and the selection of distributed components to install. When in doubt, refer to the rest of this document for details.
     - After your disk has been prepared, choose **CD-ROM** as the medium. The default values for the path and device should be ok.
     - After all the files have been unpacked, go back to the main menu and select **reboot**, after you have removed the bootfloppy from the drive.

- NetBSD will now boot. If you haven't already done so in sysinst, you should log in as `root`, and set a password for that account. You are also advised to edit the file `/etc/rc.conf` to match your system needs.
- Your installation is now complete.
- For configuring the X window system, if installed, see the files in `/usr/X11R6/lib/X11/doc`. Further information can be found on **http://www.xfree86.org/**

5. *Booting NetBSD*

   [PCMCIA] Unplug your PCMCIA devices, so that they won't be found by NetBSD.

   Boot your machine. The boot loader will start, and will print a countdown and begin booting.

   If the boot loader messages do not appear in a reasonable amount of time, you either have a bad boot floppy or a hardware problem. Try writing the install floppy image to a different disk, and using that.

   If that doesn't work, try booting after disabling your CPU's internal and external caches (if any). If it still doesn't work, NetBSD probably can't be run on your hardware. This can probably be considered a bug, so you might want to report it. If you do, please include as many details about your system configuration as you can.

   It will take a while to load the kernel from the floppy, probably around a minute or so, then, the kernel boot messages will be displayed. This may take a little while also, as NetBSD will be probing your system to discover which hardware devices are installed. You may want to read the boot messages, to notice your disk's name and geometry. Its name will be something like `sd0` or `wd0` and the geometry will be printed on a line that begins with its name. As mentioned above, you may need your disk's geometry when creating NetBSD 's partitions. You will also need to know the name, to tell **`sysinst`** on which disk to install. The most important thing to know is that `wd0` is NetBSD 's name for your first IDE disk, `wd1` the second, etc. `sd0` is your first SCSI disk, `sd1` the second, etc.

   Note that once the system has finished booting, you need not leave the floppy in the disk drive. Earlier version of the NetBSD install floppies mounted the floppy as the system's root partition (`/`), but the new installation floppies use a ramdisk file system and are no longer dependent on the floppy once it has booted.

   Once NetBSD has booted and printed all the boot messages, you will be presented with a welcome message and a main menu. It will also include instructions for using the menus.

6. *Network configuration*

   [PCMCIA] You can skip this section, as you will only get data from floppy in the first part of the install.

   If you will not use network operation during the installation, but you do want your machine to be configured for networking once it is installed, you should first go to the utilities menu, and select **`Configure network option`**. If you only want to temporarily use networking during the installation, you can specify these parameters later. If you are not using Domain Name Service (DNS), you can give an empty response in reply to answers relating to this.

7. *Installation drive selection and parameters*

   To start the installation, select the menu option to install NetBSD from the main menu.

   The first thing is to identify the disk on which you want to install NetBSD. **`sysinst`** will report a list of disks it finds and ask you for your selection. Depending on how many disks are found, you may get a different message. You should see disk names like `wd0`, `wd1`, `sd0` or `sd1`.

sysinst next tries to figure out the real and BIOS geometry of your disk. It will present you with the values it found, if any, and will give you a chance to change them.

Next, depending on whether you are using a `wd`*X* or `sd`*X* disk, you will either be asked for the type of disk (`wd`*X*) you are using or you will be asked if you want to specify a fake geometry for your SCSI disk (`sd`*X*). The types of disk are be IDE, ST-506 or ESDI. If you're installing on an ST-506 or ESDI drive, you'll be asked if your disk supports automatic sector forwarding. If you are *sure* that it does, reply affirmatively. Otherwise, the install program will automatically reserve space for bad144 tables.

8. *Partitioning the disk*

   - Which portion of the disk to use.

     You will be asked if you want to use the entire disk or only part of the disk. If you decide to use the entire disk for NetBSD, it will be checked if there are already other systems present on the disk, and you will be asked to confirm whether you want to overwrite these.

     If you want to use the entire disk for NetBSD, you can skip the following section and go to *Editing the NetBSD disklabel*.

   - *Editing the Master Boot Record*

     First, you will be prompted to specify the units of size that you want to express the sizes of the partitions in. You can either pick megabytes, cylinders or sectors.

     After this, you will be presented with the current values stored in the MBR, and will be given the opportunity to change, create or delete partitions. For each partition you can set the type, the start and the size. Setting the type to **unused** will delete a partition. You can also mark a partition as active, meaning that this is the one that the BIOS will start from at boot time.

     Be sure to mark the partition you want to boot from as active!

     After you are done editing the MBR, a sanity check will be done, checking for partitions that overlap. Depending on the BIOS capabilities of your machine and the parameters of the NetBSD partition you have specified, you may also be asked if you want to install newer bootcode in your MBR. If you have multiple operating systems on the disk that you are installing on, you will also be given the option to install a bootselector, that will allow you to pick the operating system to start up when your computer is (re-)started.

     If everything is ok, you can go on to the next step, editing the NetBSD disklabel.

   - *Editing the NetBSD disklabel*

     The partition table of the NetBSD part of a disk is called a *disklabel*. There are 3 layouts for the NetBSD part of the disk that you can pick from: **Standard, Standard with X** and **Custom**. The first two use a set of default values (that you can change) suitable for a normal installation, possibly including X. The last option lets you specify everything yourself.

     You will be presented with the current layout of the NetBSD disklabel, and given a chance to change it. For each partition, you can set the type, offset and size, block and fragment size, and the mount point. The type that NetBSD uses for normal file storage is called **4.2BSD**. A swap partition has a special type called **swap**. You can also specify a partition as type **MSDOS**. This is useful if you share the disk with MS-DOS or Windows; NetBSD is able to access the files on these partitions. You can use the values from the MBR for the MS-DOS part of the disk to specify the partition of type **MSDOS** (you don't have to do this now, you can always re-edit the disklabel to add this once you have installed NetBSD).

     Some partitions in the disklabel have a fixed purpose.

`a` Root partition. (`/`),

`b` Swap partition.

`c` The NetBSD portion of the disk.

`d` The entire disk.

`e-h` Available for other use. Traditionally, `e` is the partition mounted on `/usr`, but this is historical practice and not a fixed value.

You will then be asked to name your disk's disklabel. The default response is **mydisk**. For most purposes this will be OK. If you choose to name it something different, make sure the name is a single word and contains no special characters. You don't need to remember this name.

9. *Preparing your hard disk*

You are now at the point of no return. Nothing has been written to your disk yet, but if you confirm that you want to install NetBSD, your hard drive will be modified. If you are sure you want to proceed, enter `yes` at the prompt.

The install program will now label your disk and make the file systems you specified. The file systems will be initialized to contain NetBSD bootstrapping binaries and configuration files. You will see messages on your screen from the various NetBSD disk preparation tools that are running. There should be no errors in this section of the installation. If there are, restart from the beginning of the installation process. Otherwise, you can continue the installation program after pressing the return key.

Note: In previous versions of NetBSD, the kernel from the install floppy was copied onto the hard drive in a special step. In the new install system, the kernel on the floppy is unsuited to being copied onto the hard drive. Instead, a new set, **kern**, has been added which contains a generic kernel to be unloaded onto the drive. So, you can not boot from your hard drive yet at this point.

10. *Getting the distribution sets*

[PCMCIA] Load a kernel tar file (i.e. the `kern.tgz` set file) on to your hard disk, for example by mounting the hard disk first, copying the `kern.tgz` file from floppy and unpacking it. Example:

```
# mount /dev/wd0a /mnt
# cd /mnt

repeat the following 3 steps until all kern.* files are there
# mount -t msdos /dev/fd0a /mnt2
# cp /mnt2/kern.* .
# umount /mnt2
# cat kern.* | tar zxpvf -
```

Then halt the machine using the **halt** command. Power the machine down, and re-insert all the PCMCIA devices. Remove any floppy from the floppy drive. Start the machine up. After booting NetBSD, you will be presented with the main **sysinst** menu. Choose the option to re-install sets. Wait for the file system checks that it will do to finish, and then proceed as described below.

The NetBSD distribution consists of a number of *sets*, that come in the form of gzipped tarfiles. A few sets must be installed for a working system, others are optional. At this point of the installation, you will be presented with a menu which enables you to choose from one of the following methods of installing the sets. Some of these methods will first load the sets on your hard disk, others will extract the sets directly.

For all these methods, the first step is making the sets available for extraction, and then do the actual installation. The sets can be made available in a few different ways. The following sections describe each of those methods. After reading the one about the method you will be using, you can continue to section labeled 'Extracting the distribution sets'

11. *Installation using ftp*

    To be able to install using ftp, you first need to configure your network setup, if you haven't already at the start of the install procedure. **sysinst** will do this for you, asking you to provide some data, like IP number, hostname, etc. If you do not have name service set up for the machine that you are installing on, you can just press RETURN in answer to these questions, and DNS will not be used.

    You will also be asked to specify the host that you want to transfer the sets from, the directory on that host, and the account name and password used to log into that host using ftp. If you did not set up DNS when answering the questions to configure networking, you will need to specify an IP number instead of a hostname for the ftp server.

    sysinst will proceed to transfer all the default set files from the remote site to your hard disk.

12. *Installation using NFS*

    To be able to install using NFS, you first need to configure your network setup, if you haven't already at the start of the install procedure. **sysinst** will do this for you, asking you to provide some data, like IP number, hostname, etc. If you do not have name service set up for the machine that you are installing on, you can just press RETURN in answer to these questions, and DNS will not be used.

    You will also be asked to specify the host that you want to transfer the sets from, and the directory on that host that the files are in. This directory should be mountable by the machine you are installing on, i.e. correctly exported to your machine.

    If you did not set up DNS when answering the questions to configure networking, you will need to specify an IP number instead of a hostname for the NFS server.

13. *Installation from CD-ROM*

    When installing from a CD-ROM, you will be asked to specify the device name for your CD-ROM player (usually cd0), and the directory name on the CD-ROM where the distribution files are.

    sysinst will then check if the files are indeed available in the specified location, and proceed to the actual extraction of the sets.

14. *Installation from a floppy set*

    Because the installation sets are too big to fit on one floppy, the floppies are expected to be filled with the split set files. The floppies are expected to be in MS-DOS format. You will be asked for a directory where the sets should be reassembled. Then you will be prompted to insert the floppies containing the split sets. This process will continue until all the sets have been loaded from floppy.

15. *Installation from an unmounted file system*

    In order to install from a local file system, you will need to specify the device that the file system resides on (for example sd1e) the type of the file system, and the directory on the specified file system where the sets are located. **sysinst** will then check if it can indeed access the sets at that location.

16. *Installation from a local directory*

    This option assumes that you have already done some preparation yourself. The sets should be located in a directory on a file system that is already accessible. **sysinst** will ask you for the name of this directory.

17. *Extracting the distribution sets*

    After the install sets containing the NetBSD distribution have been made available, you can either extract all the sets (a full installation), or only extract sets that you have selected. In the latter case you will be shown the currently selected sets, and given the opportunity to select the sets you want. Some sets always need to be installed (**kern, base** and **etc**) they will not be shown in this selection menu.

    Before extraction begins, you can elect to watch the files being extracted; the name of each file that is extracted will be shown. This can slow down the installation process considerably, especially on machines with slow graphics consoles or serial consoles.

    After all the files have been extracted, all the necessary device node files will be created. If you have already configured networking, you will be asked if you want to use this configuration for normal operation. If so, these values will be installed in the network configuration files. The next menu will allow you to select the time zone that you're in, to make sure your clock has the right offset from GMT. Finally you can set a password for the "root" account, to prevent the machine coming up without access restrictions.

18. *Finalizing your installation*

    Congratulations, you have successfully installed NetBSD 1.5. You can now reboot the machine, and boot NetBSD from harddisk.

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, with the most important ones described below.

1. Configuring `/etc/rc.conf`

   If you or the installation software haven't done any configuration of `/etc/rc.conf`, 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-write. When the system asks you to choose a shell, simply press `RETURN` to get to a prompt. If you are asked for a terminal type, respond with **`vt220`** (or whatever is appropriate for your terminal type) and press `RETURN`. At this point, you need to configure at least one file in the `/etc` directory. 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).

   If your `/usr` directory is on a separate partition and you do not know how to use **`ed`**, you will have to mount your `/usr` partition to gain access to **`ex`** or **`vi`**. Do the following:

   ```
   # mount /usr
   # export TERM=vt220
   ```

   If you have `/var` on a separate partition, you need to repeat that step for it. After that, you can edit `/etc/rc.conf` with `vi`(1). When you have finished, type **`exit`** at the prompt to leave the single-user shell and continue with the multi-user boot.

   Other values that need to be set in `/etc/rc.conf` for a networked environment are *`hostname`* and possibly *`defaultroute`*, furthermore add an *`ifconfig_int`* for your interface <int>, along the lines of

```
ifconfig_de0="inet 123.45.67.89 netmask 255.255.255.0"
```

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

```
ifconfig_de0="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.

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. There is no initial password, but if you're using the machine in a networked environment, you should create an account for yourself (see below) and protect it and the "root" account with good passwords. Unless you have connected an unusual terminal device as the console you can just press `RETURN` when it prompts for `Terminal type? [...]`

3. Adding accounts

   Use the `useradd`(8) command to add accounts to your system, *do not* edit `/etc/passwd` directly. See `useradd`(8) for more information on how to add a new user to the system.

4. The X Window System

   If you have installed the X window system, look at the files in `/usr/X11R6/lib/X11/doc` for information.

   You will need to set up a configuration file, see `/usr/X11R6/lib/X11/XF86Config.eg` for an example. The `xf86config`(1) utility can interactively create a first version of such a configuration file for you. See **http://www.xfree86.org/** and the XFree86 manual page for more information.

   Don't forget to add `/usr/X11R6/bin` to your path in your shell's dot file so that you have access to the X binaries.

5. 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. This automatically handles any changes necessary to make the software run on NetBSD, retrieval and installation of any other packages on which the software may depend, and simplifies installation (and deinstallation), both from source and precompiled binaries.

   - More information on the package system is at
     http://www.netbsd.org/Documentation/software/packages.html
   - A browsable listing of available packages is at
     ftp://ftp.netbsd.org/pub/NetBSD/packages/pkgsrc/README.html
   - Precompiled binaries can be found at
     ftp://ftp.netbsd.org/pub/NetBSD/packages/
   - Package sources for compiling packages can be obtained by retrieving the file
     ftp://ftp.netbsd.org/pub/NetBSD/NetBSD-current/tar_files/pkgsrc.tar.gz
     They are typically extracted into `/usr/pkgsrc` (though other locations work fine), as with the command:

     ```
     # mkdir /usr/pkgsrc; tar -C /usr/pkgsrc -zxpf pkgsrc.tar.gz
     ```

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

6. Misc

   - Edit /etc/mail/aliases to forward root mail to the right place (run newaliases(1) afterwards.)
   - The /etc/mail/sendmail.cf file will almost definitely need to be adjusted; files aiding in this can be found in /usr/share/sendmail. See the README file there for more information.
   - 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

The upgrade to NetBSD 1.5 is a binary upgrade; it can be quite difficult to advance to a later version by recompiling from source due primarily to interdependencies in the various components.

To do the upgrade, you must have the boot floppy available. You must also have at least the **base** and **kern** binary distribution sets available, so that you can upgrade with them, using one of the upgrade methods described above. Finally, you must have sufficient disk space available to install the new binaries. Since the old binaries are being overwritten in place, you only need space for the new binaries, which weren't previously on the system. If you have a few megabytes free on each of your root (/) and /usr partitions, you should have enough space.

Since upgrading involves replacing the boot blocks on your NetBSD partition, the kernel, and most of the system binaries, it has the potential to cause data loss. You are strongly advised to *back up any important data on your disk*, whether on the NetBSD partition or on another operating system's partition, before beginning the upgrade process.

The upgrade procedure using the **sysinst** tool is similar to an installation, but without the hard disk partitioning. Another difference is that existing configuration files in /etc are backed up and merged with the new files. Getting the binary sets is done in the same manner as the installation procedure; refer to the installation part of the document for how to do this. Also, some sanity checks are done, i.e. file systems are checked before unpacking the sets.

After a new kernel has been copied to your hard disk, your machine is a complete NetBSD 1.5 system. However, that doesn't mean that you're finished with the upgrade process. You will probably want to update the set of device nodes you have in /dev. If you've changed the contents of /dev by hand, you will need to be careful about this, but if not, you can just cd into /dev, and run the command

```
# sh MAKEDEV all
```

You must also deal with certain changes in the formats of some of the configuration files. The most notable change is that the options given to many of the file systems in /etc/fstab have changed, and some of the file systems have changed names. To find out what the new options are, it's suggested that you read the manual page for the file system's mount commands, for example mount_nfs(8) for NFS.

Finally, you will want to delete old binaries that were part of the version of NetBSD that you upgraded from and have since been removed from the NetBSD distribution.

NetBSD/i386 has switched its executable format from the old a.out format to ELF, the now more commonly used and supported format. Your old binaries will continue to work just fine. The installation procedure will try to take the necessary steps to accomplish this. The most important step is to move the old a.out shared li-

braries in /usr/lib and /usr/X11R6/lib (if X was installed) to /emul/aout, where they will be automatically found if an older a.out binary is executed. Sysinst will use an existing /emul and / or /emul/aout directory if available, and will create it (as a symbolic link to /usr/aout) if necessary.

If you already had a /emul directory, or a symbolic link by that name, sysinst should rename it and tell you about it.

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 1.5.

General issues

- /etc/rc modified to use /etc/rc.d/*

  In previous releases of NetBSD, /etc/rc was a traditional BSD style monolithic file. As of NetBSD 1.5, each discrete program or substem from /etc/rc and /etc/netstart has been moved into separate scripts in /etc/rc.d/.

  At system startup, /etc/rc uses rcorder(8) to build a dependency list of the files in /etc/rc.d and then executes each script in turn with an argument of 'start'. Many rc.d scripts won't start unless the appropriate rc.conf(5) entry in /etc/rc.conf is set to 'YES.'

  At system shutdown, /etc/rc.shutdown uses rcorder(8) to build a dependency list of the files in /etc/rc.d that have a "KEYWORD: shutdown" line, reverses the resulting list, and then executes each script in turn with an argument of 'stop'. The following scripts support a specific shutdown method: cron, inetd, local, and xdm.

  Local and third-party scripts may be installed into /etc/rc.d as necessary. Refer to the other scripts in that directory and rc(8) for more information on implementing rc.d scripts.

Issues affecting an upgrading from NetBSD 1.4 or later

- named(8) leaks version information

  Previous releases of NetBSD disabled a feature of named(8) where the version number of the server could be determined by remote clients. This feature has not been disabled in NetBSD 1.5, because there is a named.conf(5) option to change the version string:

```
option {
        version "newstring";
};
```

- sysctl(8) pathname changed

  sysctl(8) is moved from /usr/sbin/sysctl to /sbin/sysctl. If you have hardcoded references to the full pathname (in shell scripts, for example) please be sure to update those.

- sendmail(8) configuration file pathname changed

  Due to sendmail(8) upgrade from 8.9.x to 8.10.x, /etc/sendmail.cf is moved to /etc/mail/sendmail.cf. Also, the default sendmail.cf(5) refers different pathnames than before. For example, /etc/aliases is now located at /etc/mail/aliases, /etc/sendmail.cw is now called /etc/mail/local-host-names, and so forth. If you have customized sendmail.cf(5) and friends, you will need to move the files to the new locations. See /usr/share/sendmail/README for more information.

Using online NetBSD documentation

Documentation is available if you first install 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.

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. Additionally, bug reports can be sent by mail to: netbsd-bugs@netbsd.org.

Use of send-pr(1) is encouraged, however, because bugs reported with it are entered into the NetBSD bugs database, and thus can't slip through the cracks.

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-help@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 ongoing work on BSD systems, support, and encouragement.

- Also, our thanks go to:

  Mike Hibler
  Rick Macklem
  Jan-Simon Pendry
  Chris Torek

  for answering lots of questions, fixing bugs, and doing the various work they've done.

- UC Berkeley's Experimental Computing Facility provided a home for sun-lamp in the past, people to look after it, and a sense of humor. Rob Robertson, too, has added his unique sense of humor to things, and for a long time provided the primary FTP site for NetBSD.

- Vixie Enterprises for hosting the NetBSD FTP, SUP, and WWW servers.

- Redback Networks, Inc. for hosting the NetBSD mail and GNATS server.

- The Helsinki University of Technology in Finland for hosting the NetBSD CVS server.

- The Internet Research Institute in Japan for hosting the server which runs the CVSweb interface to the NetBSD source tree.

- The many organisations 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.

- Dave Burgess `burgess@cynjut.infonet.net` has been maintaining the 386BSD/NetBSD/FreeBSD FAQ for quite some time, and deserves to be recognized for it.

- The following individuals and organizations (each in alphabetical order) have made donations or loans of hardware and/or money, to support NetBSD development, and deserve credit for it:

  Steve Allen
  Jason Birnschein
  Mason Loring Bliss
  Jason Brazile
  Mark Brinicombe
  David Brownlee
  Simon Burge
  Dave Burgess
  Ralph Campbell
  Brian Carlstrom
  James Chacon
  Bill Coldwell
  Charles Conn
  Tom Coulter
  Charles D. Cranor
  Christopher G. Demetriou

Scott Ellis
Hubert Feyrer
Castor Fu
Greg Gingerich
William Gnadt
Michael Graff
Guenther Grau
Ross Harvey
Charles M. Hannum
Michael L. Hitch
Kenneth Alan Hornstein
Jordan K. Hubbard
Søren Jørvang
Scott Kaplan
Noah M. Keiserman
Harald Koerfgen
John Kohl
Chris Legrow
Ted Lemon
Norman R. McBride
Neil J. McRae
Perry E. Metzger
Toru Nishimura
Herb Peyerl
Mike Price
Dave Rand
Michael Richardson
Heiko W. Rupp
Brad Salai
Chuck Silvers
Thor Lancelot Simon
Bill Sommerfeld
Paul Southworth
Eric and Rosemary Spahr
Ted Spradley
Kimmo Suominen
Jason R. Thorpe
Steve Wadlow
Krister Walfridsson
Jim Wise
Christos Zoulas

AboveNet Communications, Inc.
Advanced System Products, Inc.
Avalon Computer Systems
Bay Area Internet Solutions
Brains Corporation, Japan
Canada Connect Corporation
Co-operative Research Centre for Enterprise Distributed Systems Technology
Demon Internet, UK
Digital Equipment Corporation
Distributed Processing Technology

Easynet, UK
Free Hardware Foundation
Innovation Development Enterprises of America
Internet Software Consortium
MS Macro System GmbH, Germany
Numerical Aerospace Simulation Facility, NASA Ames Research Center
Piermont Information Systems Inc.
Salient Systems Inc.
VMC Harald Frank, Germany
Warped Communications, Inc.
Whitecross Database Systems Ltd.

(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 mentioned, tell us!)

We are...

(in alphabetical order)

The NetBSD core group:

Alistair Crooks	agc@netbsd.org
Jun-ichiro itojun Hagino	itojun@netbsd.org
Frank van der Linden	fvdl@netbsd.org
Luke Mewburn	lukem@netbsd.org
Christos Zoulas	christos@netbsd.org

The portmasters (and their ports):

Mark Brinicombe	mark@netbsd.org	**arm32**
Jeremy Cooper	jeremy@netbsd.org	**sun3x**
Ross Harvey	ross@netbsd.org	**alpha**
Jun-ichiro itojun Hagino	itojun@netbsd.org	**sh3**
Ben Harris	bjh21@netbsd.org	**arm26**
Eduardo Horvath	eeh@netbsd.org	**sparc64**
Darrin Jewell	dbj@netbsd.org	**next68k**
Søren Jørvang	soren@netbsd.org	**cobalt**
Søren Jørvang	soren@netbsd.org	**sgimips**
Wayne Knowles	wdk@netbsd.org	**mipsco**
Paul Kranenburg	pk@netbsd.org	**sparc**
Anders Magnusson	ragge@netbsd.org	**vax**
Minoura Makoto	minoura@netbsd.org	**x68k**
Phil Nelson	phil@netbsd.org	**pc532**
Tohru Nishimura	nisimura@netbsd.org	**luna68k**
NONAKA Kimihiro	nonaka@netbsd.org	**prep**
Scott Reynolds	scottr@netbsd.org	**mac68k**
Kazuki Sakamoto	sakamoto@netbsd.org	**bebox**
Noriyuki Soda	soda@netbsd.org	**arc**

Wolfgang Solfrank	ws@netbsd.org	**ofppc**
Ignatios Souvatzis	is@netbsd.org	**amiga**
Jonathan Stone	jonathan@netbsd.org	**pmax**
Shin Takemura	takemura@netbsd.org	**hpcmips**
Jason Thorpe	thorpej@netbsd.org	**alpha**
Jason Thorpe	thorpej@netbsd.org	**hp300**
Tsubai Masanari	tsubai@netbsd.org	**macppc**
Tsubai Masanari	tsubai@netbsd.org	**newsmips**
Izumi Tsutsui	tsutsui@netbsd.org	**news68k**
Frank van der Linden	fvdl@netbsd.org	**i386**
Leo Weppelman	leo@netbsd.org	**atari**
Nathan Williams	nathanw@netbsd.org	**sun3**
Steve Woodford	scw@netbsd.org	**mvme68k**

The NetBSD 1.5 Release Engineering team:

Chris G. Demetriou	cgd@netbsd.org
Havard Eidnes	he@netbsd.org
Ted Lemon	mellon@netbsd.org
John Hawkinson	jhawk@netbsd.org
Perry Metzger	perry@netbsd.org
Curt Sampson	cjs@netbsd.org
Jason Thorpe	thorpej@netbsd.org
Todd Vierling	tv@netbsd.org

Developers and other contributors:

Steve Allen	wormey@netbsd.org
Julian Assange	proff@netbsd.org
Lennart Augustsson	augustss@netbsd.org
Christoph Badura	bad@netbsd.org
Robert V. Baron	rvb@netbsd.org
Erik Berls	cyber@netbsd.org
John Birrell	jb@netbsd.org
Mason Loring Bliss	mason@netbsd.org
Manuel Bouyer	bouyer@netbsd.org
John Brezak	brezak@netbsd.org
Allen Briggs	briggs@netbsd.org
Aaron Brown	abrown@netbsd.org
David Brownlee	abs@netbsd.org
Frederick Bruckman	fredb@netbsd.org
Jon Buller	jonb@netbsd.org
Simon Burge	simonb@netbsd.org
Dave Burgess	burgess@cynjut.infonet.net
Robert Byrnes	byrnes@netbsd.org
D'Arcy J.M. Cain	darcy@netbsd.org
Dave Carrel	carrel@netbsd.org
James Chacon	jmc@netbsd.org
Bill Coldwell	billc@netbsd.org

Julian Coleman	jdc@netbsd.org
Chuck Cranor	chuck@netbsd.org
Aidan Cully	aidan@netbsd.org
Johan Danielsson	joda@netbsd.org
Matt DeBergalis	deberg@netbsd.org
Rob Deker	deker@netbsd.org
Chris G. Demetriou	cgd@netbsd.org
Jaromir Dolecek	jdolecek@netbsd.org
Andy Doran	ad@netbsd.org
Roland Dowdeswell	elric@netbsd.org
Matthias Drochner	drochner@netbsd.org
Jun Ebihara	jun@netbsd.org
Havard Eidnes	he@netbsd.org
Enami Tsugutomo	enami@netbsd.org
Bernd Ernesti	veego@netbsd.org
Erik Fair	fair@netbsd.org
Hubert Feyrer	hubertf@netbsd.org
Thorsten Frueauf	frueauf@netbsd.org
Castor Fu	castor@netbsd.org
Ichiro Fukuhara	ichiro@netbsd.org
Brian R. Gaeke	brg@dgate.org
Thomas Gerner	thomas@netbsd.org
Simon J. Gerraty	sjg@netbsd.org
Justin Gibbs	gibbs@netbsd.org
Adam Glass	glass@netbsd.org
Michael Graff	explorer@netbsd.org
Brad Grantham	grantham@tenon.com
Brian C. Grayson	bgrayson@netbsd.org
Matthew Green	mrg@netbsd.org
Juergen Hannken-Illjes	hannken@netbsd.org
Charles M. Hannum	mycroft@netbsd.org
Eric Haszlakiewicz	erh@netbsd.org
John Hawkinson	jhawk@netbsd.org
HAYAKAWA Koichi	haya@netbsd.org
René Hexel	rh@netbsd.org
Michael L. Hitch	mhitch@netbsd.org
Christian E. Hopps	chopps@netbsd.org
Ken Hornstein	kenh@netbsd.org
Marc Horowitz	marc@netbsd.org
Nick Hudson	skrll@netbsd.org
Martin Husemann	martin@netbsd.org
Dean Huxley	dean@netbsd.org
Bernardo Innocenti	bernie@netbsd.org
ITOH Yasufumi	itohy@netbsd.org
IWAMOTO Toshihiro	toshii@netbsd.org
Matthew Jacob	mjacob@netbsd.org
Lonhyn T. Jasinskyj	lonhyn@netbsd.org
Chris Jones	cjones@netbsd.org
Takahiro Kambe	taca@netbsd.org

Antti Kantee	pooka@netbsd.org
Lawrence Kesteloot	kesteloo@cs.unc.edu
Thomas Klausner	wiz@netbsd.org
Klaus Klein	kleink@netbsd.org
Wayne Knowles	wdk@netbsd.org
John Kohl	jtk@netbsd.org
Kevin Lahey	kml@netbsd.org
Johnny C. Lam	jlam@netbsd.org
Martin J. Laubach	mjl@netbsd.org
Ted Lemon	mellon@netbsd.org
Joel Lindholm	joel@netbsd.org
Mike Long	mikel@netbsd.org
Warner Losh	imp@netbsd.org
Federico Lupi	federico@netbsd.org
Brett Lymn	blymn@netbsd.org
Paul Mackerras	paulus@netbsd.org
David Maxwell	david@netbsd.org
Dan McMahill	dmcmahill@netbsd.org
Gregory McGarry	gmcgarry@netbsd.org
Neil J. McRae	neil@netbsd.org
Perry Metzger	perry@netbsd.org
der Mouse	mouse@netbsd.org
Joseph Myers	jsm@netbsd.org
Ken Nakata	kenn@netbsd.org
Bob Nestor	rnestor@netbsd.org
NONAKA Kimihiro	nonaka@netbsd.org
Masaru Oki	oki@netbsd.org
Atsushi Onoe	onoe@netbsd.org
Greg Oster	oster@netbsd.org
Herb Peyerl	hpeyerl@netbsd.org
Matthias Pfaller	matthias@netbsd.org
Dante Profeta	dante@netbsd.org
Chris Provenzano	proven@netbsd.org
Waldi Ravens	waldi@moacs.indiv.nl.net
Darren Reed	darrenr@netbsd.org
Michael Richardson	mcr@netbsd.org
Tim Rightnour	garbled@netbsd.org
Gordon Ross	gwr@netbsd.org
Heiko W. Rupp	hwr@netbsd.org
SAITOH Masanobu	msaitoh@netbsd.org
Curt Sampson	cjs@netbsd.org
Wilfredo Sanchez	wsanchez@netbsd.org
Ty Sarna	tsarna@netbsd.org
SATO Kazumi	sato@netbsd.org
Matthias Scheler	tron@netbsd.org
Karl Schilke (rAT)	rat@netbsd.org
Konrad Schroder	perseant@netbsd.org
Reed Shadgett	dent@netbsd.org
Tim Shepard	shep@netbsd.org

Takao Shinohara	shin@netbsd.org
Takuya SHIOZAKI	tshiozak@netbsd.org
Chuck Silvers	chs@netbsd.org
Thor Lancelot Simon	tls@netbsd.org
Jeff Smith	jeffs@netbsd.org
Bill Sommerfeld	sommerfeld@netbsd.org
Bill Squier	groo@netbsd.org
Bill Studenmund	wrstuden@netbsd.org
Kevin Sullivan	sullivan@netbsd.org
SUNAGAWA Keiki	kei@netbsd.org
Kimmo Suominen	kim@netbsd.org
Matt Thomas	matt@netbsd.org
Christoph Toshok	toshok@netbsd.org
UCHIYAMA Yasushi	uch@netbsd.org
Shuichiro URATA	ur@netbsd.org
Todd Vierling	tv@netbsd.org
Aymeric Vincent	aymeric@netbsd.org
Paul Vixie	vixie@netbsd.org
Krister Walfridsson	kristerw@netbsd.org
Lex Wennmacher	wennmach@netbsd.org
Assar Westerlund	assar@netbsd.org
Todd Whitesel	toddpw@netbsd.org
Rob Windsor	windsor@netbsd.org
Dan Winship	danw@netbsd.org
Jim Wise	jwise@netbsd.org
Michael Wolfson	mbw@netbsd.org
Colin Wood	ender@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:

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by The NetBSD Foundation, Inc.

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

This product includes software developed by the Computer Systems Engineering Group at Lawrence Berkeley Laboratory.

This product includes software developed by Adam Glass and Charles Hannum.

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 Alistair G. Crooks.

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 Charles D. Cranor and Washington University.

This product includes software developed by Charles D. Cranor.

This product includes software developed by Charles 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 Hannum.

This product includes software developed by Charles M. Hannum.

This product includes software developed by Chris Provenzano.

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 David Jones and Gordon Ross.

This product includes software developed by Dean Huxley.

This product includes software developed by Eric S. Hvozda.

This product includes software developed by Ezra Story.

This product includes software developed by Gardner Buchanan.

This product includes software developed by Gordon Ross.

This product includes software developed by Gordon W. Ross and Leo Weppelman.

This product includes software developed by Gordon W. Ross.

This product includes software developed by Hauke Fath.

This product includes software developed by HAYAKAWA Koichi.

This product includes software developed by Hellmuth Michaelis and Joerg Wunsch.

This product includes software developed by Herb Peyerl.

This product includes software developed by Holger Veit and Brian Moore for use with "386BSD" and similar operating systems.

This product includes software developed by Hubert Feyrer for the NetBSD Project.

This product includes software developed by Iain Hibbert.

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 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 John Polstra.

This product includes software developed by Jonathan Stone and Jason R. Thorpe for the NetBSD Project.

This product includes software developed by Jonathan Stone 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 Kenneth Stailey.

This product includes software developed by Leo Weppelman.

This product includes software developed by Lloyd Parkes.

This product includes software developed by Manuel Bouyer.

This product includes software developed by Marc Horowitz.

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 Martin Husemann and Wolfgang Solfrank.

This product includes software developed by Mats O Jansson and Charles D. Cranor.

This product includes software developed by Mats O Jansson.

This product includes software developed by Matthias Pfaller.

This product includes software developed by Michael L. Hitch.

This product includes software developed by Niels Provos.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Paul Mackerras.

This product includes software developed by Peter Galbavy.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Rodney W. Grimes.

This product includes software developed by Roland C. Dowdeswell.

This product includes software developed by Rolf Grossmann.

This product includes software developed by Scott Bartram.

This product includes software developed by SigmaSoft, Th. Lockert.

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 Theo de Raadt and John Brezak.

This product includes software developed by Theo de Raadt.

This product includes software developed by Tohru Nishimura for the NetBSD Project.

This product includes software developed by TooLs GmbH.

This product includes software developed by Winning Strategies, Inc.

This product includes software developed by Zembu Labs, Inc.

This product includes software developed by the Center for Software Science at the University of Utah.

This product includes software developed by the Computer Systems Laboratory at the University of Utah.

This product includes software developed by the University of Calgary Department of Computer Science and its contributors.

This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.

This product includes software developed for the FreeBSD project.

This product includes software developed for the Internet Software Consortium by Ted Lemon.

This product includes software developed for the NetBSD Project by Frank van der Linden.

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 for the NetBSD Project by Matthias Drochner.

This product includes software developed for the NetBSD Project by Matthieu Herrb.

This product includes software developed for the NetBSD Project by Perry E. Metzger.

This product includes software developed for the NetBSD Project by Piermont Information Systems Inc.

This product includes software developed for the NetBSD Project by Ted Lemon.

This product includes software developed by LAN Media Corporation and its contributors.

This product includes software developed by Michael Graff for the NetBSD Project.

This product includes software developed by Niklas Hallqvist, C Stone and Job de Haas.

This product includes software developed by Eric Young (eay@mincom.oz.au).

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 University of Oregon.

This product includes software developed by the University of Southern California and/or Information Sciences Institute.

This product includes software developed by Internet Initiative Japan Inc.

In the following statement, "This software" refers to the Mitsumi CD-ROM driver:

> 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).

In the following statement, "This software" refers to the parallel port driver:

> This software is a component of "386BSD" developed by William F. Jolitz, TeleMuse.