NAME

INSTALL – Installation procedure for NetBSD/alpha

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 1.4.1 on the *alpha* 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 UN*X-like operating system derived from the Berkeley Networking Release 2 (Net/2), 4.4BSD-Lite, and 4.4BSD-Lite2 sources. NetBSD runs on twenty different system architectures featuring eight distinct families of CPUs, and is being ported to more. The NetBSD 1.4.1 release contains complete binary releases for fourteen different machine types. (The six 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, 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 1.4.1 release is a substantial improvement over its predecessors. We have provided 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 results of these improvements is a stable operating system fit for production use that rivals most commercially available systems.

It is impossible to completely summarize the nearly two years of development that went into the NetBSD 1.4.1 release. Some highlights include:

- Substantial improvements in the TCP/IP implementation, including numerous performance enhancements and bug fixes by Jason Thorpe and others.
- A new, high efficiency kernel memory pool allocator by Paul Kranenburg. This has been integrated into most kernel subsystems.
- A new, totally rewritten virtual memory subsystem, UVM, created by Chuck Cranor, which is substantially cleaner and better performing than the old Mach derived VM subsystem.
- Improved POSIX and XPG standards compliance.

- Completion of the integration of all remaining 4.4BSD Lite-2 kernel improvements and bug fixes that had not been previously integrated. (Integration of all userland components was completed before NetBSD 1.3)
- Several new ports, including macppc, bebox, sparc64, next68k, and others, have been integrated into the source tree.
- The system compilers have been upgraded to egcs 1.1.1, and the system compiler toolchain now (mostly) uses the latest versions of GNU binutils instead of the obsolete versions left over from 4.4BSD Lite.
- Everyone's favorite `ftp`(1) client has been improved even further. See the man page for details.
- A new architecture independent console driver, `wscons`(4), has been integrated into many ports.
- Numerous improvements have been made to the audio subsystem support, including support for MIDI device drivers.
- Linux compatibility support has been improved.
- A number of scheduler enhancements have yielded dramatic improvements in interactive performance and better control of background tasks.
- Several network tunneling protocols, including GRE and IP in IP, have been implemented.
- Kernel support for the CODA distributed file system has been added.
- Manuel Bouyer completed major changes to the IDE support. It is now architecture independent. Major changes have been made to the IDE code for better error handling, improved ATAPI support, 32 bit data I/O support and bus-master DMA support on PCI IDE controllers.
- Lennart Augustsson has added full USB support, permitting the use of a wide variety of Universal Serial Bus peripherals. The drivers should easily port to any future platforms that support the PCI bus. See `usb`(4) for an overview.
- RAIDframe, version 1.1, from the Parallel Data Laboratory at Carnegie Mellon University, has been integrated. Supports RAID levels 0, 1, 4, 5, and more.
- Luke Mewburn added `nsswitch.conf`(5) functionality to the system to specify the search order for system databases.
- `syslogd`(8) now supports listening on multiple sockets, to make the chrooting of servers easier.
- Most third party packages have been updated to the latest stable release.

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.

There have been many, many Alpha-specific enchancements since the 1.3 release. These include:

- Many new system types are supported:
  - EB66 Evaluation Board
  - ALPHABook 1
  - Digital AlphaServer 4100 systems
  - Digital AlphaServer 1000 systems
  - Digital AlphaServer 1000A systems
  - Digital AlphaServer 800 systems

  – Digital Server 330x systems

- Jason Thorpe's new virtual memory code provides full support for the alpha's large address space with context switching and translation buffer state optimizations.
- Extensive scheduler work enables process nice levels to work as expected, fixing ancient BSD scheduler bugs that affected NetBSD/alpha more than other ports with slower clock Hz defaults. Nice +19 and +20 processes now take no time away from nice +0 programs regardless of load average.
- Many new device drivers and sound cards are supported. Floppy disk drives are now supported.
- IDE peripheral devices are now supported on Multia and AXPPCI33 systems.
- A new multi-volume boot loader allows all supported system types and device drivers to be present on floppy disk media, and enables the *sysinst* screen-menu installation program to be included on floppy disks as well. Although two floppies are now used, they load much faster than the single floppy did in 1.3.
- Installation from CD and magnetic tape media is now supported.
- A new set of boot programs provides considerable operational and performance enhancement.

NetBSD 1.4.1 on alpha is, as usual, also fully backward compatible with old NetBSD/alpha binaries, so you don't need to recompile all your local programs.

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 foundation 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. In addition, we intend to provide Anonymous CVS access to the NetBSD source tree in the near future, so that anyone on the internet can examine the full NetBSD source code repository.

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

NetBSD Mirror Site List

The following sites mirror NetBSD as of April 03, 1999.

If you wish to become a distribution site for NetBSD, contact `mirrors@netbsd.org`.

FTP mirrors

Australia

ftp.au.netbsd.org
RMIT University, Melbourne
ftp://ftp.au.netbsd.org/pub/NetBSD/

ftp2.au.netbsd.org
University of Queensland, Brisbane
ftp://ftp2.au.netbsd.org/pub/NetBSD/

Austria

ftp.at.netbsd.org
University of Technology, Vienna
ftp://ftp.at.netbsd.org/pub/NetBSD/

Brazil

ftp.ravel.ufrj.br
Cidade Universitaria
ftp://ftp.ravel.ufrj.br/pub/NetBSD/

Denmark

ftp.dk.netbsd.org
Aalborg University
ftp://ftp.dk.netbsd.org/pub/NetBSD/

Finland

ftp.fi.netbsd.org
The Finnish University and Research Network, Espoo
ftp://ftp.fi.netbsd.org/pub/NetBSD/

France

ftp.fr.netbsd.org
Paris University
ftp://ftp.fr.netbsd.org/pub/NetBSD/

Germany

ftp.de.netbsd.org
University of Trier
ftp://ftp.de.netbsd.org/pub/NetBSD/

ftp2.de.netbsd.org
University of Erlangen-Nuremberg
ftp://ftp2.de.netbsd.org/pub/NetBSD/

ftp.uni-regensburg.de
University of Regensburg
ftp://ftp.uni-regensburg.de/pub/comp/os/NetBSD/

Japan

ftp.jp.netbsd.org
Internet Research Institute Inc., Tokyo
ftp://ftp.jp.netbsd.org/pub/NetBSD/

core.ring.gr.jp
Electrotechnical Laboratory
ftp://core.ring.gr.jp/pub/NetBSD/

ftp.dti.ad.jp
Dream Train Internet Inc., Tokyo
ftp://ftp.dti.ad.jp/pub/NetBSD/

mirror.nucba.ac.jp
Nagoya University of Commerce and Business
ftp://mirror.nucba.ac.jp/mirror/NetBSD/

netbsd.tohoku.ac.jp
Tohoku University, Sendai
ftp://netbsd.tohoku.ac.jp/NetBSD/

ring.asahi-net.or.jp
ASAHI Net
ftp://ring.asahi-net.or.jp/pub/NetBSD/

Netherlands

ftp.nl.netbsd.org
University of Amsterdam
ftp://ftp.nl.netbsd.org/pub/NetBSD/

Norway

ftp.no.netbsd.org
Bergen IT Consult AS
ftp://ftp.no.netbsd.org/pub/NetBSD/

ftp.ntnu.no
Norwegian University of Science and Technology
ftp://ftp.ntnu.no/pub/NetBSD/

Russia

ftp.ru.netbsd.org
Landau Institute for Theoretical Physics, Chernogolovka
ftp://ftp.ru.netbsd.org/pub/NetBSD/

Sweden

ftp.stacken.kth.se
Royal Institute of Technology, Stockholm
ftp://ftp.stacken.kth.se/pub/OS/NetBSD/

ftp.sunet.se
Swedish University NETwork, Uppsala

ftp://ftp.sunet.se/pub/os/NetBSD/

UK

ftp.uk.netbsd.org
Domino, London
ftp://ftp.uk.netbsd.org/pub/NetBSD/

sunsite.org.uk
ftp://sunsite.org.uk/packages/netbsd/

USA

ftp.netbsd.org
Silicon Valley, California
ftp://ftp.netbsd.org/pub/NetBSD/

ftp.cs.umn.edu
University of Minnesota
ftp://ftp.cs.umn.edu/pub/NetBSD

ftp.eecs.umich.edu
University of Michigan, Ann Arbor
ftp://ftp.eecs.umich.edu/pub/NetBSD/

ftp.iastate.edu
Iowa State University
ftp://ftp.iastate.edu/pub/netbsd/

ftp.op.net
ftp://ftp.op.net/pub/NetBSD/

AFS mirrors

Sweden

ftp.stacken.kth.se
Royal Institute of Technology, Stockholm
AFS path: /afs/stacken.kth.se/ftp/pub/OS/NetBSD

USA

ftp.iastate.edu
Iowa State University
AFS path: /afs/iastate.edu/public/ftp/pub/netbsd

NFS mirrors

UK

sunsite.org.uk
Instructions: mount -o ro sunsite.org.uk:/public/packages/netbsd /mnt

SUP mirrors

Australia

sup.au.netbsd.org
RMIT University, Melbourne
Instructions: ftp://sup.au.netbsd.org/pub/NetBSD/sup/README.sup

France

sup.fr.netbsd.org
Paris University
Instructions: Similar to sup.netbsd.org

Germany

sup.de.netbsd.org
University of Trier
Instructions: ftp://sup.de.netbsd.org/pub/NetBSD/sup/supfile.example

Japan

sup.jp.netbsd.org
Internet Research Institute Inc., Tokyo
Instructions: ftp://sup.jp.netbsd.org/pub/NetBSD/sup/README.sup

Norway

sup.no.netbsd.org
Norwegian University of Science and Technology
Instructions: See /usr/src/share/examples/supfiles/sup.no.netbsd.org

UK

sup.uk.netbsd.org
Domino, London
Instructions: See ftp://ftp.netbsd.org/pub/NetBSD/sup/README.sup

USA

sup.netbsd.org
Silicon Valley, California
Instructions: See ftp://ftp.netbsd.org/pub/NetBSD/sup/README.sup

ftp.cs.umn.edu
University of Minnesota
Instructions: hostbase=/ftp/ftp/packages/NetBSD, collections are the same as on sup.NetBSD.ORG

WWW mirrors

Australia

www.au.netbsd.org
RMIT University, Melbourne
http://www.au.netbsd.org/

Austria

www.at.netbsd.org
University of Technology, Vienna
http://www.at.netbsd.org/

Finland

www.fi.netbsd.org
Global Wire Oy, Lappeenranta
http://www.fi.netbsd.org/

France

www.fr.netbsd.org
Paris University
http://www.fr.netbsd.org/

Germany

www.de.netbsd.org
http://www.de.netbsd.org/

Japan

www.jp.netbsd.org
Internet Research Institute Inc., Tokyo
http://www.jp.netbsd.org/

Norway

www.no.netbsd.org
Bergen IT Consult AS
http://www.no.netbsd.org/

USA

www.netbsd.org
Western Washington State University
http://www.netbsd.org/

www2.us.netbsd.org
New York
http://www.us.netbsd.org/

NetBSD 1.4.1 Release Contents

The root directory of the NetBSD 1.4.1 release is organized as follows:

`.../NetBSD-1.4.1/`

`BUGS`	Known bugs list (somewhat incomplete and out of date).
`CHANGES`	Changes since earlier NetBSD releases.
`LAST_MINUTE`	Last minute changes.
`MIRRORS`	A list of sites that mirror the NetBSD 1.4.1 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 1.4.1 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 (i.e. the 'domestic' portion) that may be subject to export regulations of the United States. 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:

secrsrc.tgz: This set contains the "domestic" sources. These sources may be subject to United States export regulations.
421K gzipped, 2M uncompressed

gnusrc.tgz: This set contains the "gnu" sources, including the source for the compiler, assembler, groff, and the other GNU utilities in the binary distribution sets.
19M gzipped, 84.2M uncompressed

syssrc.tgz: This set contains the sources to the NetBSD 1.4.1 kernel, config(8), and dbsym(8).
13.5M gzipped, 66.7M uncompressed

sharesrc.tgz: 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.
3M gzipped, 11.9M uncompressed

src.tgz: This set contains all of the NetBSD 1.4.1 sources which are not mentioned above.
16.1M gzipped, 73.6M uncompressed

Most of the above source sets are located in the `source/sets` subdirectory of the distribution tree. The secrsrc.tgz set is contained in the `source/security` subdirectory. This set, which is available only to users in the United States and Canada, contains the sources normally found in `/usr/src/domestic` – primarily kerberos and other cryptographic security related software. (Remember, because of United States law, it may not be legal to distribute this set to locations outside of the United States and Canada.)

The source sets are distributed as compressed tar files. They may be unpacked into `/usr/src` with the command:

```
cat set_name.tgz | gunzip | (cd /; tar xpf - )
```

The `sets/Split/` and `security/Split/` subdirectories contain 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 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.?? | gunzip | (cd /; tar xpf - )
```

In each of the source distribution set directories, there is a file named `CKSUMS` which contains the checksums of the files in that directory, as generated by the `cksum`(1) utility. You can use cksum to check the integrity of the archives, if you suspect that one of the files is corrupt and have access to a cksum binary. Checksums based on other algorithms may also be present – see the `release`(7) man page for details.

NetBSD/alpha Subdirectory Structure

The alpha-specific portion of the NetBSD 1.4.1 release is found in the `alpha` subdirectory of the distribution:
```
.../NetBSD-1.4.1/alpha/
INSTALL.html
```

`INSTALL.ps`		
`INSTALL.txt`		
`INSTALL.more`	Installation notes; this file. The `.more` file contains underlined text using the `more`(1) conventions for indicating italic and bold display.	
`binary/`		
	`sets/`	alpha binary distribution sets; see below.
	`security/`	alpha security distribution; see below.
`installation/`		
	`floppy/`	alpha boot and installation floppies; see below.
	`diskimage/`	
	`tapeimage/`	an image file `cdhdtape` is included for the case where the installer is written to a CD, hard drive, or tape. This image file is the same for the CD, HD, and tape cases, but a separate `tapeimage/` directory exists to hold a copy of the README file and to meet the NetBSD `release`(7) standard.
	`instkernel/`	contains a `netbsd.gz` installation kernel; this is the same installer kernel as in all the other install images, but without the various boot program and filesystem wrappers. It can be netbooted or diskbooted from a previous installation. Note: there is no need to ungzip this image.
	`misc/`	contains `GENERIC.fs`, a GENERIC kernel in a bootable filesystem image. This is used in some unusual installations as described in the next section.

Bootable installation/upgrade floppies:

There are three bootable images in the NetBSD alpha distribution. One is for a dual-floppy boot and is split into two separate files. The other is a single-file image containing the same install kernel, but intended to be written to a CD, tape, or hard drive. The third image is a GENERIC kernel intended for production use in unusual cases. This can be useful at some sites when:

- You want to run diskless but SRM bugs prevent the firmware from netbooting. You can work around this problem by always booting the generic kernel from the floppy.
- SRM doesn't recognize your (hard) disk controller but NetBSD does. This happens more frequently than you might think. SRM will usually only boot from `ncr`(4) or `isp`(4) SCSI devices, and on most platforms will not boot from an IDE drive. NetBSD will happily operate with almost any SCSI root or an IDE root; the solution here is to netboot a kernel or always boot from floppy.

Binary Distribution Sets

The NetBSD alpha binary distribution sets contain the binaries which comprise the NetBSD 1.4.1 release for the alpha. There are eight binary distribution sets and the *security* distribution set. The binary distribution sets can be found in the `alpha/binary/sets` subdirectory of the NetBSD 1.4.1 distribution tree, and are as follows:

base The NetBSD 1.4.1 alpha **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.
14.5M gzipped, 38.9M 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.

11.6M gzipped, 48.7M 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.)
56K gzipped, 330K uncompressed

games This set includes the games and their manual pages.
3M gzipped, 7.4M uncompressed

kern This set contains a NetBSD/alpha 1.4.1 GENERIC kernel, named `/netbsd`. You *must* install this distribution set.
1.3M gzipped, 3M 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.
4M gzipped, 15.9M uncompressed

misc This set includes the (rather large) system dictionaries, the typesettable document set, and other files from `/usr/share`.
2.2M gzipped, 8.5M uncompressed

text This set includes NetBSD's text processing tools, including `groff`(1), all related programs, and their manual pages.
1.3M gzipped, 4.9M uncompressed

The alpha security distribution set is named **secr** and can be found in the
`alpha/binary/security`
subdirectory of the NetBSD 1.4.1 distribution tree. It contains security-related binaries which depend on cryptographic source code. You do not need this distribution set to use encrypted passwords in your password file; the **base** distribution includes a crypt library which can perform only the one-way encryption function. The security distribution includes a version of the Kerberos IV network security system, and a Kerberized version of `telnet`(1) program. The **secr** distribution set can be found only on those sites which carry the complete NetBSD distribution and which can legally obtain it. Because of United States law, it may not be legal to distribute this set to locations outside of the United States and Canada.
1M gzipped, 3M 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 3.3.3.1. 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.
3.4M gzipped 12.9M uncompressed

xcomp The extra libraries and include files needed to compile X source code.
2.4M gzipped, 13.1M uncompressed

xcontrib Programs that were contributed to X.
228k gzipped, 820k uncompressed

xfont Fonts needed by X.
5.8M gzipped, 6.8M uncompressed

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

The instructions given for extracting the source sets work equally well for the binary sets, but it is worth noting that if you use that method, the 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 xfp`** command from /.

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

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

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

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

All SYSVSUM files are 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/alpha System Requirements and Supported Devices

NetBSD/alpha 1.4.1 runs on most of the DEC Alpha PCI platforms, on all of the TURBOChannel models, on the high end 8200 and 8400 systems, and on the 4100 series.

The SRM console is required. This console can be distinguished from the ARC console (which is used to boot Windows NT) by the fact that it has a command line interface, rather than a menu-driven interface. The SRM prompt is ">>>".

Some platforms have both the SRM console and the ARC console, and can switch between them, and other platforms have only one type of console loaded at any one time. If your system comes up with the ARC firmware, it may be possible to switch it to SRM with a menu or to download SRM from

ftp://gatekeeper.dec.com/pub/Digital/Alpha/firmware/index.html.

You may want to buy a firmware update CD from Compaq Computer Corporation.

More information on supported platforms and devices can be found on the alpha port web pages at *:* **http://www.netbsd.org/**.

A basic system will fit on a 200 MB disk (including swap) without too much difficulty, but you will probably want at least 500 MB of disk to have any level of comfort.

Although it is actually possible to boot and install NetBSD/alpha in only 16 MB of RAM, you will want to have at least 32 MB. We support add-in devices on the PCI, ISA, EISA and TurboChannel buses. Because NetBSD has an extremely machine-independent device driver system, many device drivers are the same as used in other ports that use the same bus. For example, the 'de' network card driver is shared by the i386 and Alpha ports. Some drivers on inspection appear as if they will work on the alpha but have not been tested because that hardware was not available to NetBSD testers; these are marked as *untested* below. If you have one of these devices, and it does work, please get in touch with `port-alpha-maintainer@netbsd.org` and let us know that it works. If it doesn't work, do the same thing and we can probably fix it pretty easily.

Supported devices by bus type are:

- PCI Bus
  - Graphics Adapters
    - VGA-compatible video (pcivga)
    - ZLXp-E1 DECchip 21030-based video, *tga*

      Note ZLXp-E2 and ZLXp-E3 boards are not supported in NetBSD 1.4.1.
  - Network Cards
    - DECchip 21x40-family 10 and 100 Mbps Ethernet (de)
    - DEC DEFPA FDDI (fpa)
    - PCI LANCE Ethernet (le, UNTESTED)
    - Efficient Networks ENI-155p ATM (en, UNTESTED)
    - 3Com 3c59x and 3c90x (except 3c906) 10 and 100 Mbps Ethernet (ep)
    - Intel EtherExpress Pro 10/100B PCI Ethernet (fxp, UNTESTED)
  - SCSI Controllers
    - Adaptec 294x, 394x, AIC-7850, AIC-7860, AIC-7870 and AIC-7880 SCSI (ahc)
    - BusLogic 9xx SCSI (bha, Works on Alpha PC164)
    - Qlogic ISP 10x0-family SCSI (isp)
    - NCR/Symbios 53c8xx-family SCSI (ncr, NCR825 Doesn't always work)
  - Miscellaneous Devices Cyclades Cyclom-Y serial boards (cy, UNTESTED)
    - PCI-PCI bridges (ppb, tested with the DECchip 21050, but should work with all bridges and system firmware revisions that comply with the PCI-PCI bridge specification)
- ISA Bus
  - Network Cards
    - 3Com 3c509 Ethernet (ep)
    - DEC DE200,DE201,DE202 (le)
    - DEC DE203,DE204,DE205 (lc)
  - Miscellaneous Devices
    - PC-style parallel ports (lpt)
    - NS16450 and NS16550 UARTs (com)
    - ISA multi-port 16x50 boards (such as ast, boca--only boca has been tested)
- EISA Bus
  - Network Cards
    - DEC DEFEA FDDI (fea)
    - 3Com 3c5xx series (ed, UNTESTED)
  - SCSI Controllers
    - Adaptec 274x and aic7770 SCSI (ahc, UNTESTED)
    - BusLogic 7xx SCSI (bha, UNTESTED)
- Turbochannel Bus
  - Graphics Adapters
    - CFB video (PMAG-BA, cfb)
    - SFB video (PMAGB-BA, sfb)

    Note Although these boards are supported by NetBSD/alpha, since there is no keyboard or mouse support available for the TurboChannel systems, they aren't very useful.
  - Network Cards
    - DEC LANCE Ethernet (PMAD-AA, le, UNTESTED)
    - DEC DEFTA FDDI (PMAF-F, fta)

Note Note that PC-style floppy disk drives are not supported in 1.3 (except to boot from), but are supported to some degree in NetBSD-current.

Note Note that some devices, especially ISA-based devices, have to have certain settings set properly for the install and GENERIC kernels to detect them. (Once installed, you can always rebuild your own kernel to detect them anywhere you wish, of course.) Here is a list of such devices and the necessary settings:

```
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]

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

AHA-2X4X or AIC-7XXX-based SCSI host adapters
                ahc0    any     any     any

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

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

ATA disks       wd0, wd1, ...
SCSI 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 ...

3COM 3x59X or 3COM 3x90X PCI Ethernet boards
                ep0     any     any     [you must assign an interrupt in your
                                         PCI BIOS, or let it do so for you]

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]

DEC DE200,201,202 EtherWORKS II/Turbo ISA Ethernet boards
                le?    0x300   5      memory at D0000-DFFFF
                le?    0x200   10     memory at D8000-DFFFF

You should enter the following SRM console command to enable the le device:
```

```
>>> isacfg -mk -slot ? -dev 0 -handle DE200-LE -irq0 5
       -membase0 d0000 -memlen0 10000 -iobase0 300 -etyp 1 -enadev 1

DEC DE203,204,205 EtherWORKS III ISA Ethernet boards
                lc0    0x300   any
                lc1    0x320   any

You should enter the following SRM console command to enable the device:

>>> add_de205
```

Getting the NetBSD System on to Useful Media

A large number of different media types can be used to hold the binary distribution sets, but they must originally be obtained from the NetBSD project via a mechanism from the list below.

- FTP
- Remote NFS partition
- CD-ROM

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

NFS Installation 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 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 closest router . Finally, you need to know the numeric IP address of the NetBSD machine itself. The install program will ask you to provide this information to be able to access the sets.

Once the NFS server is set up properly and you have the information mentioned above, you can start the actual installation process.

FTP Installation Determine an FTP site from which you can retrieve the NetBSD distribution when you're about to install. You will need to know the IP address of your nameserver or of your ftp 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 IP address of the router closest to the NetBSD machine. Finally, you need to know the numeric IP address of the NetBSD machine itself. The install program will ask you to provide this information to be able to access the sets via ftp.

Once you have this information, you can proceed to the actual installation.

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 distribution, 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 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

If you have any data on your disks that you want to keep, *back it up* before starting. Note that NetBSD/alpha does not support booting more than one operating system from a single disk, although it's fine to have multiple operating systems on your machine if you have a separate disk for NetBSD, or if one of them uses a network boot.

Installing the NetBSD System

Installation of NetBSD/alpha is now easier than ever! For the latest news, problem reports, and discussion, join the port-alpha mainlist by mailing a line saying

```
subscribe port-alpha
```

to `majordomo@netsbd.org`. Also, see **http://www.netbsd.org** for more information.

If you encounter any problems, please report them via the mailing list or the `send-pr`(1) program so that they can be fixed for the next release.

To install or upgrade NetBSD, you need to first boot an installation program and then interact with the screen-menu program **`sysinst`**. The installation program actually consists of the NetBSD kernel plus an in-memory file system of utility programs.

The usual procedure is to write the installation system to a floppy disk set and then boot from the floppies, however, there are now six ways to boot the NetBSD/alpha installation system! Each approach loads the exact same installation bits. The six paths are:

1. Floppy disk boot
2. CD boot
3. Hard Drive Boot
4. Magnetic Tape Boot
5. Existing Root FS Boot
6. Network boot

In all cases, you need to transfer a bootable image of the installation system from the NetBSD CD or from an ftp site to the chosen media type. Although booting from floppy is the usual path, the hard drive boot is useful if you have another operating system (and a spare drive) already installed, or if you don't mind swapping hard drives from box to box. CD and tape boots are nice and fast if you have a CD writer or a tape format in common with another previously installed Unix-like system. Finally, most versions of SRM can locate the NetBSD boot program **`netboot`** via bootp and download it via tftp, **`netboot`** then mounts the root file system via NFS and loads the kernel.

1. Floppy disk boot

   The 3.5", 1.44 MB boot floppy set is found under the NetBSD/alpha 1.4.1 distribution directory in `alpha/installation/floppy/` as two files called `disk1of2` and `disk2of2`. You need to put these two disk images on two floppy disks.

If you have a Unix system handy, you can do this with commands like the following:

```
dd if=disk1of2 of=/dev/rfd0a bs=18k
dd if=disk2of2 of=/dev/rfd0a bs=18k
```

If the Unix system you are using is not a NetBSD system, you will probably need to replace `/dev/rfd0a` with the name of the floppy device on your particular system.

If you have an MS-DOS or Windows system available, you can use the **`rawrite.exe`** utility to transfer the image to a floppy disk. This utility is provided with the NetBSD/i386 install tools, under `i386/installation/misc`; a documentation file, **`rawrite.doc`** is available there as well.

Once the floppy has been made, you simply need to put it in the drive and type

```
>>> B DVA0
```

2. CD boot
3. Hard Drive boot
4. Magnetic Tape Boot

All three of these media types use the same initial image:

```
.../installation/diskimage/cdhdtape
```

The image can be written to a hard drive partition with a command like:

```
dd if=cdhdtape bs=16k of=/dev/rsd0c
```

To boot from a magnetic tape device such as DAT or DLT, it is important to create the tape image with 512-byte records. Use a command like:

```
dd if=cdhdtape bs=512 of=/dev/rst0
```

If the host system is not NetBSD, the names of the destination devices are likely to be different. Be sure to use a ''raw partition'' device that doesn't skip over labels!

The use of CD-R devices varies greatly depending on the host OS and host software; it isn't possible to give typical instructions here.

5. Existing Root FS Boot

The installation subdirectory `instkernel/` contains `netbsd.gz`; this is the same install kernel but without a bootable filesystem image wrapped around it. You can perform an complete reinstall by beginning it as an upgrade, and booting this kernel in the normal way off the root file system of a previous installation.

The gzipped image can be booted directly; it is not necessary to uncompress it first.

6. Network Boot

Booting NetBSD/alpha 1.4.1 over a network requires a BOOTP or DHCP server, a TFTP server and an NFS server. (These are usually all run on the same machine.) There are three basic stages to the boot:

- Alpha console software sends a BOOTP request to get its own address, the address of the TFTP server and the file to download. It downloads this file, which is the second stage bootstrap, via TFTP and then executes it.
- The secondary boot program resends the BOOTP request, this time also locating the NFS server and root path. It mounts the root path via NFS and reads in and transfers to the kernel: `/netbsd`.
- The kernel probes and configures the devices, and then sends out another BOOTP request so it can find out its address, the NFS server, and path. It then mounts its root via NFS and continues.

You will need to set up servers for BOOTP, TFTP and NFS.

If you want to run a full system from the network, untar the NetBSD snapshot or distribution into a directory on your server and NFS export that directory to the client. Make sure you put a kernel there as well, and create the device nodes in /dev with **sh ./MAKEDEV all**. Detailed instructions on netbooting can be found by visiting the NetBSD Alpha platform page:

http://www.NetBSD.org/Ports/alpha

At the time of this release, the URL for the netbooting instructions is:

http://www.netbsd.org/Ports/alpha/netboot.html

You'll want to map root to root (rather than the default nobody) when you export your root filesystem. A typical /etc/exports line on a NetBSD system would be:

```
/usr/export/alpha -maproot=0 myclient.mydom.com
```

One option is to load just the install kernel over the network but then proceed to a normal disk-based install and disk-based operation. (Perhaps the Alpha doesn't have a floppy drive, or you just don't want to use a Micro$oft Window$ system to make the floppy; we understand.)

For this case, you still need to export an NFS root, but the only thing it needs to have in it is the instkernel image from the distribution.

The gzipped image can be booted directly; it is not necessary to uncompress it first.

The console will be using TFTP to load the NetBSD boot program, so for the TFTP setup, you need to copy the second stage bootstrap, **netboot**, into an appropriately named file such as boot.netbsd.alpha in the directory used by your TFTP server. If you extracted a full snapshot, you can get the netboot program from /usr/mdec/netboot; if not, you can get this from the installation/netboot directory where you found the alpha distribution.

For the BOOTP server you need to specify the:

- hardware type (Ethernet)
- hardware address (Ethernet MAC address)
- IP address of the client
- subnet mask of the client
- address of of the TFTP/NFS server
- name of the second stage bootstrap loaded via TFTP
- path to the root for the client (mounted via NFS)

Here's an example for a Unix system running bootpd:

```
myhost.mydom.com:       :ht=ethernet:ha=0000c0391ae4:\
          :ip=192.168.1.2:sm=255.255.255.0:\
          :sa=192.168.1.1:bf=boot.netbsd.alpha:rp=/usr/export/alpha:
```

And here's an example for a Unix system running the ISC dhcpd:

```
host axp {
        hardware ethernet 0:0:c0:39:1a:e4;
        fixed-address 192.168.1.2;
        option host-name "myhost.mydom.com";
        filename "boot.netbsd.alpha";
        option root-path "/usr/export/alpha";
        option domain-name-servers 192.168.1.1;
        option broadcast-address 255.255.255.0;
        option domain-name "my.domain";
}
```

The only Ethernet device the console on most Alpha systems knows how to boot from is the onboard Ethernet interface or a DEC Tulip (21040, 21041, 21140) PCI Ethernet card. Some older SMC 100 Mbps cards that use this chip have been known to work as well. Many older systems will not be able to use the newer 2.0 stepping of the 21140, however. If your system appears not to be receiving packets, this may be the problem. (You may or may not be able to update your firmware to fix this; see the alpha port pages on www.netbsd.org for more information on this.) In general, 10 Mb cards from manufacturers other than DEC will work, and 100 Mb cards not from DEC will not.

Once you're set up, you should be able to boot with:

```
>>> boot -proto bootp ewa0
```

You should permanently set your protocol to BOOTP with:

```
>>> set ewa0_protocols bootp
```

The 3000 series of Turbochannel systems and certain other models use *old SRM*, do not have a **-proto** option and use different device names. They also tend to not netboot very well so you probably don't need to worry about this section. However, if you want to give it a try, note the following differences:

- There is no *-proto* argument, or **ewa0_protocols** variable. Old SRM uses bootp if the device name is given as **ez0**.
- The use of the `setnetbootinfo`(8) program will probably also be necessary, as it is unlikely that an SRM from that era will properly communicate the ethernet HW address to the boot program.
- Example:

  ```
  >>> boot ez0
  ```

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*

   There is a serious bug that may make installation of NetBSD on PCMCIA machines difficult. This bug does not make *use* of PCMCIA difficult once a machine is installed. 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.

   This will explains how to work around the installation problem.

   What is the bug: 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 ports. Unfortunately, not all devices are included in the `INSTALL` kernels in order to save space. Let's say 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. This is especially bad if one of the devices in question is your ethernet card.

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

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 hit 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 floppy disks. One 1.44M 3.5" floppy.
     - A CD-ROM drive (SCSI or ATAPI), a harddisk and a minimum of 32Mb of memory installed.
     - The harddisk should have at least 200 + *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 60Mb more.
   - The Quick Installation
     - Insert the first boot floppy you just created. Boot the computer. Type
       `>>> B DVA0`
       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. 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 using the boot floppy. 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.

   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. 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.

   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`.

   Next, depending on whether you are using a wd*x* or wd*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* 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**.

  Some partitions in the disklabel have a fixed purpose. Partition **a** is always the root partition, **b** is the swap partition, and **c** is the whole disk. Partitions **e-h** are available for other use. Traditionally, **d** is the partition mounted on the `/usr` directory, but this is historical practice, 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 filesystems 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.

10. *Getting the distribution sets.*

    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 9

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 an unmounted filesystem*

    In order to install from a local filesystem, you will need to specify the device that the filesystem resides on (for example `wd1e`) the type of the filesystem, and the directory on the specified filesystem where the sets are located. **sysinst** will then check if it can indeed access the sets at that location.

15. *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 filesystem that is already accessible. **sysinst** will ask you for the name of this directory.

16. *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.

17. *Finalizing your installation.*

    Congratulations, you have successfully installed NetBSD 1.4.1. You can now reboot the machine, and boot from harddisk.

Manual and Script-Assisted Installation

All of the installation procedures consist of putting a label on the disk to provide information on the sizes and placement of the partitions into which the disk is divided, putting the boot blocks on the disk, creating the filesystems on the partitions, and unpacking the distribution tar archives.

1. Disk prep: label, boot block, and file system setup

   A. Manual Install from the Shell Prompt

      The normal installation involves running the install shell script and interactively configuring the file systems, and then simply unpacking the tar files into these followed by running MAKEDEV.

      However, as stated above it is also possible to do the installation yourself from the shell, and in any case it is helpful to understand what the install script does. The procedure is:

      - create `/etc/disktab`, see `disktab`(5)
      - run `disklabel`(8),
      - run `newfs`(8)
      - `mount`(8) the new root on `/mnt`
      - cd to `/usr/mdec` and run `installboot`(8)

      If you are reviewing man pages on NetBSD platforms other than alpha, be sure that when reading `installboot`(8) you read the alpha version by typing: "man 8 alpha/installboot".

      At this point you need only unpack the distribution sets by running `tar`(1) as described below.

   B. `/install` and `/upgrade` traditional installation scripts

      The `install` and `upgrade` scripts are still there, so by exiting the sysinst program you can type `install` or `upgrade` at the shell prompt and run them as you did in the good old days.

      You may install on either a SCSI or an IDE disk; you will be prompted for the disk to install on. The disks in your system will be numbered starting at xd0 (where x is an 's' for SCSI disks, 'w' for IDE disks) based on the SCSI ID or IDE drive order; if you have more than one disk, watch the boot messages carefully to see which ones are probed as which numbers.

      Once you've selected a disk to install on, you'll be prompted for the geometry. This is also displayed in the boot messages, and you'll be given a chance to review the boot messages again to get the exact figures for the number of cylinders, heads and sectors.

      After this you must specify the size of your partitions. Generally you'll be giving the sizes in cylinders; the install program will tell you how many bytes there are in each cylinder.

      The swap partition is the second thing you specify, after the root partition. Regardless of the size of your disk, you'll want to specify a swap partition that's at least as large as the amount of RAM you have, and probably not less than 64 MB in any case.

      If you have a small disk (under 500 MB), it's probably best to devote all of the disk (excepting 64 MB or more for the swap) to the root partition.

      If you have more space, we recommend devoting at least 32 MB, and preferably 48 MB, to the root partition. `/usr` will need 150 MB or so if you're not installing X, 200 MB or so if you are. A typical organization is 50 MB for root, 150-250 MB for swap, and the remaining space for `/usr`. With enough swap space configured, you can make `/tmp` a nice, fast mfs. See `mount_mfs`(8), and note that the mfs will require swap space for the largest planned amount of `/tmp` storage. It doesn't return space when files are deleted, but just keeps it its own freelist so the swap space required is equal to the highwater mark of `/tmp` use, plus space required to back up main memory and store inactive images.

Once you've specified this information, the install script will write the disklabel, install boot blocks to make the disk bootable, initialise the filesystems, and mount them all under `/mnt`. You are now ready to go on to the next step.

2. Configuration: arranging access to the distribution sets

   After doing the disk and file system setup with either shell commands or the script assist, you then need only unpack the distribution sets with the `tar`(1) command. To do this you will need access from the target host to the tar files that contain the operating system in order to extract them to your disk. This is done via an NFS or FTP transfer over a network, via a CD-ROM archive, a tape archive, or by preloading an accessible hard drive with the necesary tar files.

   - Preparing to Install from a CD-ROM

     All you need to do is mount the CD-ROM, which will generally be device cd0. (The initial boot messages will tell you what the CD-ROM drive is probed as.) This would be done with:

     ```
     mount -r -t cd9660 /dev/cd0a /mnt2
     ```

   - Preparing to Install from the Network

     The first thing you need to do is configure the loopback network interface, which is done with the command

     ```
     ifconfig lo0 127.0.0.1
     ```

     Then you will have to configure your Ethernet card. The command

     ```
     ifconfig -l
     ```

     will give you a list of the network interfaces on your system. It will show you your ethernet cards first, followed by lo0 (the loopback interface that we configured above), ppp0 (the PPP interface) and sl0 (the SLIP interface).

     To configure your ethernet card, type **ifconfig** *if* inet *addr* [netmask] [media *media*] where *if* is the network card (interface), almost always `de0`, *addr* is the IP address, the optional *netmask* parameter is the network mask, and the optional *media* parameter is one of:

10base2	BNC connector	10 Mbps
AUI	AUI connector	10 Mbps
10baseT/UTP	Twisted pair connector	10 Mbps
100baseTX	Twisted pair connector	100 Mbps
100baseFX	Fibre-optic connector	100 Mbps
100baseT4	T4 twisted pair interface	100 Mbps

     If the host you are getting the data files from is not on the local network, you will also have to configure a gateway into your system. Do this with

     ```
     route add default <gateway-IP-address>
     ```

     One improvement over the good old days is that the resolver is now present; by configuring `/etc/resolv.conf` you can get name resolution during any install NFS or FTP operations.

     Once networking has been configured, you may mount the directory with the install files via NFS, or download them via FTP.

     To mount them via nfs, type

     ```
     mount -t nfs <hostname:/path/to/nfs/volume> /mnt2
     ```

     If this volume has been exported read-only, you may need the '-r' option to mount.

     To download the install sets with ftp, create a directory in which to put them and then use the ftp client to download them. Mirror sites are listed at: **http://www.netbsd.org/Sites/net.html** A typical session might be:

```
mkdir /mnt/usr/release
cd /mnt/usr/release
ftp ftp.netbsd.org
```

[the following commmands are given to the ftp program after logging in]

```
prompt
cd /pub/NetBSD/NetBSD-1.4.1/alpha/binary/sets
mget *
bye
```

Feel free, of course, to leave off the sets that you don't need if you don't plan to install everything.

3. Unpack distribution sets: Extracting the Operating System Files

   Change to the root directory of your hard drive (which is `/mnt` if you've used the standard install script to this point) by typing

   ```
   cd /mnt
   ```

   For this and the following commands, replace `/mnt/usr/release/` with the path to your NFS volume or CD-ROM if that's how you chose to access your install files instead.

   The sets and kernel are extracted with

   ```
   cd /mnt
   for i in base kern comp etc games man misc text; do
       tar xpzf /mnt/usr/release/$i.tgz;
   done
   ```

   or perhaps:

   ```
   cd /mnt
   for i in /mnt/usr/release/*.tgz; do
       echo $i
       tar xpzf $i
   done
   ```

   Now make the device nodes:

   ```
   cd /mnt/dev
   sh ./MAKEDEV all
   ```

4. Restart your system

   Unmount the file systems and halt. The exact instructions to type here will depend on the file systems you created, but typically the commands are:

   ```
   cd /
   umount /mnt/usr
   umount /mnt
   sync           # not needed but traditional
   halt
   ```

   You should now be at the SRM console's >>> prompt and can reboot into the new configuration (possibly after an optional power cycle) with a command such as:

   ```
   boot dka0
   ```

   This command might be: **`boot dka100`** if your drive is on ID 1. You can usually use **`show device`** to see a full list of bootable devices in your system. Your system will come up in single-user mode, ready for you to configure it.

You can create the floppy needed for installation under DOS or Windows. Supposing your 1.44M floppy drive is drive A:, and your CD is drive **`E:`**, do the following from an MS-DOS command prompt:

```
e:
cd \NetBSD-1.4.1\installation\misc
rawrite
```

When asked for a source filename, answer

```
...\floppy\disk1of2
```

When asked for a destination drive answer

```
a
```

(Repeat the procedure for `installation/floppy/disk2of2`.)

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 propperly configured state, with the most important ones described below.

1. Configuring `/etc/rc.conf`

   If you 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 filesystem mounted read-write. When the system asks you to choose a shell, simply hit 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 hit 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. If your `/usr` directory is on a separate partition and you do not know how to use 'ed' or 'ex', you will have to mount your `/usr` partition to gain access to 'vi'. Do the following:

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

   If you have `/var` on a seperate 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 are new to NetBSD 1.4 and 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.

3. Adding accounts

Use the `vipw`(8) command to add accounts to your system, *do not* edit `/etc/passwd` directly. See `adduser`(8) for more information on the process of 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.

   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 3rd party packages

   There is a lot of software freely available for Unix-based systems, almost all of which can run on NetBSD. Modifications are usually needed to when transferring programs between different Unix-like systems, so the NetBSD packages collection incorporates any such changes necessary to make that software run on NetBSD, and makes the installation (and deinstallation) of the software packages easy. There's also the option of building a package from source, in case there's no precompiled binary available.

   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
   and extracting it into `/usr/pkgsrc`. See `/usr/pkgsrc/README` then for more information.

6. Misc

   - To adjust the system to your local timezone, point the `/etc/localtime` symlink to the appropriate file under `/usr/share/zoneinfo`.
   - Edit `/etc/aliases` to forward root mail to the right place (run `newaliases`(1) afterwards.)
   - The `/etc/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 *filename*
     is likely to give you more information on these files.

Upgrading a previously-installed NetBSD System

The upgrade to NetBSD 1.4.1 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 set 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. filesystems are checked before unpacking the sets.

After a new kernel has been copied to your hard disk, your machine is a complete NetBSD 1.4.1 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 systems' 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.

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 1.4.1

Note Only issues effecting an upgrade from NetBSD 1.3 or NetBSD 1.3.x are decribed here.

- "machine" directory/link in "/usr/include"

  Description

  Some architecture may fail to install the **comp** set because the

  `/usr/include/machine`

  directory changed to a symbolic link in NetBSD 1.4.

  Fix If this happens, you can use the command

  # **rm -r /usr/include/machine**

  to remove the old directory and it contents and reinstall the **comp** set.

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),
- `apropros`(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)m 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. 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 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 and SUP server.

- Redback Networks, Inc. for hosting the NetBSD Mail server.
- 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 organiztions (each in alphabetical order) have made donations or loans of hardware and/or money, to support NetBSD development, and deserve credit for it:

  Jason Birnschein
  Jason Brazile
  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
  Greg Gingerich
  Guenther Grau
  Ross Harvey
  Charles M. Hannum
  Michael L. Hitch
  Jordan K. Hubbard
  Scott Kaplan
  Noah M. Keiserman
  Chris Legrow
  Neil J. McRae
  Perry E. Metzger
  Herb Peyerl
  Mike Price
  Thor Lancelot Simon
  Bill Sommerfeld
  Paul Southworth
  Ted Spradley
  Kimmo Suominen
  Jason R. Thorpe
  Steve Wadlow

  Advanced System Products, Inc.
  Avalon Computer Systems
  Bay Area Internet Solutions
  Canada Connect Corporation
  Demon Internet, UK
  Digital Equipment Corporation

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.
VMC Harald Frank, Germany

(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:

Paul Kranenburg	`pk@NetBSD.ORG`
Scott Reynolds	`scottr@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**
Ignatios Souvatzis	`is@NetBSD.ORG`	**amiga**
Eduardo Horvath	`eeh@NetBSD.ORG`	**sparc64**
Paul Kranenburg	`pk@NetBSD.ORG`	**sparc**
Anders Magnusson	`ragge@NetBSD.ORG`	**vax**
Tsubai Masanari	`tsubai@NetBSD.ORG`	**macppc**
Tsubai Masanari	`tsubai@NetBSD.ORG`	**newsmips**
Minoura Makoto	`minoura@NetBSD.ORG`	**x68k**
Phil Nelson	`phil@NetBSD.ORG`	**pc532**
Scott Reynolds	`scottr@NetBSD.ORG`	**mac68k**
Darrin Jewell	`dbj@NetBSD.ORG`	**next68k**
Gordon Ross	`gwr@NetBSD.ORG`	**sun3, sun3x**
Kazuki Sakamoto	`sakamoto@NetBSD.ORG`	**bebox**
Wolfgang Solfrank	`ws@NetBSD.ORG`	**powerpc**
Jonathan Stone	`jonathan@NetBSD.ORG`	**pmax**
Jason Thorpe	`thorpej@NetBSD.ORG`	**hp300**
Frank van der Linden	`fvdl@NetBSD.ORG`	**i386**
Leo Weppelman	`leo@NetBSD.ORG`	**atari**
Steve Woodford	`scw@NetBSD.ORG`	**mvme68k**

The NetBSD 1.4.1 Release Engineering team:

Ted Lemon	mellon@NetBSD.ORG
Perry Metzger	perry@NetBSD.ORG
Curt Sampson	cjs@NetBSD.ORG

Developers and other contributors:

Steve Allen	wormey@NetBSD.ORG
Lennart Augustsson	augustss@NetBSD.ORG
Christoph Badura	bad@NetBSD.ORG
Manuel Bouyer	bouyer@NetBSD.ORG
Robert V. Baron	rvb@NetBSD.ORG
John Brezak	brezak@NetBSD.ORG
Allen Briggs	briggs@NetBSD.ORG
Aaron Brown	abrown@NetBSD.ORG
David Brownlee	abs@NetBSD.ORG
Simon Burge	simonb@NetBSD.ORG
Dave Burgess	burgess@cynjut.infonet.net
Dave Carrel	carrel@NetBSD.ORG
Bill Coldwell	billc@NetBSD.ORG
Chuck Cranor	chuck@NetBSD.ORG
Alistair Crooks	agc@NetBSD.ORG
Aidan Cully	aidan@NetBSD.ORG
Rob Deker	deker@NetBSD.ORG
Chris G. Demetriou	cgd@NetBSD.ORG
Matthias Drochner	drochner@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
Brian R. Gaeke	brg@dgate.org
Thomas Gerner	thomas@NetBSD.ORG
Justin Gibbs	gibbs@NetBSD.ORG
Adam Glass	glass@NetBSD.ORG
Michael Graff	explorer@NetBSD.ORG
Brad Grantham	grantham@tenon.com
Matthew Green	mrg@NetBSD.ORG
Juergen Hannken-Illjes	hannken@NetBSD.ORG
Charles M. Hannum	mycroft@NetBSD.ORG
Eric Haszlakiewicz	erh@NetBSD.ORG
Michael L. Hitch	osymh@NetBSD.ORG
Ken Hornstein	kenh@NetBSD.ORG
Marc Horowitz	marc@NetBSD.ORG
ITOH Yasufumi	itohy@NetBSD.ORG
Matthew Jacob	mjacob@NetBSD.ORG
Lonhyn T. Jasinskyj	lonhyn@NetBSD.ORG
Darrin Jewell	dbj@NetBSD.ORG

Lawrence Kesteloot	kesteloo@cs.unc.edu
Klaus Klein	kleink@NetBSD.ORG
John Kohl	jtk@NetBSD.ORG
Kevin Lahey	kml@NetBSD.ORG
Ted Lemon	mellon@NetBSD.ORG
Mike Long	mikel@NetBSD.ORG
Paul Mackerras	paulus@NetBSD.ORG
Neil J. McRae	neil@NetBSD.ORG
Perry Metzger	perry@NetBSD.ORG
Luke Mewburn	lukem@NetBSD.ORG
der Mouse	mouse@NetBSD.ORG
Tohru Nishimura	nisimura@NetBSD.ORG
Masaru Oki	oki@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
Darren Reed	darrenr@NetBSD.ORG
Tim Rightnour	garbled@NetBSD.ORG
Heiko W. Rupp	hwr@NetBSD.ORG
SAITOH Masanobu	msaitoh@NetBSD.ORG
Kazuki Sakamoto	sakamoto@NetBSD.ORG
Curt Sampson	cjs@NetBSD.ORG
Wilfredo Sanchez	wsanchez@NetBSD.ORG
Ty Sarna	tsarna@NetBSD.ORG
Matthias Scheler	tron@NetBSD.ORG
Karl Schilke (rAT)	rat@NetBSD.ORG
Tim Shepard	shep@NetBSD.ORG
Chuck Silvers	chs@NetBSD.ORG
Thor Lancelot Simon	tls@NetBSD.ORG
Noriyuki Soda	soda@NetBSD.ORG
Wolfgang Solfrank	ws@NetBSD.ORG
Bill Sommerfeld	sommerfeld@NetBSD.ORG
Ignatios Souvatzis	is@NetBSD.ORG
Bill Studenmund	wrstuden@NetBSD.ORG
Kevin Sullivan	sullivan@NetBSD.ORG
Kimmo Suominen	kim@NetBSD.ORG
Matt Thomas	matt@NetBSD.ORG
Jason Thorpe	thorpej@NetBSD.ORG
Christoph Toshok	toshok@NetBSD.ORG
Todd Vierling	tv@NetBSD.ORG
Paul Vixie	vixie@NetBSD.ORG
Krister Walfridsson	kristerw@NetBSD.ORG
Nathan Williams	nathanw@NetBSD.ORG
Colin Wood	ender@NetBSD.ORG

Legal Mumbo-Jumbo

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 Computer Systems Engineering Group at Lawrence Berkeley Laboratory.

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

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

This product includes software developed by Adam Glass.

This product includes software developed by Berkeley Software Design, Inc.

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 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 Herb Peyerl.

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 Mark Brinicombe.

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 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 Scott Bartram.

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

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 TooLs GmbH.

This product includes software developed by Winning Strategies, Inc.

This product includes software developed by the Center for Software Science 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 Charles Hannum.

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 the University of Vermont and State Agricultural College and Garrett A. Wollman.

This product includes software developed by Dean Huxley.

This product includes software developed by Herb Peyerl.

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.