NAME

INSTALL – Installation procedure for NetBSD/mvme68k.

CONTENTS

About this Document....2
What is NetBSD?....2
Changes Between The NetBSD 2.0.1 and 2.0.2 updates....3
  Kernel....3
  Networking....3
  System administration and user tools....3
  mvme68k specific....4
Changes Between The NetBSD 2.0 release and 2.0.1 update....4
  Kernel....4
  Networking....4
  System administration and user tools....4
  mvme68k specific....4
Changes Between The NetBSD 1.6 and 2.0 Releases....4
  Kernel....4
  Networking....5
  File system....5
  Security....5
  System administration and user tools....6
  Miscellaneous....6
  mvme68k specific....6
Important notes about NetBSD 2.0.2....6
The Future of NetBSD....7
Sources of NetBSD....7
NetBSD 2.0.2 Release Contents....7
  NetBSD/mvme68k subdirectory structure....9
  Binary distribution sets....9
NetBSD/mvme68k System Requirements and Supported Devices....12
  Supported VME147 hardware....12
  Supported VME162/VME172 hardware....12
  Supported VME167/VME177 hardware....13
Getting the NetBSD System on to Useful Media....13
  Creating boot/install tapes....14
  Boot/Install from NFS server....14
  Install/Upgrade from CD-ROM....15
  Install/Upgrade via FTP....15
Preparing your System for NetBSD installation....15
Installing the NetBSD System....17
  Installing from tape....17
  Installing from NFS....21
  Booting the miniroot....23
  Miniroot install program:....24
  Running the sysinst installation program....25
    Introduction....25
    General....25
    Quick install....25
    Booting NetBSD....26

Network configuration....26
Installation drive selection and parameters....27
Partitioning the disk....27
Preparing your hard disk....27
Getting the distribution sets....27
Installation using ftp....28
Installation using NFS....28
Installation from CD-ROM....28
Installation from an unmounted file system....28
Installation from a local directory....28
Extracting the distribution sets....29
Finalizing your installation....29
Post installation steps....29
Upgrading a previously-installed NetBSD System....31
Upgrading using the miniroot....31
Manual upgrade....32
Compatibility Issues With Previous NetBSD Releases....33
Issues affecting an upgrade from NetBSD 1.6....33
Using online NetBSD documentation....33
Administrivia....34
Thanks go to....34
We are....37
Legal Mumbo-Jumbo....44
The End....48

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 2.0.2 on the *mvme68k* 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 fifty four different system architectures (ports), featuring seventeen machine architectures across fifteen distinct CPU families, and is being ported to more. The NetBSD 2.0.2 release contains complete binary releases for many different system architectures. (A few ports are not fully supported at this time and are thus not part of the binary distribution. 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 Between The NetBSD 2.0.1 and 2.0.2 updates

The NetBSD 2.0.2 release is the second security/critical update of the NetBSD 2.0 release branch. This represents a selected subset of fixes deemed critical in nature for stability or security reasons. All of these fixes will also appear on future NetBSD releases (e.g. NetBSD 2.1 , etc) from the NetBSD 2.0 release branch as well.

Specific updates are as follows:

Kernel

- Avoid endless loop in F_CLOSEM fnctl which could be abused for a local DOS.
- Avoid possible security issues on amd64, x86 and xen by properly checking range for copyinstr and copyoutstr.
- Make pmap_map() work on sun4 machines so these are usable again.
- Fix a UVM problem that causes hangs when large processes fork.
- Fix pthreads on sun4c hardware by fixing a problem with detecting faults in atomic load/store instructions.
- Address further pthread issues on sparc and sparc64 by defining a new LWP flag which indicates that we're in the process of doing a context switch.
- Make sun3 port functional again by restoring the definition for MAXPHYS as it is now used rather than MAXBSIZE to limit page cache I/O sizes.
- Prevent lockups/panics if the VOP_LOOKUP() call unlocked the parent directory node.
- Prevent a kernel panic on boot with a PX graphics card on DECstations.
- Fix prep port to be functional by ensuring that the OpenPIC register window is mapped during startup.
- Clear freed memory in `cgd`(4) code to avoid possibly security issues.

Networking

- Disabled the oow test in ipf because it is killing valid packets.
- Prevent deadlock/panic on NFS clients after NFS server reboots and caches aren't in sync.
- Fix an NFS panic caused by truncating a file while another client is writing data to it.
- Avoid infinite loops when getting NFS readdir response without any entries or EOF.
- Fix possible remote DOS via IPSec AH packets.

System administration and user tools

- Teach groff about NetBSD versions that aren't on the branches documentation.
- Add description for Solaris 10 dual boot installation.
- Document PTHREAD_CONCURRENCY.
- Fix bug in binutils which broke Firefox under NetBSD-sparc64.
- Address xpm security problems reported in CAN-2005-0605.

mvme68k specific

Changes Between The NetBSD 2.0 release and 2.0.1 update

The NetBSD 2.0.1 release is the first security/critical update of the NetBSD 2.0 release branch. This represents a selected subset of fixes deemed critical in nature for stability or security reasons. All of these fixes will also appear on future NetBSD releases (e.g. NetBSD 2.1 , etc) from the NetBSD 2.0 release branch as well.

Specific updates are as follows:

Kernel

- Prevent panics on powerpc with DIAGNOSTIC kernels and trap handling.
- Make the macppc INSTALL kernel bootable again by moving the load address to the correct location.
- Fix a major issue with sparc64 pmap to prevent crashes under heavy load.
- Fix major performance issue with xen port so idle loop doesn't consume 100% of cpu time.
- Fix xen port so it can reboot cleanly instead of hanging.
- MMU fixes for sh3 based ports to prevent reboots under heavy load.

Networking

- Fix major performance issues with the i82547 Gig-E chip which improves performance with `wm`(4)
- Fix problems with `sk`(4) performance on ASUS A8V motherboards.
- Prevent panics in ipf when receiving IPv6 packets.
- NFS fixes to address incorrect atimes updates and cache coherancy issues.

System administration and user tools

- Fix `swapcontext`(3) for amd64 so that it works correctly.

mvme68k specific

Changes Between The NetBSD 1.6 and 2.0 Releases

The NetBSD 2.0 release provides numerous significant functional enhancements, including support for many new devices, integration of hundreds of bug fixes, new and updated kernel subsystems, and many user-land enhancements. The result of these improvements is a stable operating system fit for production use that rivals most commercially available systems.

It is impossible to completely summarize over two years of development that went into the NetBSD 2.0.2 release. Some highlights include:

Kernel

- Ports to new platforms including: amd64, evbsh5, and xen.
- Native thread support has been added, based on Scheduler Activations. Applications which support native threads can now take full advantage of the high-performance NetBSD POSIX threads implementation.
- Kernel events notification framework - kqueue kqueue provides a stateful and efficient event notification framework. Currently supported events include socket, file, directory, fifo, pipe, tty and device changes, and monitoring of processes and signals. kqueue is supported by all writable filesystems in the NetBSD tree (with the exception of Coda) and all device drivers supporting poll(2).

- Improvements have been made to NetBSD's Linux emulation to support the latest Sun JDK/JRE for Linux. Testing has shown that it now runs as well as it does on Linux natively.
- NetBSD 2.0 enforces non-executable mappings on many platforms. This means that parts of the stack and heap are made non-executable when they are marked writable. This makes exploiting potential buffer overflows harder.
- For the i386 port it now supports SMP and has a new ACPI and power management framework which takes advantage of Intel's ACPI implementation.
- The macppc port also now supports SMP as well as hardware support for newer G4 models has been added.
- SMP support has been added to the sparc port as well.
- New generic i2c framework, supporting bit-bang and "intelligent" controllers.
- `sysctl`(9) was switched from a static binding to a dynamic implementation.
- Add a new driver `satalink`(4) and move SATA support from other controllers into this along with adding support for new controllers.

Networking

- `ipf`(8) has been upgraded to version 4.1.3.
- `tcp`(4) now implements path MTU discovery blackhole detection (i.e. it will turn off path MTU discovery if the connection is losing).
- Socket buffer insertion is now O(C). This can provide a substantial performance boost to some applications which use large socket buffers.
- `wi`(4) has support for Host-AP mode, allowing Intersil Prism2/2.5/3-based boards to be used to make an 802.11 Access Point.
- Support for `ipf`(8) has been added to `bridge`(4) and `brconfig`(8)
- Change port allocation from linked list to a hash table for better performance.

File system

- FreeBSD's UFS2 has been ported to NetBSD. UFS2 is an extension to FFS, adding 64 bit block pointers and support for extended file storage. Among other enhancements, UFS2 allows for file systems larger than 1Terabyte.
- The cryptographic disk driver (cgd) can be used to encrypt disks or partitions, using some strong encryption algorithms, like AES (Rijndael) and Blowfish. cgd can also be configured to encrypt swap.

Security

- The systrace utility has been added to the system. systrace monitors and controls an application access to the system by enforcing access policies for system calls. The systrace utility might be used to trace an untrusted application's access to the system. In addition, it can be used to protect the system from software bugs (such as buffer overflows) by constraining a daemon's access to the system. The privilege elevation feature of systrace can be used to obviate the need to run large, untrusted programs as root when only one or two system calls require the elevated privilege.
- Verified Exec support has been added in this release. Verified Exec verifies a cryptographic hash before allowing execution of binaries and scripts. This can be used to prevent a system from running binaries or scripts which have been illegally modified or installed. In addition, Verified Exec can also be used to limit the use of script interpreters to authorized scripts only and disallow interactive use.

System administration and user tools

- Switched from the GPL version to a non-GPL version of various tools including gzip and awk.

Miscellaneous

- NetBSD 2.0 supports a new toolchain based on gcc 3.3.3 and binutils 2.14. gcc 3.3.1 adds support for a number of CPU targets and greatly improved support for i386 and other targets. The support for new platforms in gcc 3.3.3 has enabled the porting of NetBSD to even more architectures.
- Updates of most third party packages that are shipped in the base system to the following latest stable releases:
  - bind 8.3.7
  - binutils 2.14
  - cvs 1.11.17
  - diffutils 2.8.1
  - file 4.08
  - gcc 3.3.3
  - gdb 5.3
  - grep 2.5.1
  - groff 1.19
  - less 381
  - openssl 0.9.7d
  - postfix 2.0.19
  - sendmail 8.12.11
  - tcpump 3.7.1
  - texinfo 4.6
- Many new packages in the *pkgsrc* system, including the latest open source desktop KDE3, OpenOffice, perl, Apache and many more. A number of new platforms are supported, including Darwin, FreeBSD, IRIX, Linux, OpenBSD and Solaris. Support for various other platforms (among them AIX, BSD/OS and HP-UX) is currently being worked on thanks to our new, portable bootstrap kit which makes it much simpler to port pkgsrc support to new operating systems. At the time of writing, there are over 4500 third party packages available in pkgsrc.

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.

mvme68k specific

This is the sixth major release of NetBSD for the mvme68k series of boards.

Important notes about NetBSD 2.0.2

- As noted above, the sendmail third-party program has been upgraded since the last release. The new version of sendmail runs without requiring being set-user-ID to "root".

  In order to retain the functionality that a newly installed system can send mail "out of the box", the default has changed so that sendmail will now start by default, and listen for host-local connections.

  If this behaviour is not desired, you can either
  1. explicitly set the sendmail variable to "NO" in `/etc/rc.conf`,
  2. modify `/etc/mail/submit.cf` to point to another host,
  3. modify `/etc/mailer.conf` to point to something else than sendmail, or
  4. make sendmail set-user-ID to "root", remove `/etc/mail/submit.cf`, and set the sendmail_suid variable to "YES" in `/etc/rc.conf`.

  The latter method is strongly discouraged.

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.

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 2.0.2 Release Contents

The root directory of the NetBSD 2.0.2 release is organized as follows:

`.../NetBSD-2.0.2/`

`CHANGES`	Changes since earlier NetBSD releases.
`LAST_MINUTE`	Last minute changes.
`MIRRORS`	A list of sites that mirror the NetBSD 2.0.2 distribution.
`README.files`	README describing the distribution's contents.
`TODO`	The NetBSD 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 2.0.2 has a binary distribution.

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

gnusrc This set contains the "gnu" sources, including the source for the compiler, assembler, groff, and the other GNU utilities in the binary distribution sets.
79 MB gzipped, 367 MB uncompressed

pkgsrc This set contains the "pkgsrc" sources, which contain the infrastructure to build third-party packages.
24 MB gzipped, 200 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.
5 MB gzipped, 20 MB uncompressed

src This set contains all of the base NetBSD 2.0.2 sources which are not in **gnusrc**, **sharesrc**, or **syssrc**.
37 MB gzipped, 176 MB uncompressed

syssrc This set contains the sources to the NetBSD 2.0.2 kernel for all architectures; `config`(8); and `dbsym`(8).
26 MB gzipped, 140 MB uncompressed

xsrc This set contains the sources to the X Window System.
84 MB gzipped, 450 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. Except for the **pkgsrc** set, which is traditionally unpacked into `/usr/pkgsrc`, all sets may be unpacked into `/usr/src` with the command:

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

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

`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 ATT 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/mvme68k subdirectory structure

The mvme68k-specific portion of the NetBSD 2.0.2 release is found in the `mvme68k` subdirectory of the distribution: `.../NetBSD-2.0.2/mvme68k/`. It contains the following files and directories:

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

`binary/`
`kernel/`
`netbsd-GENERIC.gz`
A gzipped NetBSD kernel containing code for everything supported in this release.
`netbsd-VME147.gz`
A kernel for MVME147 boards.
`netbsd-VME162.gz`
A kernel for MVME162 boards.
`netbsd-VME167.gz`
A kernel for MVME167 boards.
`netbsd-VME172.gz`
A kernel for MVME172 boards.
`netbsd-VME177.gz`
A kernel for MVME177 boards.
`sets/` mvme68k binary distribution sets; see below.

`installation/`
`miniroot/`
mvme68k miniroot file system image; see below.
`netboot/`
Two programs needed to boot mvme68k kernels over the network; see below.
`tapeimage/`
Tape boot programs, and a RAMDISK kernel; see below.

Binary distribution sets

The NetBSD mvme68k binary distribution sets contain the binaries which comprise the NetBSD 2.0.2 release for the mvme68k. The binary distribution sets can be found in the `mvme68k/binary/sets` subdirectory of the NetBSD 2.0.2 distribution tree, and are as follows:

base The NetBSD 2.0.2 mvme68k **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 MB gzipped, 46 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.
18 MB gzipped, 69 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.
1 MB gzipped, 1 MB uncompressed

games This set includes the games and their manual pages.
3 MB gzipped, 7 MB uncompressed

kern-GENERIC
This set contains a NetBSD/mvme68k 2.0.2 `GENERIC` kernel, named `/netbsd`. You *must* install this distribution set.
1 MB gzipped, 2 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.
8 MB gzipped, 30 MB uncompressed

misc This set includes the (rather large) system dictionaries, the typesettable document set, and other files from `/usr/share`.
3 MB gzipped, 9 MB uncompressed

text This set includes NetBSD's text processing tools, including `groff`(1), all related programs, and their manual pages.
2 MB gzipped, 7 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 4.4.0. 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.
6 MB gzipped, 17 MB uncompressed

xcomp The extra libraries and include files needed to compile X source code.
10 MB gzipped, 37 MB uncompressed

xfont Fonts needed by X.
31 MB gzipped, 39 MB uncompressed

xetc Configuration files for X which could be locally modified.
0.03 MB gzipped, 0.17 MB uncompressed

xserver The X server.
3 MB gzipped, 7 MB uncompressed

The mvme68k binary distribution sets are distributed as gzipped tar files named with the extension **.tgz**, e.g. `base.tgz`.

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

The following are included in the `mvme68k/installation` directory:

`miniroot/`

Note: The **`sysinst`** method of installation is the preferred method for installing NetBSD/mvme68k. *The Traditional method of installation is deprecated and will be removed in a future release.* Some other important differences between the two installation methods are:

- The Installer and Mkfs utilities are not used if the installation is done using the **sysinst** method via booting an Installation Kernel.
- Mkfs creates a 4.3BSD "old" format file system. This is the only file system format understood by the Installer. The **sysinst** process creates "new" file systems which can't be processed by Mkfs or the Installer.
- The Installer (and probably Mkfs) is known to have problems with BSD file systems that are larger than 1 GB or extend beyond the 1 GB physical limit. Keep this in mind if doing a Traditional method installation with these utilities.
- The Booter is known to have problems booting from a NetBSD file system if the inode of the kernel file is to large. For large disks it is advisable to have a small root partition and one or more usr-type partitions for other files. This ensures the inode of the kernel in the root file system will be small.
- The Booter is designed to work with 4.3BSD "old" file systems, but there are enough similarities between the "old" and "new" file system formats in the first few hundred blocks that the Booter can usually boot a Kernel from a 4.3BSD new file system. If you choose to use "new" file system formats, it is advisable to use a small root partition and one or more usr-type partitions for other files. This is the default layout used by **sysinst**.

`miniroot.gz` A copy of the miniroot file system. This file system image is copied into the swap partition of the disk which you intend to boot NetBSD from. Normally, this step will be performed manually from the ramdisk boot environment.

`netboot/`

`sboot` An MVME147 bootstrap program in Motorola S-Record format. This is required when you wish to boot an MVME147 over the network since the ROM has no built-in network support. Instructions for getting this program into memory are discussed later.

`netboot` A standalone 2nd stage bootstrap program loaded over the network via TFTP. This is responsible for fetching and starting the NetBSD mvme68k kernel from an NFS server.

`tapeimage/`

`stboot` This file contains a boot sector for Motorola MVME boards. It must be the first file written to a tape in order to make the tape bootable.

`bootst` This file contains a bootstrap program which knows how to load the NetBSD mvme68k ramdisk image from tape. This must be the second file written to the tape.

`netbsd-RAMDISK.gz` This is the NetBSD mvme68k ramdisk image. It contains a `GENERIC` kernel and a built in RAMDISK with just enough tools to partition a disk, dump the miniroot kernel to it and make the disk bootable. This must be the third file written to the tape.

Note: Each directory in the mvme68k binary distribution also has its own checksum files, just as the source distribution does.

NetBSD/mvme68k System Requirements and Supported Devices

NetBSD/mvme68k 2.0.2 runs on Motorola MVME147 , MVME162 , MVME167 , MVME172 , and MVME177 Single Board Computers.

The minimal configuration requires 8 MB of RAM and ˜200 MB of disk space. To install the entire system requires much more disk space (approx. 600 MB additional space is necessary for full sources. Double that if you want to recompile it all!). To run X (clients only) or compile the system, more RAM is recommended. Good performance requires 16 MB of RAM, or 32 MB when running the X Window System.

Note that you can install NetBSD 2.0.2 on a system with only 4 MB of onboard RAM, but you will need to use a VMEbus RAM card with at least another 4 MB to augment the onboard memory in order to actually install the system.

Here is a table of recommended HD partition sizes for a full install:

Partition	Suggested	Needed
/ (root)	32 MB	26 MB
/usr	200 MB	150 MB
/var	32 MB	6 MB
swap	*2-3 ∗RAM*	16 MB

Anything else is up to you!

Note that the NetBSD/mvme68k installation procedure uses a *miniroot* image which is placed into the swap area of the disk. The swap partition must be large enough to hold this miniroot image (7.5 MB).

Supported VME147 hardware

- Serial ports (RS232)
  - built-in console
  - ttyZ1
  - ttyZ2
  - ttyZ3
- Parallel port
  - On-board centronics style printer port
- Network interfaces:
  - On-board Lance Ethernet (le)
- SCSI
  - Most SCSI disks, tapes, CD-ROMs, etc
  - On-board wd33c93 SCSI bus interface chip (async only)
- Miscellaneous:
  - Battery-backed real-time clock
  - VMEbus RAM cards
  - Any VMEbus interface cards with Machine Independent drivers

Supported VME162/VME172 hardware

- Serial ports (RS232)
  - built-in console
  - ttyC1
  - ttyC2
  - ttyC3

- Network interfaces:
  - On-board Intel Ethernet (ie)
- SCSI
  - Most SCSI disks, tapes, CD-ROMs, etc
  - On-board ncr53c710 SCSI I/O processor chip
- Miscellaneous:
  - Battery-backed real-time clock
  - VMEbus RAM cards
  - Any VMEbus interface cards with Machine Independent drivers

Supported VME167/VME177 hardware

- Serial ports (RS232)
  - built-in console
  - ttyZ1
  - ttyZ2
  - ttyZ3
- Parallel port
  - On-board centronics style printer port
- Network interfaces:
  - On-board Intel Ethernet (ie)
- SCSI
  - Most SCSI disks, tapes, CD-ROMs, etc
  - On-board ncr53c710 SCSI I/O Processor chip
- Miscellaneous:
  - Battery-backed real-time clock
  - VMEbus RAM cards
  - Any VMEbus interface cards with Machine Independent drivers

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

Getting the NetBSD System on to Useful Media

Installation is supported from several media types, including:

- Tape
- NFS
- CD-ROM
- FTP

Note that installing on a 'bare' machine requires either a bootable tape drive or an ethernet to a compatible NFS server. MVME147 may also need to be booted over an RS232 connection.

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

In order to create installation media, you will need all the files and subdirectories in these two directories:

```
.../NetBSD-2.0.2/mvme68k/installation
.../NetBSD-2.0.2/mvme68k/binary
```

Creating boot/install tapes

Installing from tape is the simplest method of all. This method uses two tapes, one containing a bootable ramdisk and miniroot, the other containing the installation sets.

The boot tape is created as follows:

```
# cd .../NetBSD-2.0.2/mvme68k/installation
# set T = /dev/nrst0
# mt -f $T rewind
# dd if=tapeimage/stboot of=$T
# dd if=tapeimage/bootst of=$T obs=8k conv=osync
# gzip -dc tapeimage/netbsd-RAMDISK.gz | dd of=$T obs=8k conv=osync
# gzip -dc miniroot/miniroot.fs.gz  | dd of=$T obs=8k conv=osync
# mt -f $T rewind
```

The installation set tape is created as follows:

```
# cd .../NetBSD-2.0.2/mvme68k/binary/sets
# set T = /dev/nrst0
# mt -f $T rewind
# for f in base etc comp games man misc text; do
    gzip -d  $f.tgz | dd of=$T bs=8k
# done
# mt -f $T rewind
```

If the tape does not work as expected, you may need to explicitly set the EOF mark at the end of each tape segment. Consult the tape-related manual pages on the system where the tapes are created for more details.

Boot/Install from NFS server

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

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

Booting an MVME147 from ethernet is not possible without first downloading a small bootstrap program (sboot) via RS232. See the section entitled **Installing from NFS** for details on how to accomplish this.

sboot expects to be able to download a second stage bootstrap program via TFTP after having acquired its IP address through RARP It will look for a filename derived from the machine's IP address expressed in hexadecimal, with an extension of '`.147`'. For example, an MVME147 with IP address 130.115.144.11 will make an TFTP request for `8273900B.147`. Normally, this file is just a symbolic link to the NetBSD/mvme68k `netboot` program, which should be located in a place where the TFTP daemon can find it (remember, many TFTP daemons run in a chroot'ed environment). The netboot program may be found in the install directory of this distribution.

The MVME162 , MVME167 , MVME172 , and MVME177 boot ROMs have code builtin to boot over ethernet from a TFTP server. You should configure it to download the same `netboot` program as is used for MVME147.

The netboot program will query a bootparamd server to find the NFS server address and path name for its root, and then load a kernel from that location. The server should have a copy of the netbsd-RAMDISK ker-

nel in the root area for your client (no other files are needed in the client root, although it might be a convenient place to put the uncompressed miniroot image) and `/etc/bootparams` on the server should have an entry for your client and its root directory. Note that you should rename the `netbsd-RAMDISK` kernel to just `netbsd` in the client's root directory before trying to netboot the client.

The client will need access to the miniroot image, which can be provided using NFS or remote shell. If using NFS, miniroot.fs.gz should be expanded on the server, because doing so from the RAMDISK shell is not so easy. The unzipped miniroot takes about 7.5 MB of space.

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

Install/Upgrade from CD-ROM

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

Install/Upgrade via FTP

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

This method, of course, requires network access to an FTP server. This might be a local system, or it might even be ftp.NetBSD.org itself. If you wish to use ftp.NetBSD.org as your FTP file server, you may want to keep the following information handy:

```
No IP Address: ftp.NetBSD.org
Login: anonymous
Password: your e-mail address
Server path: /pub/NetBSD/NetBSD-2.0.2/mvme68k/binary
```

Note: If you're not using a nameserver during installation, you might find 204.152.184.75 handy; it's the IP address of ftp.NetBSD.org as of October, 2000.

Preparing your System for NetBSD installation

mvme68k machines usually need little or no preparation before installing NetBSD, other than the usual, well advised precaution of *backing up all data* on any attached storage devices.

The exception to the above is that MVME162 , MVME167 , MVME172 and MVME177 boards require a jumper to be removed or DIP switch changed before NetBSD can be installed. On MVME162-LX and MVME172-LX pins 1-2 of jumper J11 *must* be removed. On MVME162-P2/P4 and MVME172-P2/P4 switch S4, position 8 *must* be set to OFF. On MVME167 and MVME177 pins 1-2 of jumper J1 *must* be removed.

Once you've made any necessary jumper changes, the following instructions should make your machine "NetBSD Ready".

Power-up your MVME147 board. You should have the *bug* prompt:

```
COLD Start
```

```
Onboard  RAM start = $00000000,  stop = $007FFFFF

147-Bug
```

Or, if you have an MVME162/172 or MVME167/177 board (the following boot message is from MVME167; the others are similar):

```
MVME167 Debugger/Diagnostics Release Version 2.3 - 02/25/94
COLD Start

Local Memory Found =02000000 (&33554432)

MPU Clock Speed =33Mhz

167-Bug
```

Make sure the RAM size looks ok (if you've got an 8 MB MVME147 or a 32 MB MVME167 you should have the same value as we do). Also make sure the clock is ticking:

```
1xx-Bug>time
Sunday 12/21/31 16:25:14
1xx-Bug>time
Sunday 12/21/31 16:25:15
1xx-Bug
```

Note that NetBSD bases its year at 1968, and adds the year offset in the system's real-time clock to get the current year. So the 31 here equates to 1999. You may have to adjust your clock using the **set** command to comply with NetBSD 's requirements. Don't worry if the 'Day of the week' is not correct, as NetBSD doesn't use it. Motorola has acknowledged a year 2000 bug in some versions of the MVME147 whereby the day of the week doesn't get set correctly by the 147Bug PROM.
NetBSD does not affect

Also make sure that your board's ethernet address is initialised to the correct value. You'll find the address on a label on the inside of the MVME147's front panel, and on the VMEbus P2 connector of the other board types. On the MVME147, enter the last five digits of the address using the **lsad** command. On the MVME162/172 and MVME167/177, you should use the **cnfg** command.

The NetBSD kernel reads the first two long words of the onboard NVRAM to determine the starting and ending address of any VMEbus RAM that should be used by the system. You should verify that this area is set properly for your configuration.

If you have no VMEbus RAM boards, the values should be set to zero (0).

For an MVME162, MVME167, MVME172 or MVME177 board, at the 1xx-Bug prompt:

```
1xx-Bug>mm fffc0000 ;1
fffc0000: xxxxxxxx?0
fffc0004: xxxxxxxx?0
fffc0008: xxxxxxxx?.
1xx-Bug
```

For an MVME147 board, at the 147Bug prompt:

```
147Bug>mm fffe0764 ;1
fffe0764: xxxxxxxx?0
fffe0768: xxxxxxxx?0
fffe076c: xxxxxxxx?.
```

If you do have VMEbus RAM available and want NetBSD to use it, the first long word should be set to the starting address of this RAM and the second long word should be set to the ending address.

If you have more than one VMEbus RAM board installed, the starting and ending addresses must be contiguous from one board to the next. Also note that, for various reasons beyond the scope of this document, VMEbus RAM should be configured in A32 address space.

To install successfully to a local SCSI disk, you need to ensure that the system is aware of what targets are connected to the SCSI bus. This can be done by issuing the following command:

```
1xx-Bug iot;t
```

At this point, Bug will scan for any attached SCSI devices. After a short delay, a list of SCSI devices will be displayed. 147Bug will ask if LUNs should be assigned from SCSI ids, to which you should answer Y. You should also answer Y when asked if the information is to be saved to NVRAM. 16xBug does not prompt for this information.

The following installation instructions will assume that your target SCSI disk drive appears at SCSI-ID 0. If you have a tape drive, the instructions assume is is configured for SCSI-ID 5. When the RAMDISK root boots, NetBSD will refer to these devices as `sd0` and `rst0` respectively. You may wish to note these down; you'll be using them a lot. :-)

Installing the NetBSD System

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

There are several ways to install NetBSD onto your disk. If your MVME147 machine has a tape drive the easiest way is **Installing from tape** (details below). All other machines can be installed easily over the network from a suitable NFS server. See **Installing from NFS** for details. Otherwise, if you have another mvme68k machine running NetBSD you can initialize the disk on that machine and then move the disk.

Installing from tape

Create the NetBSD/mvme68k 2.0.2 boot tape as described in the section entitled **Preparing a boot tape**. Then, with the tape in the drive, type the following at the Bug prompt:

```
147-Bug bo 5
```

```
16x-Bug bo 0,50
```

As mentioned earlier, this assumes your tape is jumpered for SCSI-ID 5.

As the tape loads (which may take 20 to 30 seconds), you will see a series of status messages. It may be useful if you can capture these messages to a file, or a scrollable xterm window. In particular, you should make a note of the lines which describe the geometry of the SCSI disks detected by NetBSD. They are of the form:

```
sd0 at scsibus0 targ 0 lun 0: Maker, Disk, Foo SCSI1 0/direct fixed
sd0: 800 MB, 800 cyl, 16 head, 128 sec, 512 bytes/sect x 1638400 sectors
```

The information of most interest is the number of sectors; for the fictitious disk above, it's 1638400. You will need this number when you come to create a disklabel for that drive.

Here is an example of an MVME147 system booting from tape:

```
RAM address from VMEbus = $00000000

Booting from: VME147, Controller 5, Device 0
Loading: Operating System

Volume: NBSD
```

```
IPL loaded at:  $003F0000
 BSD MVME147 tapeboot [$Revision: 1.17 $]
578616+422344+55540+[46032+51284]=0x11a6e4
Start @ 0x8000 ...
Copyright (c) 1996, 1997, 1998, 1999, 2000
    The NetBSD Foundation, Inc.  All rights reserved.
Copyright (c) 1982, 1986, 1989, 1991, 1993
    The Regents of the University of California.  All rights reserved.

NetBSD 1.5 (RAMDISK) #1: Sun Oct 29 16:19:04 GMT 2000
    steve@fatbob:/usr/src/sys/arch/mvme68k/compile/RAMDISK
Motorola MVME-147S: 25MHz MC68030 CPU+MMU, MC68882 FPU
real mem  = 7237632
avail mem = 6381568
using 88 buffers containing 360448 bytes of memory
mainbus0 (root)
pcc0 at mainbus0: Peripheral Channel Controller, rev 0, vecbase 0x40
clock0 at pcc0 offset 0x0 ipl 5: Mostek MK48T02, 2048 bytes of NVRAM
  .
  .
```

Note: The exact text of the messages will vary depending on which mvme68k variant you're using.

Finally, you will see the following "welcome" message:

```
        Welcome to the NetBSD/mvme68k RAMDISK root!

This environment is designed to do only four things:
  1:  Partition your disk (use the command:  edlabel /dev/rsd0c)
  2:  Copy a miniroot image into the swap partition  (/dev/rsd0b)
  3:  Make that partition bootable (using 'installboot')
  4:  Reboot (using the swap partition, i.e. /dev/sd0b).

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

The easiest is loading from tape, which is done as follows:
        mt -f /dev/nrst0 rewind
        mt -f /dev/nrst0 fsf 3
        dd bs=8k if=/dev/nrst0 of=/dev/rsd0b
(For help with other methods, please see the install notes.)

To reboot using the swap partition after running installboot, first
use
halt,
then at the Bug monitor prompt use a command like:
               1x7Bug bo 0,,b:

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

You must now create a disklabel on the disk you wish to use for the root file system (`/`). This will usually be `sd0`. The disklabel is used by NetBSD to identify the starting block and size of each partition on the disk.

Partitions are named `sd0a`, `sd0b`, `sd0c`, etc, up to `sd0h`. The mvme68k port of NetBSD makes some assumptions about the first three partitions on a boot disk:

`sd0a` The root file system (`/`).
`sd0b` The swap partition.
`sd0c` The whole disk. Also known as the *raw partition*.

The *raw partition* is special; NetBSD is able to use it even if the disk has no label. You should never create a file system on the Raw Partition, even on a non-boot disk.

It is good practice to put `/usr` on a different partition than / (root, AKA `sd0a`). So, the first available partition for `/usr` is `sd0d`. Refer to the section entitled **NetBSD System Requirements and Supported Devices** for information on the recommended sizes of the / (root), `/usr` and swap partitions.

You are not required to define any partitions beyond `sd0d`, but if you have a large disk drive, you might want to create several other partitions for file systems such as `/home` or `/usr/src`. Note that at this time you are only required to partition the root/boot disk; you will get the opportunity to partition any other disks in your system from the main **miniroot** installation program.

To create the disklabel and partitions, use the `edlabel` program, passing it the name of the Raw Partition of your root/boot disk.

```
# edlabel /dev/rsd0c
edlabel menu:
print - display the current disk label
modify - prompt for changes to the label
write - write the new label to disk
quit - terminate program
edlabel
```

The program shows what commands it recognizes; **print**, **modify**, **write**, and **quit**. It will accept the first letter of a command if you don't feel like typing each one in full.

To start creating the basic partitions, you should enter **m** (modify) at the edlabel prompt, then enter the letter corresponding to the first partition, **a**.

```
edlabel m

modify subcommands:
@: modify disk parameters
a-h: modify partition
s: standardize geometry
q: quit this subcommand
edlabel/modify a
a (root) 0 (0/00/00) 0 (0/00/00) unused
start as blkno or cyls/trks/sects: 0
length as nblks or cyls/trks/sects: 65536
type: 4.2BSD
edlabel/modify
```

When you enter the start and length of a partition, you can use either blocks or cylinder/track/sector notation. If this is the first time you've partitioned a disk for NetBSD, it's probably easiest to use block notation. The above example creates partition 'a', starting at block zero and with a size of 65536 blocks. Note that the usual size of a block is 512 bytes, so this creates a 32 MB partition.

The `type` of the partition should be `4.2BSD`, otherwise you won't be able to create a file system on it.

Next, create a swap partition (b). Note that the minimum size of this swap partition should be 8 MB, otherwise you won't be able to use a miniroot to complete the NetBSD installation!

```
edlabel/modify b
b (swap) 0 (0/00/00) 0 (0/00/00) unused
start as blkno or cyls/trks/sects: 65536
length as nblks or cyls/trks/sects: 32768
type: swap
edlabel/modify
```

Here, we specify a value for `start` such that the swap partition follows immediately after partition '`a`', i.e. 65536. The length of the swap partition should be a multiple of the amount of RAM you have in your system. Here, we've chosen 32768, or 16 MB. The next available block on the drive is thus 65536 + 32768. We will use this to create partition '`d`' for our `/usr` file system. (Note that for a busy system, or a system with more than 8 MB of RAM, you'll be better off with a 32 or 64 MB swap partition.)

```
edlabel/modify d
d (user) 0 (0/00/00) 0 (0/00/00) unused
start as blkno or cyls/trks/sects: 98304
length as nblks or cyls/trks/sects: 1540096
type: 4.2BSD
edlabel/modify q
edlabel
```

As you can see, we've chosen to assign the remainder of the disk to `/usr`. Since there are 1638400 sectors on the example disk (did you remember to note down the number of sectors on your disk during boot?), and partition **d** starts at sector 98304, a simple bit of arithmetic (1638400 - 98304) gives **d** a size of 1540096.

Note that the above partition sizes are just guidelines. If your disk is large enough, you should resize the partitions appropriately and perhaps also create a /var partition as well.

You now need to write this new disklabel, together with the partition details you've just entered, to disk. You might also try the '**p**' command to view the partitions. Once written, you can quit back to the shell using '**q**'.

```
edlabel p

        type_num: 4
        sub_type: 0
       type_name: SCSI disk
       pack_name: fictitious
    bytes/sector: 512
   sectors/track: 128
 tracks/cylinder: 16
       cylinders: 800
sectors/cylinder: 2048
partition      start         (c/t/s)      nblks         (c/t/s)  type

 a (root)          0       (0/00/00)      65536      (32/00/00)  4.2BSD
 b (swap)      65536      (32/00/00)      32768      (48/00/00)  swap
 c (disk)          0       (0/00/00)    1638400     (800/00/00)  unused
 d (user)      98304      (48/00/00)    1540096     (752/00/00)  4.2BSD

edlabel w
edlabel q
#
```

Now that your disk's partitioned, you need to get the proper installation miniroot image onto it. The miniroot image is designed to be copied into the swap partition of your disk. This is a safe place which won't be overwritten by the installation procedure. From the shell prompt, use the following commands to copy the miniroot image from tape to swap (b).

```
# mt -f /dev/nrst0 rewind
# mt -f /dev/nrst0 fsf 3
# dd bs=8k if=/dev/nrst0 of=/dev/rsd0b
```

The disk and the miniroot must now be made bootable using the `installboot`(8) command. To do this, issue the following commands:

```
# mount /dev/sd0b /mnt
# installboot /mnt/usr/mdec/bootsd /bootxx /dev/rsd0b
# umount /dev/sd0b
```

You can now shutdown the system.

```
# halt
signal 15
syncing disks... done
unmounting / (root_device)...
halted

147-Bug reset
Reset Local SCSI Bus [Y,N] N? y
Automatic reset of known SCSI Buses on RESET [Y,N] Y?
Cold/Warm Reset flag [C,W] = C?
Execute Soft Reset [Y,N] N? y
```

Resetting the other types of MVME boards are very similar. You should now reboot from that just installed miniroot. See the section entitled **Booting the miniroot** for details.

Installing from NFS

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

To get started on the MVME147, you need to download `sboot` into RAM (you will find **`sboot`** in the `install` directory of the mvme68k distribution). You can either do that through the console line or through a 2nd serial connection. For example, an MVME147 connected to a sun4/110 and accessed via `tip`(1) can be loaded as follows:

```
lo 0
~Ccat sboot
go 4000
```

Which will look like this:

```
147-Bug lo 0
~CLocal command? cat sboot
```

away for 11 seconds

```
!

147-Bug g 4000
Effective address: 00004000
```

```
sboot: serial line bootstrap program (end = 6018)
```

Now, if you want to do it through serial line 1, then connect serial line one to a machine. At the 147-Bug prompt type **tm 1** You should then login to the machine it is connected to. Then press CONTROL-A to escape to Bug. Do **lo 1;x=cat sboot** ... then when that is done you can reconnect **tm 1** and logout. Then do **go 4000** and you've got the prompt of **sboot**.

Once you've got the prompt, you can boot the RAMDISK kernel from the server:

```
b

le0: ethernet address: 8:0:3e:20:cb:87
My ip address is: 192.168.1.4
Server ip address is: 192.168.1.1
4800
Download was a success!
```

See below for the next step in booting MVME147.

The MVME162, MVME167, MVME172 and MVME177 boards are able to download netboot directly using TFTP. To enable this, you must first configure the networking parameters on the board as described in the section entitled "Preparing your System for NetBSD Installation. On a properly configured MVME162/172 or MVME167/177, all you need to type is:

```
1xx-Bug nbo
```

For all board types, the boot messages are very similar:

```
Start @ 0x8000 ...
 BSD MVME147 netboot (via sboot) [$Revision: 1.17 $]
device: le0 attached to 08:00:3e:20:cb:87
boot: client IP address: 192.168.1.4
boot: client name: soapy
root addr=192.168.1.1 path=/export/soapy
578616+422344+55540+[46032+51284]=0x11a6e4
Start @ 0x8000 ...
Copyright (c) 1996, 1997, 1998, 1999, 2000
    The NetBSD Foundation, Inc.  All rights reserved.
Copyright (c) 1982, 1986, 1989, 1991, 1993
    The Regents of the University of California.  All rights reserved.

NetBSD 1.5 (RAMDISK) #1: Sun Oct 29 16:19:04 GMT 2000
    steve@fatbob:/usr/src/sys/arch/mvme68k/compile/RAMDISK
Motorola MVME-147S: 25MHz MC68030 CPU+MMU, MC68882 FPU
real mem  = 7237632
avail mem = 6381568
using 88 buffers containing 360448 bytes of memory
mainbus0 (root)
pcc0 at mainbus0: Peripheral Channel Controller, rev 0, vecbase 0x40
clock0 at pcc0 offset 0x0 ipl 5: Mostek MK48T02, 2048 bytes of NVRAM
  .
  .
```

After the boot program loads the RAMDISK kernel, you should see the welcome screen as shown in the "tape boot" section above.

You now need to create a disklabel with partition information on the SCSI disk on which you intend to create your root file system (/). Follow the instructions in the previous section entitled **Installing from tape** to do this. (But stop short of the part which describes how to copy the miniroot from tape.)

You must now configure the network interface before you can access the NFS server containing the miniroot image. For example the command:

```
# ifconfig le0 inet 192.168.1.4 up
```

will bring up the MVME147 network interface le0 with that address. The command:

```
# ifconfig ie0 inet 192.168.1.4 up
```

will bring up the MVME162/172 or MVME167/177 network interface ie0 with that address. The next step is to copy the miniroot from your server. This can be done using either NFS or remote shell. (In the examples that follow, the server has IP address 192.168.1.1) You may then need to add a default route if the server is on a different subnet:

```
# route add default 192.168.1.2 1
```

You can look at the route table using:

```
# route show
```

Now mount the NFS file system containing the miniroot image:

```
# mount -r 192.168.1.1:/export/soapy /mnt
```

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

```
# dd bs=8k if=/mnt/miniroot of=/dev/rsd0b
```

Otherwise, you will need to use **gzcat** to expand the miniroot image while copying.

```
# gzcat miniroot.fs.gz | dd obs=8k of=/dev/rsd0b
```

You must now make the disk bootable. Refer to the previous section on installing from tape, where it describes how to run **installboot**. This is immediately following the part which explains how to copy the miniroot from tape.

Booting the miniroot

Assuming the miniroot is installed on partition 'b' of the disk with SCSI-ID 0, then the boot command is:

```
1xx-Bug bo 0,,b:
```

The command line parameters above are:

0 controller (usually zero)

,, bug argument separators

b: tell the bootstrap code to boot from partition b

You should see a bunch of boot messages, followed by messages from the miniroot kernel just as you did when the RAMDISK kernel booted.

You will then be prompted to enter the root device. Since the miniroot was booted from the swap partition, you should enter sd0b. You will then be asked for the swap device and file system type. Just press RETURN twice to accept the defaults. When asked to enter a terminal type, either accept the default, or use whatever the TERM environment variable is set to in the shell of your host system:

```
boot device: sd0
root device (default sd0a): sd0b
dump device (default sd0b): (return)
file system (default generic): (return)
root on sd0b dumps on sd0b
mountroot: trying ffs...
root file system type: ffs
init: copying out path '/sbin/init' 11
erase ^H, werase ^W, kill ^U, intr ^C
Terminal type? [vt100] return
```

Congratulations! The system should now be running the miniroot installation program.

Miniroot install program:

The miniroot's install program is very simple to use. It will guide you through the entire process, and is well automated.

Starting with NetBSD version 1.6, you have the option of using **sysinst** instead of the old installation and upgrade shell scripts. At this time, **sysinst** is not capable of installing distribution sets from tape. In such a case, you should fallback to the shell script installation or upgrade procedure. In all other cases, **sysinst** should be used. Note that as soon as **sysinst** can deal with tapes, the shell script tools will be removed.

The miniroot's install program (both **sysinst** and the shell script version) will:

- Allow you to place disklabels on additional disks. The disk we are installing on should already have been partitioned using the `RAMDISK` kernel.

  Note that partition sizes and offsets are expressed in sectors. When you fill out the disklabel, you will need to specify partition types and file system parameters. If you're unsure what these values should be, use the following defaults:

  ```
  fstype: 4.2BSD
  fsize: 1024
  bsize: 4096
  cpg: 16
  ```

  If the partition will be a swap partition, use the following:

  ```
  fstype: swap
  fsize: 0 (or blank)
  bsize: 0 (or blank)
  cpg: 0 (or blank)
  ```

  Note that partition '`c`' is special; it covers then entire disk and should not be assigned to a file system.

  The number of partitions is currently fixed at 8.

- Create file systems on target partitions.
- Allow you to set up your system's network configuration. Remember to specify host names without the domain name appended to the end. For example use `foo` instead of `foo.bar.org`. If, during the process of configuring the network interfaces, you make a mistake, you will be able to re-configure that interface by simply selecting it for configuration again.
- Mount target file systems. You will be given the opportunity to manually edit the resulting `/etc/fstab`.

- Extract binary sets from the media of your choice.
- Copy configuration information gathered during the installation process to your root file system (`/`).
- Make device nodes in your root file system under `/dev`.
- Copy a new kernel onto your root partition (`/`).
- Install a new boot block.
- Check your file systems for integrity.

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

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 for the sake of completeness. Do not let this discourage you; the install program is not hard to use.

2. *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 by running the `/sysinst` program from the command prompt. It is not necessary to reboot.

3. *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 the next section. 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).
     - One 1.44 MB 3.5" floppy.
     - A CD-ROM drive (SCSI), a hard disk and a minimum of 4 MB of memory installed.
     - The hard disk should have at least 100 + *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 120 MB more.
   - The Quick Installation
     - Insert the first boot floppy you just created. Boot the computer. After language selection, the main menu will be displayed.

```
.*********************************************.
* NetBSD-2.0 Install System                    *
*                                              *
*a: Install NetBSD to hard disk                *
* b: Upgrade NetBSD on a hard disk             *
* c: Re-install sets or install additional sets *
* d: Reboot the computer                       *
* e: Utility menu                              *
* x: Exit Install System                       *
.*********************************************.
```

- If you wish, you can configure some network settings immediately by choosing the **Utility 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/**.

4. *Booting 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.

   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 the NetBSD 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.

5. *Network configuration*

   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 **Utility menu**, and select the **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 the Domain Name System (DNS), you can give

an empty response in reply to answers relating to this.

6. *Installation drive selection and parameters*

   To start the installation, select **Install NetBSD to hard disk** 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 `sd0` or `sd1`.

7. *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.

8. *Editing the NetBSD disklabel*

   The partition table of the NetBSD part of a disk is called a *disklabel*. There are 4 layouts for the NetBSD part of the disk that you can pick from: **Standard, Standard with X, Custom** and **Use Existing**. The first two use a set of default values (that you can change) suitable for a normal installation, possibly including X. With the **Custom** option you can specify everything yourself. The last option uses the partition info already present on the disk.

   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.

`a`	Root partition (`/`)
`b`	Swap partition.
`c`	The NetBSD portion of the disk.
`d-h`	Available for other use. Traditionally, `d` 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 will be ok for most purposes. 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.

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 the 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 if you want to use DHCP, and if not to provide data like IP address, 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, the account name and password used to log into that host using ftp, and optionally a proxy server to use. If you did not set up DNS when answering the questions to configure networking, you will need to specify an IP address 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 if you want to use DHCP, and if not to provide data like IP address, 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 address 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 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.

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 file system 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. 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 UTC. Finally you will be asked to select a password encryption algorithm and can than set a password for the "root" account, to prevent the machine coming up without access restrictions.

17. *Finalizing your installation*

    Congratulations, you have successfully installed NetBSD 2.0.2. You can now reboot the machine, and boot NetBSD from hard disk.

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` (**sysinst** usually will), the system will drop you into single user mode on first reboot with the message

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

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

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

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

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

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

   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 ⟨int⟩ network interface, along the lines of

```
ifconfig_le0="inet 123.45.67.89 netmask 255.255.255.0"
```

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

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

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

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

2. Logging in

   After reboot, you can log in as root at the login prompt. Unless you've set a password in **sysinst**, there is no initial password. 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. By default, root login from the network is disabled (even via ssh(1)). One way to become root over the network is to log in as a different user that belongs to group "wheel" (see group(5)) and use su(1) to become root.

   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 vipw(8) and pwd_mkdb(8) if you want to edit the password database.

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 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 list of available packages suitable for browsing is at
     ftp://ftp.NetBSD.org/pub/NetBSD/packages/pkgsrc/README.html
   - Precompiled binaries can be found at ,:
     ftp://ftp.NetBSD.org/pub/NetBSD/packages/
     usually in the 2.0.2/mvme68k/All subdir. You can install them with the following commands under sh(1):

```
# PKG_PATH=ftp://ftp.NetBSD.org/pub/NetBSD/packages/2.0.2/mvme68k/All
# export PKG_PATH
# pkg_add -v tcsh
# pkg_add -v bash
# pkg_add -v perl
# pkg_add -v apache
# pkg_add -v kde
# pkg_add -v mozilla
...
```

If you are using csh(1) then replace the first two lines with the following:

```
# setenv PKG_PATH ftp://ftp.NetBSD.org/pub/NetBSD/packages/2.0.2/mvme68k/All
...
```

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

- Package sources for compiling packages on your own 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), with the commands:

```
# mkdir /usr/pkgsrc
# ( cd /usr/pkgsrc ; tar -zxpf - )  pkgsrc.tar.gz
```

  After extracting, see the README and Packages.txt files 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. Don't forget to 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. If you prefer postfix as MTA, adjust /etc/mailer.conf.
   - Edit /etc/rc.local to run any local daemons you use.
   - Many of the /etc files are documented in section 5 of the manual; so just invoking

```
# man 5 filename
```

     is likely to give you more information on these files.

Upgrading a previously-installed NetBSD System

It is possible to easily upgrade your existing NetBSD/mvme68k system using the upgrade program in the miniroot or by manually performing the same steps as the miniroot upgrade program.

Upgrading using the miniroot

If you wish to upgrade your system by this method, simply select the **upgrade** option once the miniroot has booted. The upgrade program with then guide you through the procedure. The upgrade program will:

1. Enable the network based on your system's current network configuration.

2. Mount your existing file systems.
3. Extract binary sets from the media of your choice.
4. Fixup your system's existing `/etc/fstab`, changing the occurrences of **ufs** to **ffs** and let you edit the resulting file.
5. Make new device nodes in your root file system under `/dev`.
6. Don't forget to extract the **kern** set from the distribution.

   Note: The existing kernel *will not* be backed up; doing so would be pointless, since older kernels may not be capable of running NetBSD 2.0.2 executables.
7. Install a new boot block.
8. Check your file systems for integrity.
9. You'll have to reboot your system manually

Manual upgrade

While using the miniroot's upgrade program is the preferred method of upgrading your system, it is possible to upgrade your system manually. To do this, follow the following procedure:

1. Place *at least* the **base** binary set in a file system accessible to the target machine. A local file system is preferred, since the NFS subsystem in the NetBSD 2.0.2 kernel may be incompatible with your old binaries.
2. Back up your pre-existing kernel and copy the 2.0.2 kernel into your root partition (`/`).
3. Reboot with the 2.0.2 kernel into single-user mode.
4. Check all file systems:

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

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

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

   ```
   # cd /
   # pax -zrvpe -f /path/to/base.tgz
   ```
8. Install a new boot block:

   ```
   # cd /usr/mdec
   # cp bootsd /.bootsd
   # ./installboot /.bootsd bootxx  root-disk
   ```

   E.g.: *root-disk* could be `/dev/rsd0a`.
9. Sync the file systems:

   ```
   # sync
   ```

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

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

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

Compatibility Issues With Previous NetBSD Releases

Users upgrading from previous versions of NetBSD may wish to bear the following problems and compatibility issues in mind when upgrading to NetBSD 2.0.2.

Issues affecting an upgrade from NetBSD 1.6

The following issues can generally be resolved by extracting the **etc** set into a temporary directory and running *postinstall*:

```
mkdir /tmp/upgrade
cd /tmp/upgrade
pax -zrpe -f /path/to/etc.tgz
./etc/postinstall -s `pwd` check
./etc/postinstall -s `pwd` fix
```

Issues fixed by *postinstall*:

- Various files in `/etc` need upgrading. These include:
  - `/etc/defaults/*`
  - `/etc/mtree/*`
  - `/etc/daily`
  - `/etc/weekly`
  - `/etc/monthly`
  - `/etc/security`
  - `/etc/rc.subr`
  - `/etc/rc`
  - `/etc/rc.shutdown`
  - `/etc/rc.d/*`
- The following files are now obsolete: `/etc/rc.d/fsck.sh` `/etc/rc.d/gated` `/etc/rc.d/kerberos` `/etc/rc.d/NETWORK` `/etc/rc.d/systemfs` `/etc/rc.d/xntpd` and `/etc/rc.d/ypset`.
- The users and groups 'smmsp', needs to be created, 'news' is no longer part of the system.

The following issues need to be resolved manually:

- `postfix`(8) configuration files require upgrading.

```
cd /usr/share/examples/postfix
cp post-install postfix-files postfix-script /etc/postfix
postfix check
```

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.

- The Internet Systems Consortium, Inc. for hosting the NetBSD FTP, CVS, AnonCVS, mail, mail archive, GNATS, SUP, Rsync and WWW servers.
- The Internet Research Institute in Japan for hosting the server which runs the CVSweb interface to the NetBSD source tree.
- The Helsinki University of Technology in Finland for hosting the NetBSD backup CVS and backup server.
- SSH Communications Security in Finland for operating the backup server.
- 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.
- 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:

  AboveNet Communications, Inc.
  Advanced System Products, Inc.
  Alex Poylisher
  Alistair Crooks
  Andrew Brown
  Atsushi YOKOYAMA
  Avalon Computer Systems
  Bay Area Internet Solutions
  Ben Collver
  Bill Coldwell
  Bill Sommerfeld
  Brad Salai
  Brains Corporation, Japan
  Brian Carlstrom
  Brian McGroarty
  Canada Connect Corporation
  Castor Fu
  Central Iowa (Model) Railroad
  Charles Conn
  Charles D. Cranor
  Charles M. Hannum
  Chris Legrow
  Christer O. Andersson
  Christopher g. Demetriou
  Christos Zoulas
  Chuck Silvers
  Co-operative Research Centre for Enterprise Distributed
  Curt Sampson
  Dave Burgess
  Dave Rand
  David Brownlee
  Demon Internet, UK
  Derek Fellion

Digital Equipment Corporation
Distributed Processing Technology
Douglas J. Trainor
Easynet, UK
Ed Braaten
Edward Richley
Eric and Rosemary Spahr
Free Hardware Foundation
Greg Gingerich
Guenther Grau
Harald Koerfgen
Harry McDonald
Heiko W. Rupp
Herb Peyerl
Hubert Feyrer
Innovation Development Enterprises of America
Internet Software Consortium
James Chacon
Jan Joris Vereijken
Jason Birnschein
Jason Brazile
Jason R. Thorpe
Jim Wise
John Kohl
Jonathan P. Kay
Jordan K. Hubbard
Kenneth Alan Hornstein
Kevin Keith Woo
Kimmo Suominen
Krister Waldfridsson
Lex Wennmacher
LinuxFest Northwest
Luke Mewburn
MS Macro System GmbH, Germany
Mark Brinicombe
Mark S. Thomas
Mason Loring Bliss
Mattias Karlsson
Michael Graff
Michael L. Hitch
Michael Richardson
Michael Thompson
Michael W. James
Mike Price
Neil J. McRae
Noah M. Keiserman
Norman R. McBride
Numerical Aerospace Simulation Facility, NASA Ames Research
Oliver Cahagne
Perry E. Metzger
Petri T. Koistinen
Piermont Information Systems Inc.

Precedence Technologies Ltd
Ralph Campbell
Reinoud Zandijk
Richard Nelson
Rob Windsor
Ross Harvey
SDF Public Access Unix, Inc. 501(c)(7)
Salient Systems Inc.
Scott Ellis
Scott Kaplan
Simon Burge
Soren Jacobsen
Soren Jorvang
Steve Allen
Steve Wadlow
SunROOT# Project
Ted Lemon
Ted Spradley
Thor Lancelot Simon
Tim Law
Tom Coulter
Toru Nishimura
VMC Harald Frank, Germany
Warped Communications, Inc.
Wasabi Systems
Whitecross Database Systems Ltd.
William Gnadt
Worria Web Hosting

(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:

Allen Briggs	`briggs@NetBSD.org`
Frank van der Linden	`fvdl@NetBSD.org`
Luke Mewburn	`lukem@NetBSD.org`
Matt Thomas	`matt@NetBSD.org`
Christos Zoulas	`christos@NetBSD.org`

The portmasters (and their ports):

Allen Briggs	briggs@NetBSD.org	**sandpoint**
Anders Magnusson	ragge@NetBSD.org	**vax**
Andrey Petrov	petrov@NetBSD.org	**sparc64**
Ben Harris	bjh21@NetBSD.org	**acorn26**
Chris Gilbert	chris@NetBSD.org	**cats**
Christian Limpach	cl@NetBSD.org	**xen**
Eduardo Horvath	eeh@NetBSD.org	**evbppc**
Frank van der Linden	fvdl@NetBSD.org	**amd64**
Frank van der Linden	fvdl@NetBSD.org	**i386**
IWAMOTO Toshihiro	toshii@NetBSD.org	**hpcarm**
Ichiro Fukuhara	ichiro@NetBSD.org	**hpcarm**
Ignatios Souvatzis	is@NetBSD.org	**amiga**
Izumi Tsutsui	tsutsui@NetBSD.org	**hp300**
Izumi Tsutsui	tsutsui@NetBSD.org	**news68k**
Jason Thorpe	thorpej@NetBSD.org	**algor**
Jason Thorpe	thorpej@NetBSD.org	**shark**
Jeremy Cooper	jeremy@NetBSD.org	**sun3**
Jonathan Stone	jonathan@NetBSD.org	**pmax**
Julian Coleman	jdc@NetBSD.org	**atari**
Jun-ichiro itojun Hagino	itojun@NetBSD.org	**evbsh3**
Jun-ichiro itojun Hagino	itojun@NetBSD.org	**mmeye**
Kazuki Sakamoto	sakamoto@NetBSD.org	**bebox**
Lennart Augustsson	augustss@NetBSD.org	**pmppc**
Marcus Comstedt	marcus@NetBSD.org	**dreamcast**
Martin Husemann	martin@NetBSD.org	**sparc64**
Matt DeBergalis	deberg@NetBSD.org	**next68k**
Matt Fredette	fredette@NetBSD.org	**hp700**
Matt Fredette	fredette@NetBSD.org	**sun2**
Matt Thomas	matt@NetBSD.org	**alpha**
Matt Thomas	matt@NetBSD.org	**netwinder**
Matthias Drochner	drochner@NetBSD.org	**cesfic**
NISHIMURA Takeshi	nsmrtks@NetBSD.org	**x68k**
NONAKA Kimihiro	nonaka@NetBSD.org	**prep**
Nathan Williams	nathanw@NetBSD.org	**sun3**
Noriyuki Soda	soda@NetBSD.org	**arc**
Paul Kranenburg	pk@NetBSD.org	**sparc**
Phil Nelson	phil@NetBSD.org	**pc532**
Reinoud Zandijk	reinoud@NetBSD.org	**acorn32**
Ross Harvey	ross@NetBSD.org	**alpha**
S(/oren Jørvang	soren@NetBSD.org	**cobalt**
S(/oren Jørvang	soren@NetBSD.org	**sgimips**
Scott Reynolds	scottr@NetBSD.org	**mac68k**
Shin Takemura	takemura@NetBSD.org	**hpcmips**
Simon Burge	simonb@NetBSD.org	**evbmips**
Simon Burge	simonb@NetBSD.org	**evbppc**
Simon Burge	simonb@NetBSD.org	**pmax**
Simon Burge	simonb@NetBSD.org	**sbmips**

Steve Woodford	scw@NetBSD.org	**evbsh5**
Steve Woodford	scw@NetBSD.org	**mvme68k**
Steve Woodford	scw@NetBSD.org	**mvmeppc**
Tohru Nishimura	nisimura@NetBSD.org	**luna68k**
Tsubai Masanari	tsubai@NetBSD.org	**macppc**
Tsubai Masanari	tsubai@NetBSD.org	**newsmips**
UCHIYAMA Yasushi	uch@NetBSD.org	**hpcsh**
UCHIYAMA Yasushi	uch@NetBSD.org	**playstation2**
Wayne Knowles	wdk@NetBSD.org	**mipsco**
Wolfgang Solfrank	ws@NetBSD.org	**ofppc**

The NetBSD 2.0.2 Release Engineering team:

Grant Beattie	grant@NetBSD.org
Erik Berls	cyber@NetBSD.org
James Chacon	jmc@NetBSD.org
Julian Coleman	jdc@NetBSD.org
Håvard Eidnes	he@NetBSD.org
Jun-ichiro itojun Hagino	itojun@NetBSD.org
SAITOH Masanobu	msaitoh@NetBSD.org
Luke Mewburn	lukem@NetBSD.org
Matthias Scheler	tron@NetBSD.org
Curt Sampson	cjs@NetBSD.org
Jim Wise	jwise@NetBSD.org

NetBSD Developers:

Nathan Ahlstrom	nra@NetBSD.org
Steve Allen	wormey@NetBSD.org
Jukka Andberg	jandberg@NetBSD.org
Julian Assange	proff@NetBSD.org
Lennart Augustsson	augustss@NetBSD.org
Christoph Badura	bad@NetBSD.org
Bang Jun-Young	junyoung@NetBSD.org
Dieter Baron	dillo@NetBSD.org
Robert V. Baron	rvb@NetBSD.org
Grant Beattie	grant@NetBSD.org
Jason Beegan	jtb@NetBSD.org
Erik Berls	cyber@NetBSD.org
Hiroyuki Bessho	bsh@NetBSD.org
John Birrell	jb@NetBSD.org
Mason Loring Bliss	mason@NetBSD.org
Charles Blundell	cb@NetBSD.org
Rafal Boni	rafal@NetBSD.org
Manuel Bouyer	bouyer@NetBSD.org
John Brezak	brezak@NetBSD.org
Allen Briggs	briggs@NetBSD.org
Mark Brinicombe	mark@NetBSD.org
Aaron Brown	abrown@NetBSD.org

Andrew Brown	atatat@NetBSD.org
David Brownlee	abs@NetBSD.org
Frederick Bruckman	fredb@NetBSD.org
Jon Buller	jonb@NetBSD.org
Simon Burge	simonb@NetBSD.org
Robert Byrnes	byrnes@NetBSD.org
D'Arcy J.M. Cain	darcy@NetBSD.org
Dave Carrel	carrel@NetBSD.org
Daniel Carosone	dan@NetBSD.org
James Chacon	jmc@NetBSD.org
Bill Coldwell	billc@NetBSD.org
Julian Coleman	jdc@NetBSD.org
Ben Collver	ben@NetBSD.org
Jeremy Cooper	jeremy@NetBSD.org
Chuck Cranor	chuck@NetBSD.org
Alistair Crooks	agc@NetBSD.org
Aidan Cully	aidan@NetBSD.org
Johan Danielsson	joda@NetBSD.org
John Darrow	jdarrow@NetBSD.org
Matt DeBergalis	deberg@NetBSD.org
Rob Deker	deker@NetBSD.org
Chris G. Demetriou	cgd@NetBSD.org
Tracy Di Marco White	gendalia@NetBSD.org
Jaromír Dolecek	jdolecek@NetBSD.org
Andy Doran	ad@NetBSD.org
Roland Dowdeswell	elric@NetBSD.org
Emmanuel Dreyfus	manu@NetBSD.org
Matthias Drochner	drochner@NetBSD.org
Jun Ebihara	jun@NetBSD.org
Håvard Eidnes	he@NetBSD.org
Stoned Elipot	seb@NetBSD.org
Enami Tsugutomo	enami@NetBSD.org
Bernd Ernesti	veego@NetBSD.org
Erik Fair	fair@NetBSD.org
Gavan Fantom	gavan@NetBSD.org
Hubert Feyrer	hubertf@NetBSD.org
Jason R. Fink	jrf@NetBSD.org
Matt Fredette	fredette@NetBSD.org
Thorsten Frueauf	frueauf@NetBSD.org
Castor Fu	castor@NetBSD.org
Ichiro Fukuhara	ichiro@NetBSD.org
Quentin Garnier	cube@NetBSD.org
Thomas Gerner	thomas@NetBSD.org
Simon J. Gerraty	sjg@NetBSD.org
Justin Gibbs	gibbs@NetBSD.org
Chris Gilbert	chris@NetBSD.org
Eric Gillespie	epg@NetBSD.org
Adam Glass	glass@NetBSD.org
Michael Graff	explorer@NetBSD.org

Brian C. Grayson	bgrayson@NetBSD.org
Matthew Green	mrg@NetBSD.org
Andreas Gustafsson	gson@NetBSD.org
Jun-ichiro itojun Hagino	itojun@NetBSD.org
Juergen Hannken-Illjes	hannken@NetBSD.org
Charles M. Hannum	mycroft@NetBSD.org
Ben Harris	bjh21@NetBSD.org
Ross Harvey	ross@NetBSD.org
Eric Haszlakiewicz	erh@NetBSD.org
John Hawkinson	jhawk@NetBSD.org
HAYAKAWA Koichi	haya@NetBSD.org
John Heasley	heas@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
Eduardo Horvath	eeh@NetBSD.org
Nick Hudson	skrll@NetBSD.org
Shell Hung	shell@NetBSD.org
Martin Husemann	martin@NetBSD.org
Dean Huxley	dean@NetBSD.org
Love Hörnquist Åstrand	lha@NetBSD.org
Bernardo Innocenti	bernie@NetBSD.org
Tetsuya Isaki	isaki@NetBSD.org
ITOH Yasufumi	itohy@NetBSD.org
IWAMOTO Toshihiro	toshii@NetBSD.org
Matthew Jacob	mjacob@NetBSD.org
Soren Jacobsen	snj@NetBSD.org
Lonhyn T. Jasinskyj	lonhyn@NetBSD.org
Darrin Jewell	dbj@NetBSD.org
Chris Jones	cjones@NetBSD.org
Søren Jørvang	soren@NetBSD.org
Takahiro Kambe	taca@NetBSD.org
Antti Kantee	pooka@NetBSD.org
Masanori Kanaoka	kanaoka@NetBSD.org
Mattias Karlsson	keihan@NetBSD.org
KAWAMOTO Yosihisa	kawamoto@NetBSD.org
Mario Kemper	magick@NetBSD.org
Min Sik Kim	minskim@NetBSD.org
Thomas Klausner	wiz@NetBSD.org
Klaus Klein	kleink@NetBSD.org
Wayne Knowles	wdk@NetBSD.org
Takayoshi Kochi	kochi@NetBSD.org
John Kohl	jtk@NetBSD.org
Daniel de Kok	daniel@NetBSD.org
Paul Kranenburg	pk@NetBSD.org
Martti Kuparinen	martti@NetBSD.org
Kevin Lahey	kml@NetBSD.org

Johnny C. Lam	jlam@NetBSD.org
Martin J. Laubach	mjl@NetBSD.org
Greg Lehey	grog@NetBSD.org
Ted Lemon	mellon@NetBSD.org
Christian Limpach	cl@NetBSD.org
Frank van der Linden	fvdl@NetBSD.org
Joel Lindholm	joel@NetBSD.org
Mike Long	mikel@NetBSD.org
Warner Losh	imp@NetBSD.org
Tomasz Luchowski	zuntum@NetBSD.org
Federico Lupi	federico@NetBSD.org
Brett Lymn	blymn@NetBSD.org
Paul Mackerras	paulus@NetBSD.org
Anders Magnusson	ragge@NetBSD.org
MAEKAWA Masahide	gehenna@NetBSD.org
David Maxwell	david@NetBSD.org
Dan McMahill	dmcmahill@NetBSD.org
Gregory McGarry	gmcgarry@NetBSD.org
Jared D. McNeill	jmcneill@NetBSD.org
Neil J. McRae	neil@NetBSD.org
Perry Metzger	perry@NetBSD.org
Simas Mockevicius	symka@NetBSD.org
Juan Romero Pardines	xtraeme@NetBSD.org
Julio M. Merino Vidal	jmmv@NetBSD.org
Minoura Makoto	minoura@NetBSD.org
Luke Mewburn	lukem@NetBSD.org
der Mouse	mouse@NetBSD.org
Joseph Myers	jsm@NetBSD.org
Ken Nakata	kenn@NetBSD.org
Takeshi Nakayama	nakayama@NetBSD.org
Phil Nelson	phil@NetBSD.org
Bob Nestor	rnestor@NetBSD.org
NISHIMURA Takeshi	nsmrtks@NetBSD.org
Tohru Nishimura	nisimura@NetBSD.org
NONAKA Kimihiro	nonaka@NetBSD.org
Jesse Off	joff@NetBSD.org
Tatoku Ogaito	tacha@NetBSD.org
OKANO Takayoshi	kano@NetBSD.org
Masaru Oki	oki@NetBSD.org
Atsushi Onoe	onoe@NetBSD.org
Greg Oster	oster@NetBSD.org
Jonathan Perkin	sketch@NetBSD.org
Herb Peyerl	hpeyerl@NetBSD.org
Matthias Pfaller	matthias@NetBSD.org
Chris Pinnock	cjep@NetBSD.org
Adrian Portelli	adrianp@NetBSD.org
Dante Profeta	dante@NetBSD.org
Chris Provenzano	proven@NetBSD.org
Niels Provos	provos@NetBSD.org

Michael Rauch	mrauch@NetBSD.org
Marc Recht	recht@NetBSD.org
Darren Reed	darrenr@NetBSD.org
Jeremy C. Reed	reed@NetBSD.org
Tyler R. Retzlaff	rtr@NetBSD.org
Scott Reynolds	scottr@NetBSD.org
Michael Richardson	mcr@NetBSD.org
Tim Rightnour	garbled@NetBSD.org
Gordon Ross	gwr@NetBSD.org
Steve Rumble	rumble@NetBSD.org
Ilpo Ruotsalainen	lonewolf@NetBSD.org
Heiko W. Rupp	hwr@NetBSD.org
David Sainty	dsainty@NetBSD.org
SAITOH Masanobu	msaitoh@NetBSD.org
Kazuki Sakamoto	sakamoto@NetBSD.org
Curt Sampson	cjs@NetBSD.org
Wilfredo Sanchez	wsanchez@NetBSD.org
Ty Sarna	tsarna@NetBSD.org
SATO Kazumi	sato@NetBSD.org
Jan Schaumann	jschauma@NetBSD.org
Matthias Scheler	tron@NetBSD.org
Karl Schilke (rAT)	rat@NetBSD.org
Amitai Schlair	schmonz@NetBSD.org
Konrad Schroder	perseant@NetBSD.org
Lubomir Sedlacik	salo@NetBSD.org
Christopher SEKIYA	sekiya@NetBSD.org
Reed Shadgett	dent@NetBSD.org
John Shannon	shannonjr@NetBSD.org
Tim Shepard	shep@NetBSD.org
Takeshi Shibagaki	shiba@NetBSD.org
Naoto Shimazaki	igy@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
Noriyuki Soda	soda@NetBSD.org
Wolfgang Solfrank	ws@NetBSD.org
SOMEYA Yoshihiko	someya@NetBSD.org
Bill Sommerfeld	sommerfeld@NetBSD.org
Ignatios Souvatzis	is@NetBSD.org
Bill Squier	groo@NetBSD.org
Jonathan Stone	jonathan@NetBSD.org
Bill Studenmund	wrstuden@NetBSD.org
Kevin Sullivan	sullivan@NetBSD.org
SUNAGAWA Keiki	kei@NetBSD.org
Kimmo Suominen	kim@NetBSD.org
Shin Takemura	takemura@NetBSD.org
TAMURA Kent	kent@NetBSD.org

Shin'ichiro TAYA	taya@NetBSD.org
Ian Lance Taylor	ian@NetBSD.org
Matt Thomas	matt@NetBSD.org
Jason Thorpe	thorpej@NetBSD.org
Christoph Toshok	toshok@NetBSD.org
Tsubai Masanari	tsubai@NetBSD.org
Izumi Tsutsui	tsutsui@NetBSD.org
UCHIYAMA Yasushi	uch@NetBSD.org
Masao Uebayashi	uebayasi@NetBSD.org
Shuichiro URATA	ur@NetBSD.org
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
Leo Weppelman	leo@NetBSD.org
Assar Westerlund	assar@NetBSD.org
Todd Whitesel	toddpw@NetBSD.org
Nathan Williams	nathanw@NetBSD.org
Rob Windsor	windsor@NetBSD.org
Dan Winship	danw@NetBSD.org
Jim Wise	jwise@NetBSD.org
Michael Wolfson	mbw@NetBSD.org
Steve Woodford	scw@NetBSD.org
Colin Wood	ender@NetBSD.org
YAMAMOTO Takashi	yamt@NetBSD.org
Yuji Yamano	yyamano@NetBSD.org
Reinoud Zandijk	reinoud@NetBSD.org
Maria Zevenhoven	maria7@NetBSD.org
Christos Zoulas	christos@NetBSD.org

Other contributors:

Dave Burgess	burgess@cynjut.infonet.net
Brian R. Gaeke	brg@dgate.org
Brad Grantham	grantham@tenon.com
Lawrence Kesteloot	kesteloo@cs.unc.edu
Waldi Ravens	waldi@moacs.indiv.nl.net

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.
This product includes software developed by The NetBSD Foundation, Inc. and its contributors.
This product includes software developed for the NetBSD Project. See http://www.netbsd.org/ for information about NetBSD.
This product contains software developed by Ignatios Souvatzis for the NetBSD project.
This product contains software written by Ignatios Souvatzis and Michael L. Hitch for the NetBSD project.
This product contains software written by Michael L. Hitch for the NetBSD project.
This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)

This product includes cryptographic software written by Eric Young (eay@mincom.oz.au)
This product includes software designed by William Allen Simpson.
This product includes software developed at Ludd, University of Lulea, Sweden and its contributors.
This product includes software developed at Ludd, University of Lulea.
This product includes software developed at the Information Technology Division, US Naval Research Laboratory.
This product includes software developed by Berkeley Software Design, Inc.
This product includes software developed by David Jones and Gordon Ross
This product includes software developed by Gordon W. Ross and Leo Weppelman.
This product includes software developed by Hellmuth Michaelis and Joerg Wunsch
This product includes software developed by Internet Research Institute, Inc.
This product includes software developed by Leo Weppelman and Waldi Ravens.
This product includes software developed by Mika Kortelainen
This product includes software developed by Aaron Brown and Harvard University.
This product includes software developed by Adam Ciarcinski for the NetBSD project.
This product includes software developed by Adam Glass and Charles M. Hannum.
This product includes software developed by Adam Glass.
This product includes software developed by Advanced Risc Machines Ltd.
This product includes software developed by Alex Zepeda, and Colin Wood for the NetBSD Projet.
This product includes software developed by Alex Zepeda.
This product includes software developed by Alistair G. Crooks.
This product includes software developed by Alistair G. Crooks. for the NetBSD project.
This product includes software developed by Allen Briggs
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 Berkeley Software Design, Inc.
This product includes software developed by Bill Paul.
This product includes software developed by Bodo Moeller. (If available, substitute umlauted o for oe)
This product includes software developed by Boris Popov.
This product includes software developed by Brad Pepers
This product includes software developed by Bradley A. Grantham.
This product includes software developed by Brini.
This product includes software developed by Causality Limited.
This product includes software developed by Charles D. Cranor and Seth Widoff.
This product includes software developed by Charles D. Cranor and Washington University.
This product includes software developed by Charles D. Cranor, Washington University, and the University of California, Berkeley and its contributors.
This product includes software developed by Charles D. Cranor, Washington University, the University of California, Berkeley and its contributors.
This product includes software developed by Charles D. Cranor.
This product includes software developed by Charles Hannum.
This product includes software developed by Charles M. Hannum, by the University of Vermont and State Agricultural College and Garrett A. Wollman, by William F. Jolitz, and by the University of California, Berkeley, Lawrence Berkeley Laboratory, and its contributors.
This product includes software developed by Charles M. Hannum.
This product includes software developed by Christian E. Hopps, Ezra Story, Kari Mettinen, Markus Wild, Lutz Vieweg and Michael Teske.
This product includes software developed by Christian E. Hopps.
This product includes software developed by Christian Limpach
This product includes software developed by Christopher G. Demetriou for the NetBSD Project.
This product includes software developed by Christopher G. Demetriou.
This product includes software developed by Christos Zoulas.
This product includes software developed by Chuck Silvers.
This product includes software developed by Colin Wood for the NetBSD Project.
This product includes software developed by Colin Wood.
This product includes software developed by Cybernet Corporation and Nan Yang Computer Services Limited
This product includes software developed by Daishi Kato
This product includes software developed by Dale Rahn.
This product includes software developed by Daniel Widenfalk and Michael L. Hitch.
This product includes software developed by Daniel Widenfalk for the NetBSD Project.
This product includes software developed by Darrin B. Jewell
This product includes software developed by David Miller.
This product includes software developed by Dean Huxley.
This product includes software developed by Eduardo Horvath.
This product includes software developed by Eric S. Hvozda.
This product includes software developed by Eric S. Raymond
This product includes software developed by Eric Young (eay@@mincom.oz.au)

This product includes software developed by Eric Young (eay@cryptsoft.com)
This product includes software developed by Eric Young (eay@mincom.oz.au)
This product includes software developed by Ezra Story and by Kari Mettinen.
This product includes software developed by Ezra Story, by Kari Mettinen and by Bernd Ernesti.
This product includes software developed by Ezra Story, by Kari Mettinen, Michael Teske and by Bernd Ernesti.
This product includes software developed by Ezra Story, by Kari Mettinen, and Michael Teske.
This product includes software developed by Ezra Story.
This product includes software developed by Frank van der Linden for the NetBSD Project.
This product includes software developed by Gardner Buchanan.
This product includes software developed by Gary Thomas.
This product includes software developed by Gordon Ross
This product includes software developed by Gordon W. Ross
This product includes software developed by HAYAKAWA Koichi.
This product includes software developed by Harvard University and its contributors.
This product includes software developed by Harvard University.
This product includes software developed by Herb Peyerl.
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 F. Darwin and others.
This product includes software developed by Ian W. Dall.
This product includes software developed by Ichiro FUKUHARA.
This product includes software developed by Ignatios Souvatzis for the NetBSD Project.
This product includes software developed by Internet Initiative Japan Inc.
This product includes software developed by James R. Maynard III.
This product includes software developed by Jared D. McNeill.
This product includes software developed by Jason L. Wright
This product includes software developed by Jason R. Thorpe for And Communications, http://www.and.com/
This product includes software developed by Joachim Koenig-Baltes.
This product includes software developed by Jochen Pohl for The NetBSD Project.
This product includes software developed by Joerg Wunsch
This product includes software developed by John Birrell.
This product includes software developed by John P. Wittkoski.
This product includes software developed by John Polstra.
This product includes software developed by Jonathan R. Stone for the NetBSD Project.
This product includes software developed by Jonathan Stone and Jason R. Thorpe for the NetBSD Project.
This product includes software developed by Jonathan Stone.
This product includes software developed by Jukka Marin.
This product includes software developed by Julian Highfield.
This product includes software developed by Kazuhisa Shimizu.
This product includes software developed by Kazuki Sakamoto.
This product includes software developed by Kenneth Stailey.
This product includes software developed by Kiyoshi Ikehara.
This product includes software developed by Klaus Burkert,by Bernd Ernesti, by Michael van Elst, and by the University of California, Berkeley and its contributors.
This product includes software developed by LAN Media Corporation and its contributors.
This product includes software developed by Leo Weppelman for the NetBSD Project.
This product includes software developed by Leo Weppelman.
This product includes software developed by Lloyd Parkes.
This product includes software developed by Luke Mewburn.
This product includes software developed by Lutz Vieweg.
This product includes software developed by MINOURA Makoto, Takuya Harakawa.
This product includes software developed by Manuel Bouyer.
This product includes software developed by Marc Horowitz.
This product includes software developed by Marcus Comstedt.
This product includes software developed by Mark Brinicombe for the NetBSD project.
This product includes software developed by Mark Brinicombe.
This product includes software developed by Mark Murray
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 Masanobu Saitoh.
This product includes software developed by Masaru Oki.
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 Matt DeBergalis
This product includes software developed by Matthew Fredette.

This product includes software developed by Matthias Pfaller.
This product includes software developed by Michael Graff for the NetBSD Project.
This product includes software developed by Michael Graff.
This product includes software developed by Michael L. Hitch.
This product includes software developed by Michael Shalayeff.
This product includes software developed by Michael Smith.
This product includes software developed by Mike Glover and contributors.
This product includes software developed by Mike Pritchard.
This product includes software developed by Minoura Makoto.
This product includes software developed by Nan Yang Computer Services Limited.
This product includes software developed by Niels Provos.
This product includes software developed by Niklas Hallqvist, Brandon Creighton and Job de Haas.
This product includes software developed by Niklas Hallqvist.
This product includes software developed by Onno van der Linden.
This product includes software developed by Paul Kranenburg.
This product includes software developed by Paul Mackerras.
This product includes software developed by Per Fogelstrom
This product includes software developed by Peter Galbavy.
This product includes software developed by Phase One, Inc.
This product includes software developed by Philip A. Nelson.
This product includes software developed by Philip L. Budne.
This product includes software developed by RiscBSD.
This product includes software developed by Roar Thronæs.
This product includes software developed by Rodney W. Grimes.
This product includes software developed by Roger Hardiman
This product includes software developed by Roland C. Dowdeswell.
This product includes software developed by Rolf Grossmann.
This product includes software developed by Ross Harvey for the NetBSD Project.
This product includes software developed by Ross Harvey.
This product includes software developed by Scott Bartram.
This product includes software developed by Scott Stevens.
This product includes software developed by Shingo WATANABE.
This product includes software developed by Softweyr LLC, the University of California, Berkeley, and its contributors.
This product includes software developed by Soren S. Jorvang.
This product includes software developed by Stephan Thesing.
This product includes software developed by Steve Woodford.
This product includes software developed by Takashi Hamada
This product includes software developed by Takumi Nakamura.
This product includes software developed by Tatoku Ogaito for the NetBSD Project.
This product includes software developed by Terrence R. Lambert.
This product includes software developed by Tetsuya Isaki.
This product includes software developed by Thomas Gerner
This product includes software developed by Thomas Klausner for the NetBSD Project.
This product includes software developed by Tobias Weingartner.
This product includes software developed by Todd C. Miller.
This product includes software developed by Tohru Nishimura and Reinoud Zandijk for the NetBSD Project.
This product includes software developed by Tohru Nishimura for the NetBSD Project.
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 Toru Nishimura.
This product includes software developed by Trimble Navigation, Ltd.
This product includes software developed by WIDE Project and its contributors.
This product includes software developed by Waldi Ravens.
This product includes software developed by Wasabi Systems for Zembu Labs, Inc. http://www.zembu.com/
This product includes software developed by Winning Strategies, Inc.
This product includes software developed by Wolfgang Solfrank.
This product includes software developed by Yasushi Yamasaki
This product includes software developed by Zembu Labs, Inc.
This product includes software developed by the Alice Group.
This product includes software developed by the Center for Software Science at the University of Utah.
This product includes software developed by the Charles D. Cranor, Washington University, University of California, Berkeley and its contributors.
This product includes software developed by the Computer Systems Engineering Group at Lawrence Berkeley Laboratory.
This product includes software developed by the David Muir Sharnoff.
This product includes software developed by the Harvard University and its contributors.
This product includes software developed by the Kungliga Tekniska Högskolan and its contributors.

This product includes software developed by the Network Research Group at Lawrence Berkeley Laboratory.
This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.OpenSSL.org/)
his product includes software developed by the PocketBSD project and its contributors.
This product includes software developed by the RiscBSD kernel team
This product includes software developed by the RiscBSD team.
This product includes software developed by the SMCC Technology Development Group at Sun Microsystems, Inc.
This product includes software developed by the University of California, Berkeley and its contributors, as well as the Trustees of Columbia University.
This product includes software developed by the University of California, Lawrence Berkeley Laboratory and its contributors.
This product includes software developed by the University of California, Lawrence Berkeley Laboratory.
This product includes software developed by the University of Illinois at Urbana and their contributors.
This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.
This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman, by William F. Jolitz, and by the University of California, Berkeley, Lawrence Berkeley Laboratory, and its contributors.
This product includes software developed for the FreeBSD project
This product includes software developed for the NetBSD Project by Bernd Ernesti.
This product includes software developed for the NetBSD Project by Christopher G. Demetriou.
This product includes software developed for the NetBSD Project by Christos Zoulas
This product includes software developed for the NetBSD Project by Emmanuel Dreyfus.
This product includes software developed for the NetBSD Project by Frank van der Linden
This product includes software developed for the NetBSD Project by Ignatios Souvatzis.
This product includes software developed for the NetBSD Project by Jason R. Thorpe.
This product includes software developed for the NetBSD Project by John M. Vinopal.
This product includes software developed for the NetBSD Project by Matthias Drochner.
This product includes software developed for the NetBSD Project by Michael L. Hitch.
This product includes software developed for the NetBSD Project by Perry E. Metzger.
This product includes software developed for the NetBSD Project by Scott Bartram and Frank van der Linden
This product includes software developed for the NetBSD Project by Allegro Networks, Inc., and Wasabi Systems, Inc.
This product includes software developed for the NetBSD Project by Genetec Corporation.
This product includes software developed for the NetBSD Project by Jonathan Stone.
This product includes software developed for the NetBSD Project by Piermont Information Systems Inc.
This product includes software developed for the NetBSD Project by SUNET, Swedish University Computer Network.
This product includes software developed for the NetBSD Project by Shigeyuki Fukushima.
This product includes software developed for the NetBSD Project by Wasabi Systems, Inc.
This product includes software developed under OpenBSD by Per Fogelstrom Opsycon AB for RTMX Inc, North Carolina, USA.
This product includes software developed under OpenBSD by Per Fogelstrom.
This software is a component of "386BSD" developed by William F. Jolitz, TeleMuse.
This software was developed by Holger Veit and Brian Moore for use with "386BSD" and similar operating systems. "Similar operating systems" includes mainly non-profit oriented systems for research and education, including but not restricted to "NetBSD", "FreeBSD", "Mach" (by CMU).
This software includes software developed by the Computer Systems Laboratory at the University of Utah.
This product includes software developed by Computing Services at Carnegie Mellon University (http://www.cmu.edu/computing/).

This product includes software developed by David Jones.
This product includes software developed by Steve C. Woodford.

The End