NAME

INSTALL – Installation procedure for NetBSD/sparc64.

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 1.5 on the *sparc64* platform. It is available in four different formats titled `INSTALL.`*ext*, where *ext* is one of `.ps`, `.html`, `.more`, or `.txt`:

`.ps` PostScript.

`.html` Standard Internet HTML.

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

`.txt` Plain old ASCII.

You are reading the *PostScript* version.

What is NetBSD?

The NetBSD Operating System is a fully functional Open Source UNIX-like operating system derived from the University of California, Berkeley Networking Release 2 (Net/2), 4.4BSD-Lite, and 4.4BSD-Lite2 sources. NetBSD runs on thirty-one different system architectures featuring twelve distinct families of CPUs, and is being ported to more. The NetBSD 1.5 release contains complete binary releases for fifteen different machine types. (The sixteen remaining are not fully supported at this time and are thus not part of the binary distribution. For information on them, please see the NetBSD web site at **http://www.netbsd.org/**)

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

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

Changes Since The Last Release

The NetBSD 1.5 release provides numerous significant functional enhancements, including support for many new devices, integration of hundreds of bug fixes, new and updated kernel subsystems, and many userland enhancements. The result of these improvements is a stable operating system fit for production use that rivals most commercially available systems.

It is impossible to completely summarize over one year of development that went into the NetBSD 1.5 release. Some highlights include:

Kernel

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

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

Networking

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

File system

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

Security

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

System administration and user tools

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

Miscellaneous

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

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

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

NetBSD 1.5 is the first major release for the sparc64!

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 1.5 Release Contents

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

`.../NetBSD-1.5/`

`CHANGES` Changes since earlier NetBSD releases.

`LAST_MINUTE` Last minute changes.

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

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

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

`patches/` Post-release source code patches.

`source/` Source distribution sets; see below.

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

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

gnusrc This set contains the ''gnu'' sources, including the source for the compiler, assembler, groff, and the other GNU utilities in the binary distribution sets.
22.3 MB gzipped, 98.8 MB uncompressed

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

sharesrc This set contains the ''share'' sources, which include the sources for the man pages not associated with any particular program, the sources for the typesettable document set, the dictionaries, and more.
3.3 MB gzipped, 13.2 MB uncompressed

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

NetBSD/sparc64 subdirectory structure

The sparc64-specific portion of the NetBSD 1.5 release is found in the `sparc64` subdirectory of the distribution: `.../NetBSD-1.5/sparc64/`
`INSTALL.html`

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

`binary/`
`kernel/`
`netbsd.GENERIC.gz` A gzipped NetBSD kernel containing code for everything supported in this release.
`sets/` sparc64 binary distribution sets; see below.

`installation/`
`floppy/` sparc64 boot and installation floppies; see below.
`miniroot/` sparc64 miniroot file system image; see below.
`misc/` Miscellaneous sparc64 installation utilities; see installation section, below.
`netboot/` Server boot-file image for diskless machines.

Binary distribution sets

The NetBSD sparc64 binary distribution sets contain the binaries which comprise the NetBSD 1.5 release for the sparc64. There are eight binary distribution sets. The binary distribution sets can be found in the `sparc64/binary/sets` subdirectory of the NetBSD 1.5 distribution tree, and are as follows:

base The NetBSD 1.5 sparc64 **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.
17.1 MB gzipped, 47.0 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.
7.8 MB gzipped, 36.2 MB uncompressed

etc This distribution set contains the system configuration files that reside in `/etc` and in several other places. This set *must* be installed if you are installing the system from scratch, but should *not* be used if you are upgrading. (If you are upgrading, it's recommended that you get a copy of this set and *carefully* upgrade your configuration files by hand.)
0.1 MB gzipped, 0.6 MB uncompressed

games This set includes the games and their manual pages.
3.0 MB gzipped, 7.6 MB uncompressed

kern This set contains a NetBSD/sparc64 1.5 `GENERIC` kernel, named `/netbsd`. You *must* install this distribution set.

man This set includes all of the manual pages for the binaries and other software contained in the **base** set. Note that it does not include any of the manual pages that are included in the other sets.
5.0 MB gzipped, 19.9 MB uncompressed

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

text This set includes NetBSD 's text processing tools, including `groff`(1), all related programs, and their manual pages.
1.3 MB gzipped, 4.6 MB uncompressed

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

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

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

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

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

xserver The Xsun, Xsun24, XsunMono and Xprt servers with man pages. The 64-bit X server current has problems with pixmaps.
? MB gzipped, ? MB uncompressed

The sparc64 binary distribution sets are distributed as gzipped tar files named with the extension **.tgz**, e.g. `base.tgz`.

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

Note: Each directory in the sparc64 binary distribution also has its own checksum files, just as the source distribution does.

NetBSD/sparc64 System Requirements and Supported Devices

Supported machines

- Sbus-based UltraSPARC systems (the Ultra1, Ultra1E, Ultra2, Ultra2E)
- PCI-based UltraSPARC systems (known to work on the Ultra5, Ultra10, Ultra30 and Ultra250, and may work on many other systems)

Unsupported machines

- Chassis-based Enterprise Systems (Ex000, Ex500, E10000)

The minimal configuration requires 16 MB of RAM and ˜60 MB of disk space. To install the entire system requires much more disk space, and to run X or compile the system, more RAM is recommended, as NetBSD with 16 MB of RAM feels like Solaris with 16 MB of RAM - slow. Note that until you have at least 32 MB of RAM, getting more RAM is more important than getting a faster CPU.

Supported devices

- Ethernet devices
  - Sbus LANCE ethernet (*le*)
  - Sbus HME ethernet (*hme*)
  - Sbus BigMac ethernet (*be*)
  - PCI HME ethernet (*hme*)
  - PCI Tulip (*tlp*)
- Video devices
  - Sbus cgsix
- SCSI host controllers
  - ncr53c9x based controllers (*esp*)
  - ncr53c8xx based controllers (*siop*)
  - ISP10x0 based controllers (*isp*)
  - Adaptec AHA-2x4x[U][2][W] cards and onboard PCI designs using the AIC-7770, AIC-7850, AIC-7860, AIC-7870, AIC-7880 and AIC-789x chipsets.
- PCI IDE host controllers
  - CMD Tech PCI064[3689] IDE Controllers (*pciide*)

    Note: Access to hard disks on the secondary PCIIDE channel currently do not work, but CD-ROM does.
- Serial devices
  - Sbus z8530 serial (*zs*)
- Parallel devices
  - Sbus parallel ports (*bpp*)
  - PCI/Ebus parallel ports (*lpt*)

There are a large number of untested PCI drivers that have never been tested on UltraSPARC PCI systems that may 'just work'.

Unsupported devices

- Floppy drives
- Audio devices
  - Sbus & PCI based system audio (*audiocs*)
- Serial devices
  - PCI based systems (*sab*)
- Keyboard & Mouse devices
  - PCI based systems
- Video devices
  - Creator3D framebuffers (*ffb*)
  - General PCI VGA support
- SCSI host controllers
  - ESP366 based controllers found in Ultra1E and Ultra2 systems (*esp*)

Getting the NetBSD System on to Useful Media

Installation is supported from several media types, including:

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

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

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

Proceed to the instruction on installation.

MS-DOS floppy Count the number of `set_name.`*xx* files that make up the distribution sets you want to install or upgrade. You will need one sixth that number of 1.44 MB floppies.

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

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

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

FTP The preparations for this installation/upgrade method are easy; all you need to do is make sure that there's an FTP site from which you can retrieve the NetBSD distribution when you're about to install or upgrade. You need to know the numeric IP address of that site, and, if it's not on a network directly connected to the machine on which you're installing or upgrading NetBSD, you need to know the numeric IP address of the router closest to the NetBSD machine. Finally, you need to know the numeric IP address of the NetBSD machine itself. If you don't have access to a functioning nameserver during installation, the IP address of **ftp.netbsd.org** is `204.152.184.75` (as of October, 2000).

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

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

NFS

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

You need to know the numeric IP address of the NFS server, and, if the server is not on a network directly connected to the machine on which you're installing or upgrading NetBSD, you need to know the numeric IP address of the router closest to the NetBSD machine. Finally, you need to know the numeric IP address of the NetBSD machine itself.

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

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

Tape

To install NetBSD from a tape, you need to make a tape that contains the distribution set files, in 'tar' format.

If you're making the tape on a UNIX-like system, the easiest way to do so is probably something like:

```
# tar -cf tape_device dist_directories
```

where *tape_device* is the name of the tape device that describes the tape drive you're using; possibly `/dev/rst0`, or something similar, but it will vary from system to system. (If you can't figure it out, ask your system administrator.) In the above example, *dist_directories* are the distribution sets' directories, for the distribution sets you wish to place on the tape. For instance, to put the **misc, base,** and **etc** distributions on tape (in order to do the absolute minimum installation to a new disk), you would do the following:

```
# cd .../NetBSD-1.5
# cd sparc64/binary
# tar -cf tape_device misc etc kern
```

Note: You still need to fill in *tape_device* in the example.

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

Upgrade

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

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

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

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

Preparing your System for NetBSD installation

If you're installing NetBSD/sparc64 for the first time it's a good idea to look at the partition sizes of disk you intend installing NetBSD on.

Asumming a classic partition scheme with / (root) and `/usr` file systems, a comfortable size for the NetBSD / partition is about 20 MB; a good initial size for the swap partition is twice the amount of physical memory in your machine (though, unlike SunOS 4.x, there are no restrictions on the size of the swap partition that would render part of your memory unusable). A full binary installation including X11R6 takes about 150 MB in `/usr`.

You cannot use the security modes of the SPARC OpenBoot PROM.

```
ok setenv security-mode none
```

Some network devices (i.e. certain SBus cards) allow a choice between operating on a UTP or a AUI port. The `le` driver supports automatic detection of the port which is actually connected to the wire. If automatic detection is not available or not working properly in your environment, you may have to specify the type connection using the *media* parameter of `ifconfig`(8). During installation, you'll get the opportunity to specify the appropriate medium. Use `10base5/AUI` to select the AUI connector, or `10baseT/UTP` to select the UTP connector.

Installing the NetBSD System

Installing NetBSD is a relatively complex process, but if you have this document in hand it shouldn't be too much trouble.

There are several ways to install NetBSD onto a disk. The easiest way in terms of preliminary setup is to use the NetBSD miniroot that can be booted off your local disk's swap partition. Alternatively, if your UltraSPARC is hooked up in a network you can find a server and arrange for a diskless setup which is a convenient way to install on a machine whose disk does not currently hold a usable operating system (see the section 'Installing NetBSD by using a diskless setup' below). If you have problems with these, it is possible to install NetBSD from Solaris (see the section 'Installing NetBSD using Solaris' below).

Installing NetBSD by using the NetBSD miniroot

The miniroot is a self-contained NetBSD file system holding all utilities necessary to install NetBSD on a local disk. It is distributed as a plain file designed to be transferred to a raw disk partition from which it can be booted using the appropriate OpenBoot PROM command. Usually, the miniroot will be loaded into the swap partition of a disk. If needed, you can use any other unused partition, but remember that the partition will then not available during the installation process.

Loading the miniroot onto your raw partition is simple. On NetBSD as well as Solaris you use a command like:

```
# dd if=miniroot.fs of=/dev/rdsk/c0t0d0s1 bs=4k conv=sync
```

(Here, `/dev/rdsk/c0t0d0s1` is assumed to be your swap partition.) There's a potential problem here if `/dev/rdsk/c0t0d0s1` is actually in use as a swap partition by your currently running system. If you don't have another disk or partition to spare, you can usually get away with running this command anyway

after first booting into single-user mode to ensure a quiet system.

After transferring the miniroot to disk, bring the system down by:

```
# halt
```

Then boot the miniroot by typing the appropriate command at the OpenBoot PROM:

```
ok boot disk:b netbsd -s
```

If you've loaded the miniroot onto some other disk than `sd0` adapt the boot specifier accordingly, e.g.:

```
ok boot disk1:b netbsd -s
```

The monitor boot command will cause the NetBSD kernel contained in the miniroot image to be booted. After the initial probe messages you'll be asked to start the install or upgrade procedure. Proceed to the section **Running the installation scripts** below.

Installing NetBSD by using a diskless setup

First, you must setup a diskless client configuration on a server. 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, you'll have to consult documentation that came with it. (On SunOS systems, `add_client`(8) is a good start.)

Your SPARCstation expects to be able to download a second stage bootstrap program via TFTP after having acquired its IP address through RevARP when instructed to boot 'over the net'. It will look for a filename composed of the machine's IP address followed by the machine's architecture, separated by a period. For example, a sun4c machine which has been assigned IP address 130.115.144.11, will make an TFTP request for `8273900B.SUN4U`. Normally, this file is a symbolic link to an appropriate second-stage boot 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). You can find the boot program in `/usr/mdec/ofwboot.net` in the NetBSD/sparc64 distribution.

Note: The `/usr/mdec/ofwboot` does not know about netbooting.

After the boot program has been loaded into memory and given control by the OpenBoot PROM, it starts locating the machine's remote root directory through the BOOTPARAM protocol. First a BOOTPARAM WHOAMI request is broadcast on the local net. The answer to this request (if it comes in) contains the client's name. This name is used in next step, a BOOTPARAM GETFILE request - sent to the server that responded to the WHOAMI request - requesting the name and address of the machine that will serve the client's root directory, as well as the path of the client's root on that server.

Finally, this information (if it comes in) is used to issue a REMOTE MOUNT request to the client's root file system server, asking for an NFS file handle corresponding to the root file system. If successful, the boot program starts reading from the remote root file system in search of the kernel which is then read into memory.

As noted above in the section **Preparing your System for** NetBSD Installation you have several options when choosing a location to store the installation filesets. However, the easiest way is to put the `*.tgz` files you want to install into the root directory for your client on the server.

Next, unpack `base.tgz` and `etc.tgz` on the server in the root directory for your machine. If you elect to use a separately NFS-mounted file system for `/usr` with your diskless setup, make sure the `./usr` base files in `base.tgz` end up in the correct location. One way to do this is to temporarily use a loopback mount on the server, re-routing *root*`/usr` to your server's exported NetBSD `/usr` directory. Also put the kernel and the install/upgrade scripts into the root directory.

A few configuration files need to be edited:

root/etc/hosts
: Add the IP addresses of both server and client.

root/etc/myname
: This files contains the client's hostname; use the same name as in <root>/etc/hosts.

root/etc/fstab
: Enter the entries for the remotely mounted file systems. For example:

```
server:/export/root/client / nfs rw 0 0
server:/export/exec/sun4u.netbsd /usr nfs rw 0 0
```

Now you must populate the /dev directory for your client. If your server runs SunOS 4.x, you can simply change your working directory to *root*/dev and run the MAKEDEV script: **sh MAKEDEV all**.

On Solaris systems, **MAKEDEV** can also be used, but there'll be error messages about unknown user and groups. These errors are inconsequential for the purpose of installing NetBSD. However, you may want to correct them if you plan to use the diskless setup regularly. In that case, you may re-run **MAKEDEV** on your NetBSD machine once it has booted.

Boot your workstation from the server by entering the appropriate **boot** command at the monitor prompt. Depending on the OpenBoot PROM version in your machine, this command takes one of the following forms:

```
ok boot net netbsd -s
```

This will boot the NetBSD kernel in single-user mode.

If you use a diskless setup with a separately NFS-mounted /usr file system, mount /usr by hand now:

```
netbsd# mount /usr
```

Note: For miniroot installs, the text editor is vi.

When using disklabel(8) to edit disklabels the -e switch will invoke the editor on the label. The -i switch will run an interactive session.

At this point, it's worth checking the disk label and partition sizes on the disk you want to install NetBSD onto. NetBSD understands SunOS-style disklabels, so if your disk was previously used by SunOS there will be a usable label on it. Use **disklabel -e <disk>** or **disklabel -i <disk>** (where *<disk>* is the device name assigned by the NetBSD kernel, e.g. sd0) to view and modify the partition sizes. See the section **Preparing your System for** NetBSD Installation above for suggestions about disk partition sizes.

If you are installing on a SCSI disk that does *not* have a SunOS or NetBSD label on it, you may still be able to use disklabel(8) but you'll have to create all partitions from scratch. If your disk is listed in /etc/disktab, you may use the entry (which in most cases only defines a 'c' partition to describe the whole disk) to put an initial label on the disk. Then proceed with **disklabel -e <disk>** **disklabel -i <disk>** to create a partition layout that suits your needs.

Note: Because of the built-in compatibility with SunOS-style labels, *Always make sure all your partitions start and end on cylinder boundaries.*

Here follows an example of what you'll see while in the dislabel editor. Do not touch any of the parameters except for the 'label:' entry and the actual partition size information at the bottom (the lines starting with 'a:', 'b:',

The size and offset fields are given in sector units. Be sure to make these numbers multiples of the of the number of sectors per cylinder: the kernel might be picky about these things, but aside from this you'll have the least chance of wasting disk space. Partitions on which you intend to have a mountable file system, should be given fstype 4.2BSD. Remember, the 'c' partition should describe The whole disk and typically

does not require editing. The '`(Cyl. x - y)`' info that appears after the hash '`#`' character is treated as a comment and need not be filled in when altering partitions.

Note: The line containing '`8 partitions:`' is best left alone, even if you define less then eight partitions. If this line displays a different number and the program complains about it (after you leave the editor), then try setting it to '`8 partitions:`'.

Sample disklabel screen

```
netbsd# disklabel sd2
# /dev/rsd2c:
type: SCSI
disk: SCSI disk
label: Hold Your Breath
flags:
bytes/sector: 512
sectors/track: 64
tracks/cylinder: 7
sectors/cylinder: 448
cylinders: 1429
rpm: 3600
interleave: 1
trackskew: 0
cylinderskew: 0
headswitch: 0           # milliseconds
track-to-track seek: 0  # milliseconds
drivedata: 0

8 partitions:
#        size   offset    fstype [fsz bsz cpg]
  a:    50176        0    4.2BSD   0   0   0   # (Cyl.    0 - 111)
  b:    64512    50176      swap                # (Cyl.  112 - 255)
  c:   640192        0   unknown                # (Cyl.    0 - 1428)
  d:   525504   114688    4.2BSD   0   0   0   # (Cyl.  256 - 1428)
```

To begin installation or the upgrade procedure, run sysinst:

```
netbsd# sysinst
```

Running sysinst

The sysinst program will do most of the work of transferring the system from the distribution sets onto your disk. You will frequently be asked for confirmation before sysinst proceeds with each phase of the installation process.

Occasionally, you will have to provide a piece of information such as the name of the disk you want to install on or IP addresses and domain names you want to assign. If your system has more than one disk, you may want to look at the output of the `dmesg`(8) command to see how your disks have been identified by the kernel.

The installation script goes through the following phases:

- determination of the disk to install NetBSD on

- checking of the partition information on the disk
- setting of the local timezone
- creating and mounting the NetBSD file systems
- setup of IP configuration
- extraction of the distribution tar files
- installation of boot programs

Installing NetBSD using Solaris

These instructions were kindly contributed by Murray Stokely <murray@osd.bsdi.com>

Preparing the disk

The first step is to format and label the disk that you would like to use with NetBSD. This can be accomplished with the format command in Solaris. The format command should allow you to create disk slices and write a disklabel. You will probably at least want to create a root partition and a swap partition, plus potentially a /usr or /var partitions, but of course you can layout the disk however you see fit.

```
partition> pr
Current partition table (original):
Total disk cylinders available: 8186 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders         Size            Blocks
0       root    wm       0 - 1923        1.00GB    (1924/0/0) 2097160
1       swap    wu    1924 - 2863      500.29MB    (940/0/0)  1024600
2     backup    wu       0 - 8185        4.25GB    (8186/0/0) 8922740
3 unassigned    wm       0               0         (0/0/0)          0
4        usr    wm    2864 - 8184        2.77GB    (5321/0/0) 5799890
5 unassigned    wm       0               0         (0/0/0)          0
6     backup    wm       0 - 8185        4.25GB    (8186/0/0) 8922740
7 unassigned    wm       0               0         (0/0/0)          0
```

After your disk has been labelled you need to create filesystems on your slices. The Solaris newfs command will create ffs filesystems that can be used by NetBSD. You should create filesystems for all of your slices except for root with a command similar to below, given we have root on c1t2d0s0 and /usr on c1t2d0s4.

```
# newfs /dev/dsk/c1t2d0s0
# newfs /dev/dsk/c1t2d0s4
```

Installing NetBSD Software

You should now mount your NetBSD root and `/usr partitions under` Solaris so that you can populate the filesystems with NetBSD NetBSD binaries.

```
# mount /dev/dsk/c1t2d0s0 /mnt
# mkdir /mnt/usr
# mount /dev/dsk/c1t2d0s4 /mnt/usr
```

First, you should uncompress all of the binary distribution sets for sparc64 into a temporary directory and then extract them into the filesystems you just mounted.

```
# cd ~/netbsd/binary/sets
# gunzip *.tar.gz
# mkdir ~/netbsd/temp
# cd ~/netbsd/temp
```

```
# echo ~/netbsd/binary/sets/*.tar | (cd /mnt; xargs -n1 pax -rpe)
```

Now you should copy the NetBSD kernel and second stage bootloader into your new NetBSD root partition and install the bootblocks using Solaris's installboot command.

```
# cp ~/netbsd/binary/kernel/netbsd.GENERIC /mnt
# cp /mnt/netbsd.GENERIC /mnt/netbsd
# cp ~/netbsd/installation/misc/ofwboot /mnt
# installboot ~/netbsd/installation/misc/bootblk /dev/c1t2d0s0
```

Creating NetBSD Device Nodes under Solaris

Now you will need to create a minimum set of device nodes so that NetBSD can boot correctly. You should create all of the devices listed in the 'std' section of NetBSD 's `/dev/MAKEDEV`. You can use the Solaris version of mknod to create device nodes but you must be careful to use numeric group id's since the groups are numbered differently between the two systems.

```
# mknod console    c 0 0
# mknod tty        c 2 0   ; chmod 666 tty
# mknod kmem       c 3 1   ; chmod 640 kmem ; chgrp 2 kmem
# mknod mem        c 3 0   ; chmod 640 mem ; chgrp 2 mem
# mknod null       c 3 2   ; chmod 666 null
# mknod zero       c 3 12  ; chmod 666 zero
# mknod eeprom     c 3 11  ; chmod 640 eeprom ; chgrp 2 eeprom
# mknod openprom   c 70 0  ; chmod 640 openprom;chgrp 2 openprom
# mknod drum       c 7 0   ; chmod 640 drum ; chgrp 2 drum
# mknod klog       c 16 0  ; chmod 600 klog
# mknod stdin      c 24 0  ; chmod 666 stdin
# mknod stdout     c 24 1  ; chmod 666 stdout
# mknod stderr     c 24 2  ; chmod 666 stderr
# mknod fb         c 22 0  ; chmod 666 fb
# mknod mouse      c 13 0  ; chmod 666 mouse
# mknod kbd        c 29 0  ; chmod 666 kbd
```

You also must create device nodes for the disk devices that you are installing NetBSD onto, again you can use NetBSD 's `/dev/MAKEDEV` as a reference.

```
# mknod sd0a       b 7 0
# mknod sd0b       b 7 1
# mknod sd0c       b 7 2
# mknod sd0d       b 7 3
# mknod sd0e       b 7 4
# mknod sd0f       b 7 5
# mknod sd0g       b 7 6
# mknod sd0h       b 7 7
# mknod sd1a       b 7 8
# mknod sd1b       b 7 9
# mknod sd1c       b 7 10
# mknod sd1d       b 7 11
# mknod sd1e       b 7 12
# mknod sd1f       b 7 13
# mknod sd1g       b 7 14
# mknod sd1h       b 7 15
# mknod rsd0a      c 17 0
# mknod rsd0b      c 17 1
```

```
# mknod rsd0c      c 17 2
# mknod rsd0d      c 17 3
# mknod rsd0e      c 17 4
# mknod rsd0f      c 17 5
# mknod rsd0g      c 17 6
# mknod rsd0h      c 17 7
# mknod rsd1a      c 17 8
# mknod rsd1b      c 17 9
# mknod rsd1c      c 17 10
# mknod rsd1d      c 17 11
# mknod rsd1e      c 17 12
# mknod rsd1f      c 17 13
# mknod rsd1g      c 17 14
# mknod rsd1h      c 17 15
# chgrp 5 *sd[0-1][a-h]
# chmod 640 *sd[0-1][a-h]
```

For IDE disks, use 12 instead of 7 and 26 instead of 17, for `wd0a` and `rwd0a`, etc.

Configuring the NetBSD system (still under Solaris)

You will now need to configure some of the files in `/mnt/etc` to allow the system to work properly. In particular, you will need to modify `/etc/rc.conf` or else you will only get to single user mode. Read through the examples in `/etc/defaults/rc.conf` and override any settings in `/etc/rc.conf`. You will also need to create a `/etc/fstab` file to tell the system which slice to mount as `/usr`, etc. You should follow the examples in `/etc/fstab.wd` or `/etc/fstab.sd` depending on whether you are using IDE or SCSI disks. You may also want to setup your `/etc/resolv.conf` file for name services and your `/etc/hosts file` if you are confident that you will get networking setup on your first try.

Booting NetBSD for the first time

Now it is time to boot NetBSD for the first time. Initially we'd suggest you **`boot netbsd -bs`**, then try multiuser after that. If you boot single-user the NetBSD incantation to make the root file system (/) writable is

```
netbsd# mount -u /dev/sd0a /
```

Congratulations, you have successfully installed NetBSD 1.5.

Post installation steps

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

1. Configuring `/etc/rc.conf`

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

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

   and with the root file system (/) mounted read-write. When the system asks you to choose a shell, simply press `RETURN` to get to a prompt. If you are asked for a terminal type, respond with **`vt220`** (or whatever is appropriate for your terminal type) and press `RETURN`. At this point, you need to configure at least one file in the `/etc` directory. Change to the `/etc` directory and take a look at the `/etc/rc.conf` file. Modify it to your tastes, making sure that you set `rc_configured=YES` so that your changes will be enabled and a multi-user boot can proceed. Default values for the various programs can be found in /etc/defaults/rc.conf, where some in-line documentation may be found. More

complete documentation can be found in rc.conf(5).

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

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

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

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

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

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

```
ifconfig_de0="inet myname.my.dom netmask 255.255.255.0"
```

To enable proper hostname resolution, you will also want to add an /etc/resolv.conf file or (if you are feeling a little more adventurous) run named(8). See resolv.conf(5) or named(8) for more information.

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

2. Logging in

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

3. Adding accounts

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

4. The X Window System

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

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

5. Installing third party packages

   If you wish to install any of the software freely available for UNIX-like systems you are strongly advised to first check the NetBSD package system. This automatically handles any changes necessary to make the software run on NetBSD, retrieval and installation of any other packages on which the software may depend, and simplifies installation (and deinstallation), both from source and precompiled binaries.

   - More information on the package system is at
     http://www.netbsd.org/Documentation/software/packages.html
   - A browsable listing of available packages is at
     ftp://ftp.netbsd.org/pub/NetBSD/packages/pkgsrc/README.html

- Precompiled binaries can be found at
  ftp://ftp.netbsd.org/pub/NetBSD/packages/
- Package sources for compiling packages can be obtained by retrieving the file
  ftp://ftp.netbsd.org/pub/NetBSD/NetBSD-current/tar_files/pkgsrc.tar.gz
  They are typically extracted into `/usr/pkgsrc` (though other locations work fine), as with the command:

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

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

6. Misc
   - Edit `/etc/mail/aliases` to forward root mail to the right place (run `newaliases`(1) afterwards.)
   - The `/etc/mail/sendmail.cf` file will almost definitely need to be adjusted; files aiding in this can be found in `/usr/share/sendmail`. See the README file there for more information.
   - Edit `/etc/rc.local` to run any local daemons you use.
   - Many of the `/etc` files are documented in section 5 of the manual; so just invoking

     ```
     # man 5 filename
     ```

     is likely to give you more information on these files.

Upgrading a previously-installed NetBSD System

To upgrade to NetBSD 1.5 from a previous version follow the instructions in the section **Installing NetBSD**, but run sysinst and choose the upgrade option rather than the install option.

The upgrade option will use the existing disk partitions to install the new system in, and also preserves the files in `/etc` by moving them to `/etc.old`.

Compatibility Issues With Previous NetBSD Releases

Users upgrading from previous versions of NetBSD may wish to bear the following problems and compatibility issues in mind when upgrading to NetBSD 1.5.

General issues

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

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

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

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

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

Issues affecting an upgrading from NetBSD 1.4 or later

- named(8) leaks version information

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

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

- sysctl(8) pathname changed

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

- sendmail(8) configuration file pathname changed

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

Using online NetBSD documentation

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

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

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

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

```
# man passwd
```

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

```
# man 5 passwd
```

instead.

If you are unsure of what man page you are looking for, enter **apropos** *subject-word*

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

Administrivia

If you've got something to say, do so! We'd like your input. There are various mailing lists available via the mailing list server at `majordomo@netbsd.org`. To get help on using the mailing list server, send mail to that address with an empty body, and it will reply with instructions.

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

To report bugs, use the `send-pr`(1) command shipped with NetBSD, and fill in as much information about the problem as you can. Good bug reports include lots of details. Additionally, bug reports can be sent by mail to: `netbsd-bugs@netbsd.org`.

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

There are also port-specific mailing lists, to discuss aspects of each port of NetBSD. Use majordomo to find their addresses, or visit **http://www.netbsd.org/MailingLists/**. If you're interested in doing a serious amount of work on a specific port, you probably should contact the 'owner' of that port (listed below).

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

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

Thanks go to

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

  Keith Bostic
  Ralph Campbell
  Mike Karels
  Marshall Kirk McKusick

  for their ongoing work on BSD systems, support, and encouragement.

- Also, our thanks go to:

  Mike Hibler
  Rick Macklem
  Jan-Simon Pendry
  Chris Torek

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

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

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

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

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

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

- The many organisations that provide NetBSD mirror sites.
- Without CVS, this project would be impossible to manage, so our hats go off to Brian Berliner, Jeff Polk, and the various other people who've had a hand in making CVS a useful tool.
- Dave Burgess `burgess@cynjut.infonet.net` has been maintaining the 386BSD/NetBSD/FreeBSD FAQ for quite some time, and deserves to be recognized for it.
- The following individuals and organizations (each in alphabetical order) have made donations or loans of hardware and/or money, to support NetBSD development, and deserve credit for it:

  Steve Allen
  Jason Birnschein
  Mason Loring Bliss
  Jason Brazile
  Mark Brinicombe
  David Brownlee
  Simon Burge
  Dave Burgess
  Ralph Campbell
  Brian Carlstrom
  James Chacon
  Bill Coldwell
  Charles Conn
  Tom Coulter
  Charles D. Cranor
  Christopher G. Demetriou
  Scott Ellis
  Hubert Feyrer
  Castor Fu
  Greg Gingerich
  William Gnadt
  Michael Graff
  Guenther Grau
  Ross Harvey
  Charles M. Hannum
  Michael L. Hitch
  Kenneth Alan Hornstein
  Jordan K. Hubbard
  Søren Jørvang
  Scott Kaplan
  Noah M. Keiserman
  Harald Koerfgen
  John Kohl
  Chris Legrow
  Ted Lemon
  Norman R. McBride
  Neil J. McRae
  Perry E. Metzger
  Toru Nishimura
  Herb Peyerl
  Mike Price
  Dave Rand

Michael Richardson
Heiko W. Rupp
Brad Salai
Chuck Silvers
Thor Lancelot Simon
Bill Sommerfeld
Paul Southworth
Eric and Rosemary Spahr
Ted Spradley
Kimmo Suominen
Jason R. Thorpe
Steve Wadlow
Krister Walfridsson
Jim Wise
Christos Zoulas

AboveNet Communications, Inc.
Advanced System Products, Inc.
Avalon Computer Systems
Bay Area Internet Solutions
Brains Corporation, Japan
Canada Connect Corporation
Co-operative Research Centre for Enterprise Distributed Systems Technology
Demon Internet, UK
Digital Equipment Corporation
Distributed Processing Technology
Easynet, UK
Free Hardware Foundation
Innovation Development Enterprises of America
Internet Software Consortium
MS Macro System GmbH, Germany
Numerical Aerospace Simulation Facility, NASA Ames Research Center
Piermont Information Systems Inc.
Salient Systems Inc.
VMC Harald Frank, Germany
Warped Communications, Inc.
Whitecross Database Systems Ltd.

(If you're not on that list and should be, tell us! We probably were not able to get in touch with you, to verify that you wanted to be listed.)

- Finally, we thank all of the people who've put sweat and tears into developing NetBSD since its inception in January, 1993. (Obviously, there are a lot more people who deserve thanks here. If you're one of them, and would like to mentioned, tell us!)

We are...

(in alphabetical order)

The NetBSD core group:

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

The portmasters (and their ports):

Mark Brinicombe	mark@netbsd.org	**arm32**
Jeremy Cooper	jeremy@netbsd.org	**sun3x**
Ross Harvey	ross@netbsd.org	**alpha**
Jun-ichiro itojun Hagino	itojun@netbsd.org	**sh3**
Ben Harris	bjh21@netbsd.org	**arm26**
Eduardo Horvath	eeh@netbsd.org	**sparc64**
Darrin Jewell	dbj@netbsd.org	**next68k**
Søren Jørvang	soren@netbsd.org	**cobalt**
Søren Jørvang	soren@netbsd.org	**sgimips**
Wayne Knowles	wdk@netbsd.org	**mipsco**
Paul Kranenburg	pk@netbsd.org	**sparc**
Anders Magnusson	ragge@netbsd.org	**vax**
Minoura Makoto	minoura@netbsd.org	**x68k**
Phil Nelson	phil@netbsd.org	**pc532**
Tohru Nishimura	nisimura@netbsd.org	**luna68k**
NONAKA Kimihiro	nonaka@netbsd.org	**prep**
Scott Reynolds	scottr@netbsd.org	**mac68k**
Kazuki Sakamoto	sakamoto@netbsd.org	**bebox**
Noriyuki Soda	soda@netbsd.org	**arc**
Wolfgang Solfrank	ws@netbsd.org	**ofppc**
Ignatios Souvatzis	is@netbsd.org	**amiga**
Jonathan Stone	jonathan@netbsd.org	**pmax**
Shin Takemura	takemura@netbsd.org	**hpcmips**
Jason Thorpe	thorpej@netbsd.org	**alpha**
Jason Thorpe	thorpej@netbsd.org	**hp300**
Tsubai Masanari	tsubai@netbsd.org	**macppc**
Tsubai Masanari	tsubai@netbsd.org	**newsmips**
Izumi Tsutsui	tsutsui@netbsd.org	**news68k**
Frank van der Linden	fvdl@netbsd.org	**i386**
Leo Weppelman	leo@netbsd.org	**atari**
Nathan Williams	nathanw@netbsd.org	**sun3**
Steve Woodford	scw@netbsd.org	**mvme68k**

The NetBSD 1.5 Release Engineering team:

Chris G. Demetriou	cgd@netbsd.org
Havard Eidnes	he@netbsd.org
Ted Lemon	mellon@netbsd.org
John Hawkinson	jhawk@netbsd.org

Perry Metzger	perry@netbsd.org
Curt Sampson	cjs@netbsd.org
Jason Thorpe	thorpej@netbsd.org
Todd Vierling	tv@netbsd.org

Developers and other contributors:

Steve Allen	wormey@netbsd.org
Julian Assange	proff@netbsd.org
Lennart Augustsson	augustss@netbsd.org
Christoph Badura	bad@netbsd.org
Robert V. Baron	rvb@netbsd.org
Erik Berls	cyber@netbsd.org
John Birrell	jb@netbsd.org
Mason Loring Bliss	mason@netbsd.org
Manuel Bouyer	bouyer@netbsd.org
John Brezak	brezak@netbsd.org
Allen Briggs	briggs@netbsd.org
Aaron Brown	abrown@netbsd.org
David Brownlee	abs@netbsd.org
Frederick Bruckman	fredb@netbsd.org
Jon Buller	jonb@netbsd.org
Simon Burge	simonb@netbsd.org
Dave Burgess	burgess@cynjut.infonet.net
Robert Byrnes	byrnes@netbsd.org
D'Arcy J.M. Cain	darcy@netbsd.org
Dave Carrel	carrel@netbsd.org
James Chacon	jmc@netbsd.org
Bill Coldwell	billc@netbsd.org
Julian Coleman	jdc@netbsd.org
Chuck Cranor	chuck@netbsd.org
Aidan Cully	aidan@netbsd.org
Johan Danielsson	joda@netbsd.org
Matt DeBergalis	deberg@netbsd.org
Rob Deker	deker@netbsd.org
Chris G. Demetriou	cgd@netbsd.org
Jaromir Dolecek	jdolecek@netbsd.org
Andy Doran	ad@netbsd.org
Roland Dowdeswell	elric@netbsd.org
Matthias Drochner	drochner@netbsd.org
Jun Ebihara	jun@netbsd.org
Havard Eidnes	he@netbsd.org
Enami Tsugutomo	enami@netbsd.org
Bernd Ernesti	veego@netbsd.org
Erik Fair	fair@netbsd.org
Hubert Feyrer	hubertf@netbsd.org
Thorsten Frueauf	frueauf@netbsd.org
Castor Fu	castor@netbsd.org
Ichiro Fukuhara	ichiro@netbsd.org

Brian R. Gaeke	brg@dgate.org
Thomas Gerner	thomas@netbsd.org
Simon J. Gerraty	sjg@netbsd.org
Justin Gibbs	gibbs@netbsd.org
Adam Glass	glass@netbsd.org
Michael Graff	explorer@netbsd.org
Brad Grantham	grantham@tenon.com
Brian C. Grayson	bgrayson@netbsd.org
Matthew Green	mrg@netbsd.org
Juergen Hannken-Illjes	hannken@netbsd.org
Charles M. Hannum	mycroft@netbsd.org
Eric Haszlakiewicz	erh@netbsd.org
John Hawkinson	jhawk@netbsd.org
HAYAKAWA Koichi	haya@netbsd.org
René Hexel	rh@netbsd.org
Michael L. Hitch	mhitch@netbsd.org
Christian E. Hopps	chopps@netbsd.org
Ken Hornstein	kenh@netbsd.org
Marc Horowitz	marc@netbsd.org
Nick Hudson	skrll@netbsd.org
Martin Husemann	martin@netbsd.org
Dean Huxley	dean@netbsd.org
Bernardo Innocenti	bernie@netbsd.org
ITOH Yasufumi	itohy@netbsd.org
IWAMOTO Toshihiro	toshii@netbsd.org
Matthew Jacob	mjacob@netbsd.org
Lonhyn T. Jasinskyj	lonhyn@netbsd.org
Chris Jones	cjones@netbsd.org
Takahiro Kambe	taca@netbsd.org
Antti Kantee	pooka@netbsd.org
Lawrence Kesteloot	kesteloo@cs.unc.edu
Thomas Klausner	wiz@netbsd.org
Klaus Klein	kleink@netbsd.org
Wayne Knowles	wdk@netbsd.org
John Kohl	jtk@netbsd.org
Kevin Lahey	kml@netbsd.org
Johnny C. Lam	jlam@netbsd.org
Martin J. Laubach	mjl@netbsd.org
Ted Lemon	mellon@netbsd.org
Joel Lindholm	joel@netbsd.org
Mike Long	mikel@netbsd.org
Warner Losh	imp@netbsd.org
Federico Lupi	federico@netbsd.org
Brett Lymn	blymn@netbsd.org
Paul Mackerras	paulus@netbsd.org
David Maxwell	david@netbsd.org
Dan McMahill	dmcmahill@netbsd.org
Gregory McGarry	gmcgarry@netbsd.org
Neil J. McRae	neil@netbsd.org

Perry Metzger	perry@netbsd.org
der Mouse	mouse@netbsd.org
Joseph Myers	jsm@netbsd.org
Ken Nakata	kenn@netbsd.org
Bob Nestor	rnestor@netbsd.org
NONAKA Kimihiro	nonaka@netbsd.org
Masaru Oki	oki@netbsd.org
Atsushi Onoe	onoe@netbsd.org
Greg Oster	oster@netbsd.org
Herb Peyerl	hpeyerl@netbsd.org
Matthias Pfaller	matthias@netbsd.org
Dante Profeta	dante@netbsd.org
Chris Provenzano	proven@netbsd.org
Waldi Ravens	waldi@moacs.indiv.nl.net
Darren Reed	darrenr@netbsd.org
Michael Richardson	mcr@netbsd.org
Tim Rightnour	garbled@netbsd.org
Gordon Ross	gwr@netbsd.org
Heiko W. Rupp	hwr@netbsd.org
SAITOH Masanobu	msaitoh@netbsd.org
Curt Sampson	cjs@netbsd.org
Wilfredo Sanchez	wsanchez@netbsd.org
Ty Sarna	tsarna@netbsd.org
SATO Kazumi	sato@netbsd.org
Matthias Scheler	tron@netbsd.org
Karl Schilke (rAT)	rat@netbsd.org
Konrad Schroder	perseant@netbsd.org
Reed Shadgett	dent@netbsd.org
Tim Shepard	shep@netbsd.org
Takao Shinohara	shin@netbsd.org
Takuya SHIOZAKI	tshiozak@netbsd.org
Chuck Silvers	chs@netbsd.org
Thor Lancelot Simon	tls@netbsd.org
Jeff Smith	jeffs@netbsd.org
Bill Sommerfeld	sommerfeld@netbsd.org
Bill Squier	groo@netbsd.org
Bill Studenmund	wrstuden@netbsd.org
Kevin Sullivan	sullivan@netbsd.org
SUNAGAWA Keiki	kei@netbsd.org
Kimmo Suominen	kim@netbsd.org
Matt Thomas	matt@netbsd.org
Christoph Toshok	toshok@netbsd.org
UCHIYAMA Yasushi	uch@netbsd.org
Shuichiro URATA	ur@netbsd.org
Todd Vierling	tv@netbsd.org
Aymeric Vincent	aymeric@netbsd.org
Paul Vixie	vixie@netbsd.org
Krister Walfridsson	kristerw@netbsd.org
Lex Wennmacher	wennmach@netbsd.org

Assar Westerlund	assar@netbsd.org
Todd Whitesel	toddpw@netbsd.org
Rob Windsor	windsor@netbsd.org
Dan Winship	danw@netbsd.org
Jim Wise	jwise@netbsd.org
Michael Wolfson	mbw@netbsd.org
Colin Wood	ender@netbsd.org

Legal Mumbo-Jumbo

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

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

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

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

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

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

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

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

This product includes software developed by Adam Glass.

This product includes software developed by Alistair G. Crooks.

This product includes software developed by Amancio Hasty and Roger Hardiman.

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

This product includes software developed by Bill Paul.

This product includes software developed by Charles D. Cranor and Washington University.

This product includes software developed by Charles D. Cranor.

This product includes software developed by Charles Hannum, by the University of Vermont and State Agricultural College and Garrett A. Wollman, by William F. Jolitz, and by the University of California, Berkeley, Lawrence Berkeley Laboratory, and its contributors.

This product includes software developed by Charles Hannum.

This product includes software developed by Charles M. Hannum.

This product includes software developed by Chris Provenzano.

This product includes software developed by Christian E. Hopps.

This product includes software developed by Christopher G. Demetriou for the NetBSD Project.

This product includes software developed by Christopher G. Demetriou.

This product includes software developed by Christos Zoulas.

This product includes software developed by David Jones and Gordon Ross.

This product includes software developed by Dean Huxley.

This product includes software developed by Eric S. Hvozda.

This product includes software developed by Ezra Story.

This product includes software developed by Gardner Buchanan.

This product includes software developed by Gordon Ross.

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

This product includes software developed by Gordon W. Ross.

This product includes software developed by Hauke Fath.

This product includes software developed by HAYAKAWA Koichi.

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

This product includes software developed by Herb Peyerl.

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

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

This product includes software developed by Iain Hibbert.

This product includes software developed by Ian W. Dall.

This product includes software developed by Ignatios Souvatzis for the NetBSD Project.

This product includes software developed by Jason R. Thorpe for And Communications, http://www.and.com/.

This product includes software developed by Joachim Koenig-Baltes.

This product includes software developed by Jochen Pohl for The NetBSD Project.

This product includes software developed by John Polstra.

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

This product includes software developed by Jonathan Stone for the NetBSD Project.

This product includes software developed by Jonathan Stone.

This product includes software developed by Julian Highfield.

This product includes software developed by Kenneth Stailey.

This product includes software developed by Leo Weppelman.

This product includes software developed by Lloyd Parkes.

This product includes software developed by Manuel Bouyer.

This product includes software developed by Marc Horowitz.

This product includes software developed by Mark Brinicombe.

This product includes software developed by Mark Tinguely and Jim Lowe.

This product includes software developed by Markus Wild.

This product includes software developed by Martin Husemann and Wolfgang Solfrank.

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

This product includes software developed by Mats O Jansson.

This product includes software developed by Matthias Pfaller.

This product includes software developed by Michael L. Hitch.

This product includes software developed by Niels Provos.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Paul Mackerras.

This product includes software developed by Peter Galbavy.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Rodney W. Grimes.

This product includes software developed by Roland C. Dowdeswell.

This product includes software developed by Rolf Grossmann.

This product includes software developed by Scott Bartram.

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

This product includes software developed by Tatoku Ogaito for the NetBSD Project.

This product includes software developed by Terrence R. Lambert.

This product includes software developed by Theo de Raadt and John Brezak.

This product includes software developed by Theo de Raadt.

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

This product includes software developed by TooLs GmbH.

This product includes software developed by Winning Strategies, Inc.

This product includes software developed by Zembu Labs, Inc.

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

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

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

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

This product includes software developed for the FreeBSD project.

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

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

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

This product includes software developed for the NetBSD Project by John M. Vinopal.

This product includes software developed for the NetBSD Project by Matthias Drochner.

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

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

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

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

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

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

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

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

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes software developed by the University of Oregon.

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

This product includes software developed by Internet Initiative Japan Inc.

This product includes software developed by the Alice Group.

This product includes software developed by David Miller.