NAME

INSTALL — Installation procedure for NetBSD/sgimips.

CONTENTS

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 11.1 on the *sgimips* 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 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 many different different system architectures (ports) across a variety of distinct CPU families, and is being ported to more. The NetBSD 11.1 release contains complete binary releases for most of these system architectures, with preliminary support for the others included in source form. For more information please visit **https://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, NetBSD would not exist.

Changes Between The NetBSD 10 and 11 Releases

The NetBSD 11.1 release provides many significant changes, including support for many new devices, hundreds of bug fixes, new and updated kernel subsystems, and numerous userland enhancements. The result of these improvements is a stable operating system fit for production use that rivals most commercially available systems.

One important new feature in this release is the support for extended attributes and access control lists on FFS file systems.

For new installations the installer will by default disable these features, so the file system is compatible with older NetBSD releases (before 10), and allow other operating systems to mount this file systems at least in read-only mode.

If you want a new installed file system to support extended attributes, change the file system type from "FFSv2" to "FFSv2ea" in the partitioning menu. You can also convert file systems later, using the `fsck_ffs`(8) utility. More details are available in *this guide:*
 https://wiki.netbsd.org/tutorials/acls_and_extended_attributes_on_ffs.

If you are upgrading from a version of NetBSD-current please also check the **Compatibility Issues With Previous NetBSD Releases**. It is impossible to completely summarize the massive development that went into the NetBSD 11.1 release. The complete list of changes can be found in
CHANGES:
 https://cdn.NetBSD.org/pub/NetBSD/NetBSD-11.1/CHANGES
CHANGES-11.1:
 https://cdn.NetBSD.org/pub/NetBSD/NetBSD-11.1/CHANGES-11.1

which are also present in the top level directory of the NetBSD 11.1 release tree.

Features to be removed in a later release

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

- `groff`(1). Man pages are now handled with `mandoc`(1), and `groff`(1) can still be found in pkgsrc as `textproc/groff`.
- `pf`(4). This packet filter is obsolete and unmaintained in NetBSD. It will be eventually removed due to possible long-standing security issues and lack of multiprocessor support. New installations should use `npf`(7).

The NetBSD Foundation

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

Sources of NetBSD

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

NetBSD 11.1 Release Contents

The root directory of the NetBSD 11.1 release is organized as follows:

`.../NetBSD-11.1/`

`CHANGES`	Changes between the 10.0 and 11.0 releases.
`CHANGES-11.0`	Changes between the initial 11.0 branch and final release of 11.0.
`CHANGES-11.1`	Changes between the 11.0 and the 11.1 release.
`CHANGES.prev`	Changes in previous NetBSD releases.
`LAST_MINUTE`	Last minute changes and notes about the release.
`README.files`	README describing the distribution's contents.
`images/`	Images (ISO 9660 or USB) for installing NetBSD. Depending on your system, these may be bootable.
`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 11.1 has a binary distribution.

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

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

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

src This set contains all of the base NetBSD 11.1 sources which are not in **gnusrc**, **sharesrc**, or **syssrc**.

syssrc This set contains the sources to the NetBSD 11.1 kernel for all architectures as well as the `config`(1) utility.

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

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

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

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

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

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

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

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

NetBSD/sgimips subdirectory structure

The sgimips-specific portion of the NetBSD 11.1 release is found in the `sgimips` subdirectory of the distribution: `.../NetBSD-11.1/sgimips/`. 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-GENERIC32_IP12.gz`
    A kernel suitable for booting on R3000 Indigo, IRIS 4D/3x and similar machines.
  - `netbsd-GENERIC32_IP12.ecoff.gz`
    GENERIC32_IP2x targeted for being bootable on the above machines with PROMs that do not understand the ELF binary format.
  - `netbsd-GENERIC32_IP2x.gz`
    A kernel suitable for booting on Indy, Indigo 2 and similar machines.
  - `netbsd-GENERIC32_IP2x.ecoff.gz`
    GENERIC32_IP2x targeted for being bootable on older Indigo 2 and possibly Indy machines with PROMs that do not understand the ELF binary format.
  - `netbsd-GENERIC32_IP3x.gz`
    A kernel suitable for booting on O2.
- `sets/` sgimips binary distribution sets; see below.

`installation/`
- `netboot/` NFS-root images

Binary distribution sets

The NetBSD sgimips binary distribution sets contain the binaries which comprise the NetBSD 11.1 release for sgimips. The binary distribution sets can be found in the `sgimips/binary/sets` subdirectory of the NetBSD 11.1 distribution tree, and are as follows:

base The NetBSD 11.1 sgimips **base** binary distribution. You *must* install this distribution set. It contains the base NetBSD utilities that are necessary for the system to run and be minimally functional.

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

debug This distribution set contains debug information for all base system utilities. It is useful when reporting issues with binaries or during development. This set is huge, if the target disk is small, do not install it.

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

games This set includes the games and their manual pages.

kern-GENERIC32_IP2x
This set contains a NetBSD/sgimips 11.1 `GENERIC32_IP12` kernel (for R3000 Indigo, IRIS 4D/3x machines), named `/netbsd`. This set also contains an ECOFF kernel named `/netbsd.ecoff` that your system may need to boot.

kern-GENERIC32_IP2x
This set contains a NetBSD/sgimips 11.1 `GENERIC32_IP2x` kernel (for Indy, Indigo 2 and Challenge S/M machines), named `/netbsd`. This set also contains an ECOFF kernel named `/netbsd.ecoff` that your system may need to boot.

kern-GENERIC32_IP3x
This set contains a NetBSD/sgimips 11.1 `GENERIC32_IP3x` kernel (for O2), named `/netbsd`.

You *must* install one of these distribution sets for your machine.

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

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

modules This set includes kernel modules to add functionality to a running system.

rescue This set includes the statically linked emergency recovery binaries installed in `/rescue`.

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

NetBSD maintains its own set of sources for the X Window System in order to assure tight integration and compatibility. These sources are based on X.Org. 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.

xcomp The extra libraries and include files needed to compile X source code.

xdebug This distribution set contains debug information for all X11 binaries. It is useful when reporting issues with these binaries or during development. This set is huge, if the target disk is small, do not install it.

xfont Fonts needed by the X server and by X clients.

xetc Configuration files for X which could be locally modified.

xserver The X server.

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

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

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

`diskimage/`

`diskimage.gz` FFS disk image of the installation filesystem.

`netboot/`

`diskimage.tgz` Minimal file set necessary for NFS-root installations; to be extracted on the NFS server.

The following are included in the `sgimips/binary/kernel` directory:

`netbsd-GENERIC32_IPxx.ecoff.gz` An ECOFF version of the `GENERIC32_IPxx` kernel (gzipped). This kernel can be used when booting machines whose PROMs do not understand the ELF binary format.

`netbsd-INSTALL32_IPxx.gz` A gzipped `INSTALL32_IPxx` kernel, which includes an embedded ramdisk with the installation tools.

`netbsd-INSTALL32_IPxx.ecoff.gz` An ECOFF version of the `netbsd-INSTALL32_IPxx kernel (also gzipped).`

`netbsd-INSTALL32_IPxx.symbols.gz` Symbols for `netbsd-INSTALL32_IPxx.gz`.

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

NetBSD/sgimips System Requirements and Supported Devices

NetBSD/sgimips is aimed to run on MIPS-based computers manufactured by SGI.

Supported machines

- IP12 R3000 – Indigo, Personal Iris 4D/30, Personal Iris 4D/35
- IP20 R4000/R4400 – Indigo
- IP22 R4000/R4400/R4600 – Challenge M, Challenge S, Indigo 2 and Indy
- IP22 R5000 – Challenge S, Indy

- IP32 R5000 - O2
- IP32 R10000 - O2

Unsupported machines

- IP19 R4000 - Challenge Onyx
- IP21 R8000 - Power Challenge / Power Onyx
- IP25 R10000 - Power Challenge 10000 / Power Onyx 10000
- IP26 R8000 - Indigo 2 R8000
- IP27 R10000/R12000 - O200 / Origin 2000 / Onyx2
- IP28 R10000 - Indigo 2 R10000
- IP30 R10000/R12000 - Octane
- IP35 R12000A - Origin 3000 / Onyx 3000

Supported devices

- Serial ports
  - On-board Zilog Z8530 Dual Uart (IP12, IP20, IP22/24) (*ttyz*) ,
  - On-board 16C550 Serial ports (IP32). (*ttyC*) .
- Ethernet
  - On-board SGI Seeq 80c03 ethernet controller (IP12, IP20, IP22/24) (*sq*) ,
  - Phobos G100/G130/G160 Fast Ethernet adapters (IP20, IP22/IP24) (*tlp*) ,
  - On-board MACE Ethernet controller (IP32) (*mec*) ,
  - Most PCI Ethernet controllers (IP32)
- SCSI
  - On-board Western Digital WD33C93 SCSI controller(s) (IP12, IP20, IP22/24) (*wdsc*) ,
  - On-board Adaptec AIC-7889 SCSI controller (IP32) (*aic*) ,
- Audio
  - On-board HAL2 controller (IP22/24) (*haltwo*)
- Graphics adapters
  - SGI CRIME (Found in O2s) (*crmfb*) ,
  - SGI NG1 "newport" (IP22/24) (*newport*) ,
  - SGI GR2 (various models) (*grtwo*)
- Keyboard and mouse devices
  - PC keyboard/mouse controller (IP22/24) (*pckbc*)
  - zs keyboard/mouse controller (IP12/20) (*zs*)
- Miscellaneous Devices
  - On-board Parallel Port (IP22/24) (*pi1ppc*)

Unsupported devices

- Some graphics adapters
- Video hardware
- Some parallel ports

Getting the NetBSD System on to Useful Media

Installation is supported from several media types, including:

- CD-ROM / DVD / USB stick
- 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 / DVD / USB stick
: Find out where the distribution set files are on the CD-ROM, DVD or USB stick. Likely locations are `binary/sets` and `sgimips/binary/sets`. (You only need to know this if you are mixing installer and installation media from different versions - the installer will know the proper default location for the sets it comes with).

  Proceed to the instructions on installation.

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. If you don't have DHCP available on your network, you will 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.

  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.

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 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 you don't have DHCP available on your network and 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.

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_sets
```

where *tape_device* is the name of the tape device that represents the tape drive you're using. This might be /dev/rst0, or something similar, but it will vary from system to system. In the above example, *dist_sets* is a list of filenames corresponding to the distribution sets that you wish to place on the tape. For instance, to put the **kern-GENERIC, base,** and **etc** distributions on tape (the absolute minimum required for installation), you would do the following:

```
# cd .../NetBSD-11.1
# cd sgimips/binary
# tar -cf tape_device kern-GENERIC.tgz
base.tgz etc.tgz
```

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.

Preparing your System for NetBSD installation

First and foremost, before beginning the installation process, *make sure you have a reliable backup* of any data (e.g., an old IRIX installation) you wish to preserve.

Booting NetBSD/sgimips for the first time is most commonly done from the network. You must set up DHCP, TFTP, and possibly NFS for the installation procedure to be successful. Although newer SGI bootproms can read ELF kernels directly from the network, others require that you boot an ECOFF version of the kernel. Attempting to boot an ELF kernel on a system with a PROM that only supports ECOFF binaries will result in the following message being displayed by the PROM:

```
Illegal f_magic number 0x7f45, expected MIPSELMAGIC or MIPSEBMAGIC.
```

You can convert the standard ELF format kernel into an ECOFF kernel using the objcopy tool.

Note that some older bootproms have an interesting bug in reading the kernel via TFTP. They handle the port number as a signed entity, and can thus not connect to ports >32767. You can work around this problem on the TFTP server (given that it is a NetBSD host) by using "sysctl" to set "net.inet.ip.anonportmin" and "net.inet.ip.anonportmax" to more suitable values. For example:

```
# sysctl -w net.inet.ip.anonportmin=16384
# sysctl -w net.inet.ip.anonportmax=32767
```

```
# cu -l /dev/ttyXX
```

where XX is the serial port used on your remote machine (usually /dev/tty00 or /dev/tty01). Enter the Command Monitor from the System Maintenance Menu. In order to boot the kernel properly you also have to modify the PROM environment with

```
> setenv console serial
```

Afterwards, just continue with the normal installation procedure.

Installing the NetBSD System

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

The installation methods are:

1. Installation from an existing NetBSD or IRIX system by booting the install kernel from an existing file system.
2. Installation in a diskless environment. This means that you will install your entire system onto a remotely shared disk.
3. Booting the install kernel over the network and installing onto a local disk. The beginning of the procedure is similar to diskless installation, but you will eventually end up writing the system onto your local disk.

You should familiarize yourself with the console PROM environment and the hardware configuration. A good place to study PROM tricks is the IRIX "prom" manual page. Be sure you know how to print the configuration of your machine, and how to boot from disk or network, as appropriate.

To boot from disk, use:

- **All machine types:**

```
>> setenv systempartition scsi(0)disk(X)rdisk(0)partition(8)
>> setenv osloadpartition scsi(0)disk(X)rdisk(0)partition(0)
>> setenv osloadfilename netbsd
>> setenv osloadoptions auto
>> setenv osloader boot
```

For all drives (including all SCSI-2 drives), the kernel should correctly detect the disk geometry.

If you're installing NetBSD/sgimips for the first time it's a very good idea to pre-plan partition sizes for the disks onto which you're installing NetBSD. Changing the size of partitions after you've installed is difficult. If you do not have a spare bootable disk, it may be simpler to re-install NetBSD again from scratch.

To be able to boot from local disk an SGI Volume Header partition is required. This special partition is accessed by the PROM to load the NetBSD/sgimips boot program. The partition should be about 2 MB in size, but can be reduced somewhat. The minimum size can approach 150k, but this doesn't leave room for upgrade boot programs.

Install via a bootable CD-ROM

The NetBSD/sgimips CD-ROM is not ISO-formatted. Since SGI PROMs can not boot from an ISO/CD-9660 filesystem, the CD-ROM image is generated with an SGI Volume Header for the bootstrap loader and an ISO/CD-9660-formatted partition for the distribution.

To install from a NetBSD/sgimips CD-ROM, enter the PROM command-line environment and execute one of the following commands, replacing *X* with the SCSI-ID of the CD-ROM:

(for most Indigo2 (IP22) and Indy (IP24) machines)

`>> boot -f scsi(0)cdrom(X)rdisk(0)partition(8)ip2xboot`

(for Indigo (IP20) and Indigo2/Indy machines with old PROMs)

```
>> boot -f scsi(0)cdrom(X)rdisk(0)partition(8)aoutboot
```

(for O2 (IP32) machines)

```
>> boot -f pci(0)scsi(0)cdrom(X)rdisk(0)partition(8)ip3xboot
```

Installation for diskless configurations

This section assumes you know how to configure DHCP, TFTP, and NFS services on your server.

1. Place the kernel into the TFTP server's download area. If the kernel has a ".gz" suffix, you must first uncompress it using the `gunzip` program. You must also decide which kernel you need to boot. Because of differences in SGI hardware between different machines, the kernel load addresses differ. You will need GENERIC32_IP2x for the Indy and Indigo2, GENERIC32_IP2x.ecoff for the Indigo (and Indy/Indigo2 machines with old PROMs), and GENERIC32_IP3x for the O2.

   ```
   server# gzip -d -c netbsd-GENERIC32_IP2x.gz > /tftpboot/netbsd-sgimips
   ```
   or
   ```
   server# gzip -d -c netbsd-GENERIC32_IP2x.ecoff.gz > /tftpboot/netbsd-sgimips
   ```
   or
   ```
   server# gzip -d -c netbsd-GENERIC32_IP2x.gz > /tftpboot/netbsd-sgimips
   ```

   Note that your DHCP server must be configured to specify this file as the boot file for the client.

2. Unpack the distribution sets into the client's root directory on the server.

   ```
   server# cd /export/client/mipseb
   server# gzcat .../base.tgz | tar xvpf -
   server# gzcat .../etc.tgz | tar xvpf -
   [ repeat for all sets you wish to unpack ]
   ```

3. Load the kernel from the TFTP server:

   ```
   >> setenv SystemPartition bootp():
   >> setenv netaddr your.ip.here.please
   >> boot
   ```

The installation of your diskless NetBSD/sgimips system is now complete! You may now skip ahead to any post-installation instructions.

Installation to local disk from the network

This section is a subset of the above section, and also assumes that you know how to configure DHCP and TFTP services on your server.

1. Place the install kernel into the TFTP server's download area. Just as above, you need to place the correct bootable kernel into the TFTP server's download area. However, instead of a GENERIC kernel, you must use the correct INSTALL kernel. This kernel contains a ramdisk with the installation tools and will allow you to proceed with installation without further setup.

   If the kernel has a ".gz" suffix, you must first uncompress it using the `gunzip` program. As above, you must choose either the netbsd-INSTALL32_IP2x, netbsd-INSTALL32_IP2x.ecoff, or netbsd-INSTALL32_IP3x kernel.

   Note that your DHCP server must be configured to specify this file as the boot file for the client.

2. Load the kernel from the TFTP server:

```
>> setenv SystemPartition bootp():
>> setenv netaddr your.ip.here.please
>> boot
```

Once you've booted the diskimage

Once you've booted the installation kernel you will need to select your terminal type. Use **vt100** for a serial console with a vt100-compatible terminal, or **xterm** or **xterms** for a tip(1) or cu(1) connection running in an xterm(1).

The system will then start the **sysinst** program.

Running the sysinst installation program

1. *Introduction*

   Using **sysinst**, installing NetBSD is a relatively easy process. Still, you should read this document and have it available during the installation process. This document tries to be a good guide to 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 installing NetBSD on your hard disk. **sysinst** is a menu-driven program that guides you through the installation process. 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 / DVD as the install medium.

   - What you need.
     - The distribution sets (in this example, they are on the CD or DVD).
     - A minimum of 16 MB of memory installed.
     - An optical drive.
     - A hard drive with at least 600 MB of free space for a complete base install, not including room for swap. If you wish to install the X Window System as well, you will need at least 225 MB more.
   - The Quick Installation
     - Boot the system as described above. You should be at the **sysinst** main menu.

```
.*********************************************.
* NetBSD-11.1 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                             *
* f: Config 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 the setup of your disk.
- You will be asked to choose which distribution sets to install.
- When prompted, choose **CD-ROM** as the install medium if booted from CD-ROM. The default values for the path and device should be OK.
- After the installation process has completed, you will be brought back to the main menu, where you should select **Reboot.**
- Once the system reaches the PROM prompt, you will need to modify your PROM environment settings for SystemPartition, OSLoadPartition, OSLoader, OSLoadFilename and OSLoadOptions and then boot the hard drive.
- NetBSD will now boot. If you didn't set a password for the `root` user when prompted by **`sysinst`**, logging in as `root` and setting a password should be your first task. You are also advised to read `afterboot`(8).

4. *Booting NetBSD*

   You may want to read the boot messages, to notice your disk's name and capacity. Its name will be something like `sd0` and the geometry will be printed on a line that begins with its name. As mentioned above, you may need your disk's geometry when creating NetBSD's partitions. You will also need to know the name, to tell **`sysinst`** which disk to use. The most important thing to know is that `sd0` is your first SCSI disk, `sd1` the second, etc.

   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 do not intend to use networking during the installation, but you do want your machine to be configured for networking once the system 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 when asked to provide a server.

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. You should see disk names like `sd0` or `sd1`.

7. *Selecting which sets to install*

   The next step is to choose which distribution sets you wish to install. Options are provided for full, minimal, and custom installations. If you choose sets on your own, **base**, **etc**, and a kernel must be selected.

8. *Partitioning the disk*

   - Please note that shared installs of IRIX and NetBSD on the same drive have not been tested, and as such may cause problems or may not work. It is *strongly* recommended that all data is backed before attempting such installs.

9. *Editing the NetBSD disklabel*

   The partition table of the NetBSD part of a disk is called a *disklabel*. If your disk already has a disklabel written to it, you can choose **Use existing partition sizes**. Otherwise, select **Set sizes of NetBSD partitions**.

   After you have chosen your partitions and their sizes (or if you opted to use the existing partitions), you will be presented with the layout of the NetBSD disklabel and given one more 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 entire disk.

   `d` The SGI volume header (boot partition)

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

   You will then be asked to name your disk's disklabel. The default response 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.

10. *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, select **yes**.

    The install program will now label your disk and create 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.

11. *Getting the distribution sets*

    The NetBSD distribution consists of a number of *sets* that come in the form of gzipped tar files. At this point, 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 transfer the sets to your hard disk, others will extract the sets directly.

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

12. *Installation from CD-ROM*

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

    sysinst will then check that the files are actually present in the specified location and proceed to the extraction of the sets.

13. *Installation using FTP*

    To install using FTP, you first need to configure your network setup if you haven't already done so. **sysinst** will help you with this, asking if you want to use DHCP. If you do not use DHCP, you can enter network configuration details yourself. If you do not have DNS set up for the machine that you are installing on, you can just press RETURN in answer to this question, 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, you will need to specify an IP address instead of a hostname for the FTP server.

    sysinst will then transfer the set files from the remote site to your hard disk.

14. *Installation using NFS*

    To install using NFS, you first need to configure your network setup if you haven't already done so. **sysinst** will do this for you, asking you if you want to use DHCP. If you do not use DHCP, you can enter network configuration details yourself. If you do not have DNS set up for the machine that you are installing on, you can just press RETURN in answer to this question, 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, you will need to specify an IP address instead of a hostname for the NFS server.

15. *Installation from an unmounted file system*

    In order to install from a local file system, you will need to specify the device that the file system resides on (for example `wd1e`), 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.

    If this is a CD-ROM installation, the device used will be the name for your CD-ROM player with partition letter 'a' (usually cd0a) .

16. *Installation from a local directory*

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

17. *Extracting the distribution sets*

    A progress bar will be displayed while the distribution sets are being extracted.

After all the files have been extracted, the 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.

18. *Configure additional items*

    The next menu will allow you to select a number of additional items to configure, including the time zone that you're in, to make sure your clock has the right offset from UTC, the root user's shell, and the initial root password.

    You can also enable installation of binary packages, which installs the `pkgin`(1) tool for managing binary packages for third-party software. This will feel familiar to users of package tools such as **apt-get** or **yum**. If you prefer to install third-party software from source, you can install the `pkgsrc`(7) tree.

    Finally, you can enable some daemons such as `sshd`(8), `ntpd`(8), or `mdnsd`(8).

19. *Finalizing your installation*

    Congratulations, you have successfully installed NetBSD 11.1. 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. The most important steps are described below.

1. Before all else, read `postinstall`(8).

2. Configuring `/etc/rc.conf`

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

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

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

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

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

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

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

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

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

```
ifconfig_sq0="inet 192.0.2.123 netmask 255.255.255.0"
```

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

```
ifconfig_sq0="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 networking, DHCP can be used by setting dhcpcd=YES in /etc/rc.conf.

3. Logging in

   After reboot, you can log in as root at the login prompt. If you didn't set a password in **sysinst**, there is no initial password. You should create an account for yourself (see below) and protect it and the "root" account with good passwords. By default, root login from the network is disabled (even via ssh(1)). One way to become root over the network is to log in as a different user that belongs to group "wheel" (see group(5)) and use su(1) to become root.

4. 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.

5. The X Window System

   If you installed the X Window System, you may want to read the chapter about X in the *NetBSD Guide:*
   https://www.NetBSD.org/docs/guide/en/chap-x.html:

6. Installing third-party packages

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

   - More information on the package system is available at
     https://www.NetBSD.org/docs/software/packages.html
   - A list of available packages suitable for browsing is at
     https://cdn.NetBSD.org/pub/pkgsrc/current/pkgsrc/README.html
   - Precompiled binaries can be found at
     https://cdn.NetBSD.org/pub/pkgsrc/packages/NetBSD/
     usually in the sgimips/11.1/All subdir. If you installed pkgin(1) in the **sysinst** post-installation configuration menu, you can use it to automatically install binary packages over the network. Assuming that /usr/pkg/etc/pkgin/repositories.conf is correctly configured, you can install them with the following commands:

     ```
     # pkgin install tcsh bash perl apache xfce4 firefox
     ...
     ```

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

     If it was not automatically installed, pkgin(1) can be installed on a fresh NetBSD system with pkg_add(1):

```
export PKG_PATH=https://cdn.netbsd.org/pub/pkgsrc/packages/NetBSD/$(uname -p)/$
pkg_add pkgin
```

- If you wish to use the pkgsrc(7) framework for compiling packages and did not install it from the sysinst(8) post-installation configuration menu, you can obtain it by retrieving the file
  https://cdn.NetBSD.org/pub/pkgsrc/stable/pkgsrc.tar.gz.
  It is typically extracted into /usr/pkgsrc (though other locations work fine) with the commands:

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

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

7. Misc

   - Edit /etc/mail/aliases to forward root mail to the right place. Don't forget to run newaliases(1) afterwards.
   - Edit /etc/rc.local to run any local daemons you use.
   - Many of the /etc files are documented in section 5 of the manual; so just invoking

     # **man 5** *filename*

     is likely to give you more information on these files.

Upgrading a previously-installed NetBSD System

The easiest way to upgrade to NetBSD 11.1 is with binaries, and that is the method documented here.

To do the upgrade, you must have one form of boot media available. You must also have at least the **base** and **kern** binary distribution sets available. Finally, you must have sufficient disk space available to install the new binaries. Since files already installed on the system are overwritten in place, you only need additional free space for files which weren't previously installed or to account for growth of the sets between releases.

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

The upgrade procedure is similar to an installation, but without the hard disk partitioning.

Fetching the binary sets is done in the same manner as the installation procedure; refer to the installation part of the document for help. File systems are checked before unpacking the sets.

After a new kernel has been copied to your hard disk, your machine is a complete NetBSD 11.1 system. However, that doesn't mean that you're finished with the upgrade process. You will probably want to update the set of device nodes you have in /dev. If you've changed the contents of /dev by hand, you will need to be careful about this, but if not, you can just cd into /dev, and run the command:

```
# sh MAKEDEV all
```

sysinst will attempt to merge the settings stored in your /etc directory with the new version of NetBSD using the postinstall(8) utility. However, postinstall(8) is only able to deal with changes that are easily automated. It is recommended that you use the etcupdate(8) tool to merge any remaining configuration changes.

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 11.1.

Note that **sysinst** will automatically invoke

```
postinstall fix
```

and thus all issues that are fixed by **postinstall** by default will be handled.

In NetBSD 9 and earlier, filesystems listed in `/etc/fstab` would be mounted before non-legacy **zfs** filesystems. Starting from NetBSD 10 this order has been reversed.

If you have ever run a version of NetBSD-current between April 18, 2020 and September 23, 2022 (the version numbers used in the affected time range are between 9.99.56 and 9.99.106) your FFS file systems might have broken extended attributes stored.

You should follow this *guide:*

https://wiki.netbsd.org/features/UFS2ea/

before booting the updated system multi-user for the first time.

Note that you do not need to do anything special if you never did run any affected kernel, especially if you have never run NetBSD-current.

A number of things have been removed from the NetBSD 11.1 release. See the "Components removed from NetBSD" section near the beginning of this document for a list.

Using online NetBSD documentation

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

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

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

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

```
# man passwd
```

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

```
# man 5 passwd
```

instead.

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

```
# apropos subject-word
```

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

Administrivia

If you've got something to say, do so! We'd like your input. There are various mailing lists available via the mailing list server at majordomo@NetBSD.org. See **https://www.NetBSD.org/mailinglists/** for details.

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

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

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

There are also port-specific mailing lists, to discuss aspects of each port of NetBSD. Use majordomo to find their addresses, or visit **https://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 NetBSD, and have an idea as to how you could be useful, send us mail or subscribe to: netbsd-users@NetBSD.org.

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

Thanks go to

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

  Keith Bostic
  Ralph Campbell
  Mike Karels
  Marshall Kirk McKusick

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

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

Legal Mumbo-Jumbo

This product includes software developed by Gardner Buchanan.
This product includes software developed by Garrett D'Amore.
This product includes software developed by Gary Thomas.
This product includes software developed by Gordon Ross
This product includes software developed by Harvard University.
This product includes software developed by Harvard University and its contributors.
This product includes software developed by Hellmuth Michaelis and Joerg Wunsch
This product includes software developed by Henrik Vestergaard Draboel.
This product includes software developed by Herb Peyerl.
This product includes software developed by Hidetoshi Shimokawa.
This product includes software developed by Hubert Feyrer for the NetBSD Project.
This product includes software developed by Ian W. Dall.
This product includes software developed by Intel Corporation and its contributors.
This product includes software developed by Internet Initiative Japan Inc.
This product includes software developed by Internet Research Institute, 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.
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 Julian Highfield.
This product includes software developed by K. Kobayashi
This product includes software developed by K. Kobayashi and H. Shimokawa
This product includes software developed by 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 Kyma Systems.
This product includes software developed by Leo Weppelman and Waldi Ravens.
This product includes software developed by Lloyd Parkes.
This product includes software developed by Lutz Vieweg.
This product includes software developed by Marc Horowitz.
This product includes software developed by Marcus Comstedt.
This product includes software developed by Mark Brinicombe.
This product includes software developed by Mark Brinicombe for the NetBSD Project.
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 Marshall M. Midden.
This product includes software developed by Masanobu Saitoh.
This product includes software developed by Masaru Oki.
This product includes software developed by Matt DeBergalis
This product includes software developed by Matthew Fredette.
This product includes software developed by Michael Smith.
This product includes software developed by Microsoft
This product includes software developed by Mika Kortelainen
This product includes software developed by Mike Pritchard.
This product includes software developed by Mike Pritchard and contributors.
This product includes software developed by Minoura Makoto.
This product includes software developed by MINOURA Makoto, Takuya Harakawa.
This product includes software developed by Niels Provos.
This product includes software developed by Niklas Hallqvist.
This product includes software developed by Niklas Hallqvist, Brandon Creighton and Job de Haas.
This product includes software developed by Paolo Abeni.
This product includes software developed by Paul Kranenburg.
This product includes software developed by Paul Mackerras.

This product includes software developed by Paul Mackerras <paulus@samba.org>.
This product includes software developed by Pedro Roque Marques <pedro_m@yahoo.com>
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 QUALCOMM Incorporated.
This product includes software developed by Ravikanth.
This product includes software developed by RiscBSD.
This product includes software developed by Roar Thronæs.
This product includes software developed by Rodney W. Grimes.
This product includes software developed by Roger Hardiman
This product includes software developed by Rolf Grossmann.
This product includes software developed by Ross Harvey.
This product includes software developed by Ross Harvey for the NetBSD Project.
This product includes software developed by Scott Bartram.
This product includes software developed by Scott Stevens.
This product includes software developed by Shingo WATANABE.
This product includes software developed by Softweyr LLC, the University of California, Berkeley, and its contributors.
This product includes software developed by Stephan Thesing.
This product includes software developed by Steven M. Bellovin
This product includes software developed by Takashi Hamada.
This product includes software developed by Takumi Nakamura.
This product includes software developed by Tatoku Ogaito for the NetBSD Project.
This product includes software developed by Tommi Komulainen <Tommi.Komulainen@iki.fi>.
This product includes software developed by TooLs GmbH.
This product includes software developed by Trimble Navigation, Ltd.
This product includes software developed by Waldi Ravens.
This product includes software developed by WIDE Project and its contributors.
This product includes software developed by Winning Strategies, Inc.
This product includes software developed by Yen Yen Lim and North Dakota State University
This product includes software developed by Zembu Labs, Inc.
This product includes software developed by the Alice Group.
This product includes software developed by the Computer Systems Engineering Group at Lawrence Berkeley Laboratory.
This product includes software developed by the Computer Systems Laboratory at the University of Utah.
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/)
This product includes software developed by the PocketBSD project and its contributors.
This product includes software developed by the RiscBSD kernel team
This product includes software developed by the RiscBSD team.
This product includes software developed by the SMCC Technology Development Group at Sun Microsystems, Inc.
This product includes software developed by the University of California, Lawrence Berkeley Laboratories.
This product includes software developed by the University of California, Lawrence Berkeley Laboratory.
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 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 by the Urbana-Champaign Independent Media Center.
This product includes software developed for the FreeBSD project
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 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 Eiji Kawauchi.
This product includes software developed for the NetBSD Project by Frank van der Linden
This product includes software developed for the NetBSD Project by Genetec Corporation.
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 Jonathan Stone.
This product includes software developed for the NetBSD Project by Kyma Systems LLC.
This product includes software developed for the NetBSD Project by Matthias Drochner.
This product includes software developed for the NetBSD Project by Perry E. Metzger.
This product includes software developed for the NetBSD Project by Piermont Information Systems Inc.
This product includes software developed for the NetBSD Project by Shigeyuki Fukushima.
This product includes software developed for the NetBSD Project by SUNET, Swedish University Computer Network.

The End