NAME

`INSTALL` — Installation procedure for NetBSD/next68k.

CONTENTS

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 11.99.10 on the *next68k* 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 11.99.10 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, and newer

The NetBSD 11.99.10 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.

Please note that this build is a development snapshot, not a formal release. You are testing the bleeding edge of NetBSD development, and no formal or informal testing may have been done for this snapshot (yet).

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 11.99.10 release. The complete list of changes can be found in the following files:
CHANGES:
https://nycdn.NetBSD.org/pub/NetBSD-daily/HEAD/latest/CHANGES
CHANGES.prev:
https://nycdn.NetBSD.org/pub/NetBSD-daily/HEAD/latest/CHANGES.prev

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 11.99.10 Release Contents

The root directory of the NetBSD 11.99.10 release is organized as follows:

`.../NetBSD-11.99.10/`

`CHANGES`	Changes after the netbsd-11 branch.
`CHANGES.prev`	Changes in previous NetBSD releases.
`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 11.99.10 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 11.99.10 sources which are not in **gnusrc**, **sharesrc**, or **syssrc**.
syssrc	This set contains the sources to the NetBSD 11.99.10 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/next68k subdirectory structure

The next68k-specific portion of the NetBSD 11.99.10 release is found in the `next68k` subdirectory of the distribution: `.../NetBSD-11.99.10/next68k/`. It contains the following files and directories:

`INSTALL.html`			
`INSTALL.ps`			
`INSTALL.txt`			
`INSTALL.more`			Installation notes in various file formats, including this file. The `.more` file contains underlined text using the `more`(1) conventions for indicating italic and bold display.
`binary/`			
	`kernel/`		
		`netbsd-GENERIC.gz`	A gzipped NetBSD kernel containing code for everything supported in this release.
	`sets/`		next68k binary distribution sets; see below.
`installation/`			
	`misc/`		Miscellaneous next68k installation utilities; see installation section below.

Binary distribution sets

The NetBSD next68k binary distribution sets contain the binaries which comprise the NetBSD 11.99.10 release for next68k. The binary distribution sets can be found in the `next68k/binary/sets` subdirectory of the NetBSD 11.99.10 distribution tree, and are as follows:

base	The NetBSD 11.99.10 next68k **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-GENERIC	This set contains a NetBSD/next68k 11.99.10 `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.
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 next68k 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.

Note: Each directory in the next68k binary distribution also has its own checksum files, just as the source distribution does.

NetBSD/next68k System Requirements and Supported Devices

NetBSD/next68k 11.99.10 will run on the 25 MHz 68040-based NeXT workstations. The Turbo (33 MHz) models are not supported. The 68030 model is not supported. NetBSD/next68k 11.99.10 does not have any local disk support, so you must netboot and run diskless.

The minimum configuration requires 4 MB of RAM and a network server capable of netbooting NetBSD/next68k. Serial consoles are poorly supported by the hardware, see the FAQ for help. Typically it is just easier to use the NeXT monitor and keyboard.

http://www.NetBSD.org/ports/next68k/faq.html

Supported hardware

- CPUs
  - 68040-25 2-bit grayscale (NeXTcube)
  - 68040-25 2-bit grayscale (NeXTstation)
  - 68040-25 16-bit color (NeXTstation Color)
- RS232 interfaces
  - on-board A and B serial ports (*zsc*).
- Network interfaces:
  - on-board ethernet (*xe*).
- Displays

  - on-board NeXT monochrome display (*nextdisplay*).
  - on-board NeXT 12-bit color display (*nextdisplay*).
- Input devices
  - NeXT non-adb keyboard (*nextkbd*).
- Miscellaneous
  - Battery-backed real-time clock.

Unsupported hardware

- CPUs
  - 68030-25 2-bit grayscale (NeXT Computer)
  - 68040-33 2-bit grayscale (NeXTcube Turbo)
  - 68040-33 2-bit grayscale (NeXTstation Turbo)
  - 68040-33 16-bit color (NeXTstation Color Turbo)
- Disk interfaces
  - on-board SCSI interface and disks
  - Floppy drive
  - Optical disk
  - non-SCSI CD-ROM
- Miscellaneous devices
  - NeXTdimension 32-bit color framebuffer
  - 16-bit digital audio
  - DSP
  - NeXT Printer

Getting the NetBSD System on to Useful Media

Because NetBSD/next68k must run diskless, the distribution must be installed to a remote machine that NFS exports to the NeXT machine. As a result, there is no automated install procedure. Begin by retrieving the NetBSD set files onto the serving machine. These can be found at ftp.NetBSD.org. You may want to keep the following information handy:

```
IP Address: ftp.NetBSD.org
Login: anonymous
Password: <your e-mail address>
Server path: /pub/NetBSD/NetBSD-11.99.10/next68k/binary/sets
```

Preparing your System for NetBSD installation

Get to the ROM Monitor prompt

After the system performs its power-on self test, and while it tries to boot an operating system, press the `COMMAND-`` ` `` keys (the backquote is above 7 on the keypad on the right of the keyboard). You should see a prompt like this:

```
                    NeXT ROM Monitor 2.5 (v66)

CPU MC68040 25 MHz, memory 100 nS
Backplane slot #0
Ethernet address:  0:0:f:0:fb:90
Memory size 40 MB
NeXT>
```

Write down the ethernet address. You will need this later to configure the netboot server for your NetBSD/next68k system.

Change the configuration parameters to netboot

Press the `p` key to modify the configuration parameters. You will need to change the `boot command` to `en()netbsd` which is to boot from the `en()` ethernet device the kernel named `netbsd`. You should write down what the original `boot command` was in case you wish to boot NEXTSTEP in the future. It will probably be `sd()`. While modifying the parameters, you may wish to set the system to use `verbose test mode` which will print detailed messages while it is attempting to load the kernel, as opposed to the normal graphic of a network cable. You should not set the `extended diagnostics` as the system will try to boot a diagnostic kernel. Here is an example configuration session:

```
NeXT> p
boot command: sd()? en()netbsd
DRAM tests: yes? yes
perform power-on system test: yes? yes
    sound out tests: yes? yes
    SCSI tests: yes? yes
    loop until keypress: no? no
    verbose test mode: no? yes
boot extended diagnostics: no? no
serial port A is alternate console: no? yes
allow any ROM command even if password protected: no? no
allow boot from any device even if password protected: no? no
allow optical drive #0 eject even if password protected: yes? yes
enable parity checking if parity memory is present: no? no
```

Now, whenever you reset or power on your system, it will attempt to netboot. From the ROM Monitor prompt, you can simply type `b` to netboot. You can also press the power key (above the arrows) to turn off the system.

Installing the NetBSD System

1. Introduction

   To netboot a next68k, you must configure one or more servers to provide information and files to your next68k (the 'client'). If you are using NetBSD (any architecture) on your netboot server(s), the information provided here should be sufficient to configure everything. Additionally, you may wish to look at the `diskless`(8) manual page and the manual pages for each daemon you'll be configuring. If the server(s) are another operating system, you should consult the NetBSD Diskless HOW-TO, which will walk you through the steps necessary to configure the netboot services on a variety of platforms:

   https://www.NetBSD.org/docs/network/netboot/

   Briefly, the netboot process involves discovery, bootstrap, kernel and file system stages. In the first stage, the client discovers information about where to find the bootstrap program. Next, it downloads and executes the bootstrap program. The bootstrap program goes through another discovery phase to determine where the kernel is located. The bootstrap program tries to mount the NFS share containing the kernel. Once the kernel is loaded, it starts executing. The kernel tries to mount the NFS share that had the kernel and starts executing `init`(8). All next68k systems use BOOTP for the discovery stage. TFTP is used in the bootstrap phase to download the bootstrap program, `boot`. NFS is used in both the kernel and file system stages to download the kernel, and to access files on the file server.

   We will use '`CC:CC:CC:CC:CC:CC`' as the MAC address (ethernet hardware address) of your netboot client machine. You should have determined this address in an earlier stage. In this example, we will use '`192.168.1.10`' as the IP address of your client and '`client.test.net`' as its name.

We will assume you're providing all of your netboot services on one machine called '`server.test.net`' with the client's files exported from the directory `/export/client/root`. You should, of course, replace all of these with the names, addresses, and paths appropriate to your environment.

You should set up each netboot stage in order (i.e., discovery, bootstrap, kernel, and then file system) so that you can test them as you proceed.

2. `dhcpd`(8) in `bootpd`(8) compatible mode

   Put the following lines in your `/etc/dhcpd.conf` (see `dhcpd.conf`(5) and `dhcp-options`(5) for more information):

```
ddns-update-style none;
                # Do not use any dynamic DNS features
                #
allow bootp;    # Allow bootp requests, thus the dhcp server
                # will act as a bootp server.
                #
authoritative;  # master DHCP server for this subnet
                #
subnet 192.168.1.0 netmask 255.255.255.0 {
                # Which network interface to listen on.
                # The zeros indicate the range of addresses
                # that are allowed to connect.
}
group {
                # Set of parameters common to all clients
                # in this "group".
                #
        option broadcast-address        192.168.1.255;
        option domain-name              "test.net";
        option domain-name-servers      dns.test.net;
        option routers                  router.test.net;
        option subnet-mask              255.255.255.0;
                #
                # An individual client.
                #
        host client.test.net {
                hardware ethernet       CC:CC:CC:CC:CC:CC;
                fixed-address           192.168.1.10;
                #
                # Name of the host (if the fixed address
                # doesn't resolve to a simple name).
                #
                option host-name        "client";
                #
                # Name of the bootloader or kernel
                # to download via tftp.
#
                filename                "boot";

                #
                # The path on the NFS server.
```

```
                        #
                        option root-path "/export/client/root";

                        #
                        #next68k machines require non-RFC1048 BOOTP
                        #
                        always-reply-rfc1048 false;

                        #
                        # The host address of the NFS server. This is mandatory for
                        # NetBSD kernels even it's the same host as the DHCP server.
                        #
                        next-server server.test.net;
                }
        #you may paste another "host" entry here for additional
        #clients on this network
       }
```

You will need to make sure that the `dhcpd.leases` file exists.

```
# touch /var/db/dhcpd.leases
```

You will need to start the **dhcpd**. If it's already running, you will need to restart it to force it to re-read its configuration file. If the server is running NetBSD, you can achieve this with:

```
# /etc/rc.d/dhcpd restart
```

3. `tftpd`(8)

   The default configuration of the TFTP server is to run in a `chroot`(8) environment in the `/tftpboot` directory. Thus, the first order of business is to create this directory:

```
# mkdir -p /tftpboot
```

   Next, edit `/etc/inetd.conf` and uncomment the line with the TFTP daemon:

```
tftp  dgram  udp  wait  root  /usr/libexec/tftpd tftpd -l -s /tftpboot
```

   Now, restart `inetd`(8). If the server is running NetBSD, you can achieve this with:

```
# /etc/rc.d/inetd restart
```

   Now, you need to copy the bootloader for your next68k machine to `/tftpboot`. Get `boot` from the `installation` directory of the distribution.

```
# cp boot /tftpboot
```

   Just to be sure, let's make everything readable.

```
# chmod -R a+rX /tftpboot
```

   Sometimes, the `arp`(8) table gets messed up, and the TFTP server can't communicate with the client. In this case, it will write a log message (via `syslogd`(8)) to `/var/log/messages` saying: '`tftpd: write: Host is down`'. If this is the case, you may need to force the server to map your client's ethernet address to its IP address:

```
# arp -s client CC:CC:CC:CC:CC:CC
```

4. `nfsd`(8), `mountd`(8), and `rpcbind`(8)

Now your system should be able to load the bootstrap program and start looking for the kernel. Let's set up the NFS server. Create the directory you are exporting for the netboot client:

```
# mkdir -p /export/client/root
```

Put the following line in `/etc/exports` to enable NFS sharing:

```
/export/client/root -maproot=root client.test.net
```

If your server is currently running an NFS server, you only need to restart `mountd`(8). Otherwise, you need to start `rpcbind`(8) and `nfsd`(8). If the server is running NetBSD, you can achieve this with:

```
# /etc/rc.d/rpcbind start
# /etc/rc.d/nfsd start
# /etc/rc.d/mountd restart
```

5. NetBSD kernel

   Now, if you place a kernel named `netbsd` in `/export/client/root` your client should boot the kernel. Use `binary/kernel/netbsd-GENERIC.gz`.

   ```
   # gunzip netbsd-GENERIC.gz
   # mv netbsd-GENERIC /export/client/root/netbsd
   ```

6. Client file system

   You need to extract and set up the client's installation of NetBSD. The Diskless HOW-TO describes how to provide better security and save space on the NFS server over the procedure listed here. See *for details:*

   https://www.NetBSD.org/docs/network/netboot/nfs.html.

- Extracting distribution sets

  ```
  # cd /export/client/root
  # tar -xpzf /path/to/files/base.tgz
  # tar -xpzf /path/to/files/etc.tgz
  ```

  Continue with the other non-essential distribution sets if desired.

- Set up swap

  ```
  # mkdir /export/client/root/swap
  # dd if=/dev/zero of=/export/client/swap bs=4k count=4k
  # echo '/export/client/swap -maproot=root:wheel client.test.net' >> /etc/exports
  # /etc/rc.d/mountd restart
  ```

  This creates a 16 MB swap file and exports it to the client.

- Create device nodes

  ```
  # cd /export/client/root/dev
  # ./MAKEDEV all
  ```

  This procedure only works on NetBSD hosts.

- Set up the client's `fstab`(5)

  Create a file in `/export/client/root/etc/fstab` with the following lines:

  ```
  server:/export/client/swap   none  swap  sw,nfsmntpt=/swap
  server:/export/client/root   /     nfs   rw 0 0
  ```

- Set up the client's `rc.conf`(5)

  Edit `/export/client/root/etc/rc.conf`

  ```
  rc_configured=YES
  hostname="client"
  defaultroute="192.168.1.1"
  nfs_client=YES
  auto_ifconfig=NO
  net_interfaces=""
  ```

  Make sure rc does not reconfigure the network device since it will lose its connection to the NFS server with your root file system.

- Set up the client's `hosts`(5) file.

  Edit `/export/client/root/etc/hosts`

  ```
  ::1                     localhost
  127.0.0.1               localhost
  192.168.1.10 client.test.net client
  192.168.1.5  server.test.net server
  ```

7. Setting up the server daemons

   You need these services to start up every time you boot your server, make sure the following lines are present in your `/etc/rc.conf`:

   ```
   dhcpd=YES        dhcpd_flags="-q"
   nfs_server=YES         # enable server daemons
   mountd=YES
   rpcbind=YES      rpcbind_flags="-l"   # -l logs libwrap
   ```

   Also, you'll need to make sure the `tftpd` line in `/etc/inetd.conf` remains uncommented.

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_xe0="inet 192.0.2.123 netmask 255.255.255.0"
```

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

```
ifconfig_xe0="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 `next68k/11.99.10/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/$(uname -p)/$(uname -r |
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

NetBSD/next68k does not currently support an upgrade procedure.

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 11.99.10.

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 11.99.10 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

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:

NetBSD is a registered trademark of The NetBSD Foundation, Inc.
This product includes software developed by the University of California, Berkeley and its contributors.
This product includes software developed by the NetBSD Foundation.
This product includes software developed by The NetBSD Foundation, Inc. and its contributors.
This product includes software developed for the NetBSD Project. See https://www.NetBSD.org/ for information about NetBSD.
This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)
This product includes cryptographic software written by Eric Young (eay@mincom.oz.au)
This product includes software designed by William Allen Simpson.
This product includes software developed at Ludd, University of Luleå.
This product includes software developed at Ludd, University of Luleå, Sweden and its contributors.
This product includes software developed at the Information Technology Division, US Naval Research Laboratory.
This product includes software developed by Aaron Brown and Harvard University.
This product includes software developed by Adam Ciarcinski for the NetBSD project.
This product includes software developed by Adam Glass.
This product includes software developed by Adam Glass and Charles M. Hannum.
This product includes software developed by Alex Zepeda.
This product includes software developed by Alex Zepeda, and Colin Wood for the NetBSD Project.
This product includes software developed by Allen Briggs.

This product includes software developed by Amancio Hasty and Roger Hardiman
This product includes software developed by Berkeley Software Design, Inc.
This product includes software developed by Bill Paul.
This product includes software developed by Bodo Möller.
This product includes software developed by Brini.
This product includes software developed by Bruce M. Simpson.
This product includes software developed by Causality Limited.
This product includes software developed by Charles Hannum.
This product includes software developed by Charles M. Hannum.
This product includes software developed by Charles M. Hannum, by the University of Vermont and State Agricultural College and Garrett A. Wollman, by William F. Jolitz, and by the University of California, Berkeley, Lawrence Berkeley Laboratory, and its contributors.
This product includes software developed by Christian E. Hopps.
This product includes software developed by Christian E. Hopps, Ezra Story, Kari Mettinen, Markus Wild, Lutz Vieweg and Michael Teske.
This product includes software developed by Christopher G. Demetriou.
This product includes software developed by Christopher G. Demetriou for the NetBSD Project.
This product includes software developed by Cisco Systems, Inc.
This product includes software developed by Colin Wood.
This product includes software developed by Colin Wood for the NetBSD Project.
This product includes software developed by Computing Services at Carnegie Mellon University (http://www.cmu.edu/computing/).
This product includes software developed by Daishi Kato
This product includes software developed by Daniel Widenfalk and Michael L. Hitch.
This product includes software developed by Daniel Widenfalk for the NetBSD Project.
This product includes software developed by David Jones and Gordon Ross
This product includes software developed by David Miller.
This product includes software developed by Dean Huxley.
This product includes software developed by Emmanuel Dreyfus
This product includes software developed by Eric S. Hvozda.
This product includes software developed by Eric S. Raymond
This product includes software developed by Eric Young (eay@cryptsoft.com)
This product includes software developed by Eric Young (eay@mincom.oz.au)
This product includes software developed by Ezra Story.
This product includes software developed by Ezra Story and by Kari Mettinen.
This product includes software developed by Ezra Story, by Kari Mettinen and by Bernd Ernesti.
This product includes software developed by Ezra Story, by Kari Mettinen, and Michael Teske.
This product includes software developed by Ezra Story, by Kari Mettinen, Michael Teske and by Bernd Ernesti.
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 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 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 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, 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 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 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 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 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.
This product includes software developed for the NetBSD Project by Wasabi Systems, Inc.
This product includes software developed or owned by Caldera International, Inc.
This product includes software developed under OpenBSD by Per Fogelstrom.
This product includes software developed under OpenBSD by Per Fogelstrom Opsycon AB for RTMX Inc, North Carolina, USA.
This software was developed by Holger Veit and Brian Moore for use with "386BSD" and similar operating systems. "Similar operating systems" includes mainly non-profit oriented systems for research and education, including but not restricted to "NetBSD", "FreeBSD", "Mach" (by CMU).
The Institute of Electrical and Electronics Engineers and The Open Group, have given us permission to reprint portions of their documentation.

In the following statement, the phrase "this text" refers to portions of the system documentation.

Portions of this text are reprinted and reproduced in electronic form in NetBSD, from IEEE Std 1003.1, 2004 Edition, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 6, Copyright (C) 2001-2004 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between these versions and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document.

The original Standard can be obtained online at http://www.opengroup.org/unix/online.html.

This notice shall appear on any product containing this material.

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

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

Some files have the following copyright:

> Mach Operating System
> Copyright (c) 1991,1990,1989 Carnegie Mellon University
> All Rights Reserved.
>
> Permission to use, copy, modify and distribute this software and its documentation is hereby granted, provided that both the copyright notice and this permission notice appear in all copies of the software, derivative works or modified versions, and any portions thereof, and that both notices appear in supporting documentation.
>
> CARNEGIE MELLON ALLOWS FREE USE OF THIS SOFTWARE IN ITS CONDITION. CARNEGIE MELLON DISCLAIMS ANY LIABILITY OF ANY KIND FOR ANY DAMAGES WHATSOEVER RESULTING FROM THE USE OF THIS SOFTWARE.
>
> Carnegie Mellon requests users of this software to return to
> Software Distribution Coordinator or Software.Distribution@CS.CMU.EDU
> School of Computer Science
> Carnegie Mellon University
> Pittsburgh PA 15213-3890

any improvements or extensions that they make and grant Carnegie the rights to redistribute these changes.

Some files have the following copyright:
Copyright (c) 1994, 1995 Carnegie-Mellon University.
All rights reserved.

Author: Chris G. Demetriou

Permission to use, copy, modify and distribute this software and its documentation is hereby granted, provided that both the copyright notice and this permission notice appear in all copies of the software, derivative works or modified versions, and any portions thereof, and that both notices appear in supporting documentation.
CARNEGIE MELLON ALLOWS FREE USE OF THIS SOFTWARE IN ITS "AS IS" CONDITION. CARNEGIE MELLON DISCLAIMS ANY LIABILITY OF ANY KIND FOR ANY DAMAGES WHATSOEVER RESULTING FROM THE USE OF THIS SOFTWARE.

Carnegie Mellon requests users of this software to return to
Software Distribution Coordinator or Software.Distribution@CS.CMU.EDU
School of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213-3890

any improvements or extensions that they make and grant Carnegie the rights to redistribute these changes.

Some files have the following copyright:
Copyright 1996 The Board of Trustees of The Leland Stanford Junior University. All Rights Reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies. Stanford University makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

The End