NAME

INSTALL — Installation procedure for NetBSD/evbarm.

CONTENTS

About this Document 2
What is NetBSD? 2
Dedication 2
Changes Between The NetBSD 8 and 9 Releases 3
  Installation and Partitioning Changes 3
Features to be removed in a later release 3
The NetBSD Foundation 3
Sources of NetBSD 3
NetBSD 9.0 Release Contents 3
  NetBSD/evbarm subdirectory structure 4
  Binary distribution sets 5
NetBSD/evbarm System Requirements and Supported Devices 6
  Supported devices (Technologic Systems TS-7200) 6
    TS-5620 Real Time Clock 6
    Serial ports 7
    Ethernet adapters 7
    Other on-board functions 7
  Supported devices (ARM, Ltd. Integrator) 7
    PrimeCell PL030 Real Time Clock 7
    Serial ports 7
    SCSI host adapters 7
    Ethernet adapters 7
  Supported devices (Intel IQ80310) 7
    Serial ports 7
    Ethernet adapters 7
    i80312 Companion I/O functions 7
  Supported devices (Intel IQ80321) 7
    Serial ports 7
    Ethernet adapters 8
    i80321 I/O Processor functions 8
  Supported devices (Team ASA Npwr) 8
    Serial ports 8
    SCSI host adapters 8
    Ethernet adapters 8
  Supported devices (Intel IXM1200) 8
    Serial ports 8
    Ethernet adapters 8
  Supported devices (Samsung SMDK2800) 8
    Serial ports 8
  Supported devices (ADI BRH) 8
    Serial ports 8
    Ethernet adapters 8
  Supported devices (Arcom Viper) 9
    Serial ports 9
    Ethernet adapters 9
Getting the NetBSD System on to Useful Media 9
Preparing your System for NetBSD installation 10
  Preparation for the Technologic Systems TS-7200 10

Preparation for the Intel IQ80310 11
Preparation for the Intel IQ80321 13
Installing the NetBSD System 15
Installation for the Technologic Systems TS-7200 15
Installation for the Intel IQ80310 16
Installation for the Intel IQ80321 17
Post installation steps 19
Upgrading a previously-installed NetBSD System 21
Compatibility Issues With Previous NetBSD Releases 21
Important note regarding ABI change on ARM ports 21
Using online NetBSD documentation 22
Administrivia 22
Thanks go to 23
Legal Mumbo-Jumbo 23
The End 27

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 9.0 on the *evbarm* platform. It is available in four different formats titled `INSTALL.`*`ext`*, where *`.ext`* is one of `.ps`, `.html`, `.more`, or `.txt`:

`.ps` PostScript.

`.html` Standard Internet HTML.

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

`.txt` Plain old ASCII.

You are reading the *PostScript* version.

What is NetBSD?

The NetBSD Operating System is a fully functional Open Source UNIX-like operating system derived from the University of California, Berkeley Networking Release 2 (Net/2), 4.4BSD-Lite, and 4.4BSD-Lite2 sources. NetBSD runs on many different different system architectures (ports) across a variety of distinct CPU families, and is being ported to more. The NetBSD 9.0 release contains complete binary releases for most of these system architectures, with preliminary support for the others included in source form. Please see the NetBSD *website:* **https://www.NetBSD.org/** for information on them.

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.

Dedication

NetBSD 9.0 is dedicated to the memory of Matthias Drochner, who passed away in August 2018 and Eric Schnoebelen, who passed away in March 2019.

Matthias' technical contributions are too many to list here in full. He was a long term contributor and commited more than 3000 changes all over the NetBSD source tree and lately was especially active in keeping some of our most weired ancient VME architectures in shape.

Eric was a long term pkgsrc developer and well known community member.

Beyond their technical contributions, Eric and Matthias were always helpful and friendly. Their example encouraged users to contribute to the project and share their work with the community.

Changes Between The NetBSD 8 and 9 Releases

The NetBSD 9.0 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.

It is impossible to completely summarize the massive development that went into the NetBSD 9.0 release. The complete list of changes can be found in the following files:
CHANGES:
https://cdn.NetBSD.org/pub/NetBSD/NetBSD-9.0/CHANGES
CHANGES-9.0:
https://cdn.NetBSD.org/pub/NetBSD/NetBSD-9.0/CHANGES-9.0
files in the top level directory of the NetBSD 9.0 release tree.

Installation and Partitioning Changes

The sysinst installation program has been reworked for this release.

It now supports arbitrary big disks and offers GPT partitions as alternative to MBR/fdisk partitions on a lot architectures.

Unfortunately it has not been tested on all hardware supported by NetBSD. If you have problems partitioning the target disk or installing the system, please report bugs with as much details as possible. See the Administrivia section below on how to report bugs or contact other users and ask for support.

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`.

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 9.0 Release Contents

The root directory of the NetBSD 9.0 release is organized as follows:

`.../NetBSD-9.0/`

`CHANGES`	Changes between the 8.0 and 9.0 releases.
`CHANGES-9.0`	Changes between the initial 9.0 branch and final release of 9.0.
`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 9.0 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 9.0 sources which are not in **gnusrc**, **sharesrc**, or **syssrc**.

syssrc This set contains the sources to the NetBSD 9.0 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. Except for the **pkgsrc** set, which is traditionally unpacked into `/usr/pkgsrc`, all sets may be unpacked into `/usr/src` with the command:

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

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

`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/evbarm subdirectory structure

The evbarm-specific portion of the NetBSD 9.0 release is found in the `evbarm` subdirectory of the distribution: `.../NetBSD-9.0/evbarm/`. 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 sup-
                                ported in this release.
            sets/   evbarm binary distribution sets; see below.
installation/
            misc/       Miscellaneous evbarm installation utilities; see installation section below.
```

Binary distribution sets

The NetBSD evbarm binary distribution sets contain the binaries which comprise the NetBSD 9.0 release for evbarm. The binary distribution sets can be found in the `evbarm/binary/sets` subdirectory of the NetBSD 9.0 distribution tree, and are as follows:

base The NetBSD 9.0 evbarm **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 developement. 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-INTEGRATOR
(see below)

kern-IQ80310
(see below)

kern-IQ80321
(see below)

kern-TEAMASA_NPWR

These sets contain a NetBSD/evbarm 9.0 kernel, named `/netbsd`. Some of these sets also contain a binary format kernel named `/netbsd.bin` and an S-record format kernel named `/netbsd.srec` that your firmware may need to boot. You *must* install the kernel that matches your hardware.

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 recover 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 developement. 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 evbarm 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 evbarm binary distribution also has its own checksum files, just as the source distribution does.

NetBSD/evbarm System Requirements and Supported Devices

NetBSD 9.0 runs on the following ARM architecture evaluation boards:

- Technologic Systems TS-7200
- ARM, Ltd. Integrator
- Intel IQ80310
- Intel IQ80321
- Team ASA, Inc. Npwr
- Intel IXM1200
- Samsung SMDK2800
- ADI Engineering, Inc. BRH ("Big Red Head")
- Arcom Viper

Supported devices (Technologic Systems TS-7200)

For devices that have machine-independent drivers, this is not an exhaustive list, but merely reflects the devices that have been tested on this platform.

- TS-5620 Real Time Clock

- Serial ports
  - On-chip EP9302 UARTs
  - TS-SER1, TS-SER2, TS-SER4 PC/104 serial adapters
  - TS-MODEM, TS-MODEM2 PC/104 land-line/cellular modems
- Ethernet adapters
  - On-chip EP9302 10/100 ethernet
  - TS-ETH10 PC/104 expansion ethernet adapters
- Other on-board functions
  - CompactFlash socket
  - USB OHCI host
  - Text-mode HD44780 LCD display header
  - Matrix keypad/GPIO header
  - Board temperature sensor and watchdog timer

Supported devices (ARM, Ltd. Integrator)

For devices that have machine-independent drivers, this is not an exhaustive list, but merely reflects the devices that have been tested on this platform.

- PrimeCell PL030 Real Time Clock
- Serial ports
  - PrimeCell PL010 UART
- SCSI host adapters
  - Adaptec AHA-2910, 2915, 2920, and 2930C adapters.
  - Adaptec AHA-2x4x[U][2][W] cards and onboard PCI designs using the AIC-7770, AIC-7850, AIC-7860, AIC-7870, AIC-7880 and AIC-789x chipsets.
  - Adaptec AHA-394x[U][W] cards [b]
  - Adaptec AHA-3950U2 cards
  - Adaptec AHA-3960, 19160 and 29160 Ultra-160 adapters
- Ethernet adapters
  - Intel i82557, i82558, i82559, i82550

Supported devices (Intel IQ80310)

For devices that have machine-independent drivers, this is not an exhaustive list, but merely reflects the devices that have been tested on this platform.

- Serial ports
  - On-board ST16c550 UARTs
- Ethernet adapters
  - Intel i82557, i82558, i82559, i82550
- i80312 Companion I/O functions
  - General Purpose I/O unit

Supported devices (Intel IQ80321)

For devices that have machine-independent drivers, this is not an exhaustive list, but merely reflects the devices that have been tested on this platform.

- Serial ports

  - On-board TI 16550 UART

- Ethernet adapters
  - Intel i82542, i82543, i85244 Gigabit Ethernet

- i80321 I/O Processor functions
  - Application Accelerator Unit
  - Watchdog timer

Supported devices (Team ASA Npwr)

The Npwr is a storage appliance based on the Intel IOP310 I/O processor chipset. This is an exhaustive list of devices supported on the Npwr as the Npwr has no expansion capability.

- Serial ports
  - On-board TI 16550 UART

- SCSI host adapters
  - On-board LSI Logic 53c1010 Ultra160 SCSI adapter

- Ethernet adapters
  - On-board Intel i82544 Gigabit Ethernet

Supported devices (Intel IXM1200)

For devices that have machine-independent drivers, this is not an exhaustive list, but merely reflects the devices that have been tested on this platform.

- Serial ports
  - IXP1200 on-chip serial port

- Ethernet adapters
  - Intel i82557, i82558, i82559, i82550

Supported devices (Samsung SMDK2800)

For devices that have machine-independent drivers, this is not an exhaustive list, but merely reflects the devices that have been tested on this platform.

- Serial ports
  - S3C2800 on-chip serial ports

Supported devices (ADI BRH)

For devices that have machine-independent drivers, this is not an exhaustive list, but merely reflects the devices that have been tested on this platform.

- Serial ports
  - On-board TI 16752 DUART

- Ethernet adapters
  - Intel i82557, i82558, i82559, i82550
  - Intel i82544, i82546 Gigabit Ethernet

Supported devices (Arcom Viper)

Current this is an exhaustive list, since expansion ports such as PC/104 and USB are not yet supported.

- Serial ports
  - On-chip PXA255 UARTs

  Ethernet adapters
  - On-board SMC91C111

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 `evbarm/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-9.0
# cd evbarm/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 on the target system that you wish to keep. Mistakes in partitioning may lead to data loss.

Preparation for the Technologic Systems TS-7200

The TS-7200 uses RedBoot firmware which supports loading the kernel via serial port and the on-chip Ethernet. These notes assume that the Ethernet will be used to load the kernel. RedBoot can either use TFTP or HTTP to load the kernel via the network.

These notes also assume the TS-7200 will be run from the CompactFlash; that a FFS filesystem on the CompactFlash card will provide the root file system. The minimum size of the CompactFlash card is 64MB with 512MB being the recommended minimum size if attempting a full installation.

The first step to install NetBSD is to interrupt normal system bootup and drop to the RedBoot prompt by hitting Ctrl-C immediately after applying power. To load both the installation kernel and the generic post-installation TS7200 kernel you will need network connectivity from RedBoot which you will have to configure using the RedBoot **fconfig** command; see the RedBoot documentation for more information.

A compressed kernel image and loader suitable for placing into flash is provided with the distribution. Use of this image is optional; a kernel that can be directly loaded over the network by RedBoot is also provided.

The image for the TS-7200 can be found under the NetBSD/evbarm 9.0 distribution directory in `evbarm/binary/gzimg/gzimg_TS7200_flash_0x60660000.gz`, and corresponds to the kernel in `evbarm/binary/kernel/netbsd-TS7200.gz`. The following steps describe how to copy the compressed kernel image into flash.

1. Uncompress the gzimg file and place it into the TFTP server's download area (note, the kernel inside the gzimg is still compressed after this step).

   ```
   server# gzcat gzimg_TS7200_flash_0x60660000.gz > /tftpboot/gzimg_TS7200_flash_0x60660000
   ```

2. Load the gzimg from the TFTP server.

   ```
   RedBoot> load -r -b 0x00200000 gzimg_TS7200_flash_0x60660000
   Raw file loaded 0x00200000-0x00605fe0
   ```

3. Use the RedBoot **fis** command to copy the image into flash. For more information on the RedBoot Flash Image System, see the RedBoot documentation. The factory default Linux image may already be present in flash and must be deleted before continuing.

   ```
   RedBoot> fis delete vmlinux
   RedBoot> fis create -b 0x00200000 -f 0x60660000 -l 0x00160000 netbsd.gz
   ```

Once the compressed kernel image has been copied into flash, it may be started by jumping to the flash address of the image:

```
RedBoot> g 0x60660000
```

Preparation for the Intel IQ80310

The IQ80310 uses RedBoot firmware from Red Hat, Inc. RedBoot supports loading the kernel via the serial port and the on-board Ethernet. These notes assume that the Ethernet will be used to load the kernel. RedBoot uses the Trivial File Transfer Protocol (TFTP) to load the kernel via the network.

These notes also assume that the IQ80310 will be run as a diskless system; that an NFS server will provide the root file system. The NetBSD kernel uses the Dynamic Host Configuration Protocol (DHCP) to obtain network address and root file system information.

The first step is to configure your DHCP, NFS, and TFTP server software. You will need the MAC address for the on-board Ethernet, which RedBoot can provide; see the RedBoot documentation for more information.

The next step is to configure RedBoot to obtain its IP address from your DHCP server; see the RedBoot documentation for more information.

Once your DHCP server and RedBoot are properly configured, you should see information similar to the following when the IQ80310 is reset or powered-up:

```
RedBoot(tm) debug environment - built 17:16:14, Feb 12 2001
Platform: IQ80310 (XScale)
Copyright (C) 2000, Red Hat, Inc.

RAM: 0xa0000000-0xa2000000
FLASH: 0x00000000 - 0x00800000, 64 blocks of 0x00020000 bytes each.
IP: 192.168.0.10, Default server: 192.168.0.1
RedBoot>
```

A compressed kernel image and loader suitable for placing into flash is provided with the distribution. Use of this image is optional; a kernel that can be directly loaded over the network by RedBoot is also provided.

The image for the IQ80310 can be found under the NetBSD/evbarm 9.0 distribution directory in `evbarm/binary/gzimg/gzimg_IQ80310_flash_0x00080000.gz`, and corresponds to the kernel in `evbarm/binary/kernel/netbsd-IQ80310.gz`. The following steps describe how to copy the compressed kernel image into flash.

1. Uncompress the gzimg file and place it into the TFTP server's download area (note, the kernel inside the gzimg is still compressed after this step).

   ```
   server# gzcat gzimg_IQ80310_flash_0x00080000.gz > /tftpboot/gzimg_IQ80310_flash_0x00080000
   ```

2. Load the gzimg from the TFTP server.

   ```
   RedBoot> load -r -b 0xa0200000 gzimg_IQ80310_flash_0x00080000
   Raw file loaded 0xa0200000-0xa0305fe0
   ```

3. Use the RedBoot **fis** command to copy the image into flash. For more information on the RedBoot Flash Image System, see the RedBoot documentation.

   ```
   RedBoot> fis create -b 0xa0200000 -f 0x00080000 -l 0x00200000 netbsd.gz
   ... Erase from 0x00080000-0x00280000: ....................
   ... Program from 0xa0200000-0xa0400000 at 0x00080000: .....................
   ... Unlock from 0x007e0000-0x00800000:.
   ... Erase from 0x007e0000-0x00800000:.
   ... Program from 0xa1fd0000-0xa1ff0000 at 0x007e0000:.
   ... Lock from 0x007e0000-0x00800000:.
   ```

Once the compressed kernel image has been copied into flash, it may be started by jumping to the flash address of the image:

```
RedBoot> g 0x00080000

>> NetBSD/IQ80310 Gzip Boot, Revision 1.1
>> (root@tgm, Thu Mar 28 18:32:45 PST 2002)
>> RAM 0xa0000000 - 0xaffffff, heap at 0xaffd0000
>> Load address: 0xa0200000
>> Image size: 741244
Uncompressing image...done.
Jumping to image @ 0xa0200000...

NetBSD/evbarm (IQ80310) booting ...
Resetting secondary PCI bus...
initarm: Configuring system ...
physmemory: 65536 pages at 0xa0000000 -> 0xaffffff
init subsystems: stacks vectors undefined pmap irq done.
[ using 155084 bytes of netbsd ELF symbol table ]
Copyright (c) 1996, 1997, 1998, 1999, 2000, 2001, 2002
    The NetBSD Foundation, Inc.  All rights reserved.
Copyright (c) 1982, 1986, 1989, 1991, 1993
    The Regents of the University of California.  All rights reserved.

NetBSD 1.5ZC (IQ80310) #20: Fri Mar 29 10:25:53 PST 2002
    root@tgm:/u1/netbsd/src/sys/arch/evbarm/compile/IQ80310
```

```
total memory = 256 MB
avail memory = 232 MB
using 1228 buffers containing 13208 KB of memory
mainbus0 (root)
cpu0 at mainbus0: i80200 step A-0 (XScale core)
cpu0: DC enabled IC enabled WB enabled LABT
cpu0: 32KB/32B 32-way Instruction cache
cpu0: 32KB/32B 32-way write-back-locking Data cache
obio0 at mainbus0: board rev. F, CPLD rev. D, backplane present
com0 at obio0 addr 0xfe810000: ns16550a, working fifo
com0: console
com1 at obio0 addr 0xfe800000: ns16550a, working fifo
iopxs0 at mainbus0: i80312 Companion I/O, acting as PCI host
iopxs0: configuring Secondary PCI bus
pci0 at iopxs0 bus 1
pci0: i/o space, memory space enabled
ppb0 at pci0 dev 7 function 0: Digital Equipment DECchip 21154 PCI-PCI Bridge (rev. 0
pci1 at ppb0 bus 2
pci1: i/o space, memory space enabled
fxp0 at pci1 dev 0 function 0: i82559S Ethernet, rev 9
fxp0: interrupting at iq80310 irq 1
fxp0: Ethernet address 00:80:4d:46:0b:b9
inphy0 at fxp0 phy 1: i82555 10/100 media interface, rev. 4
inphy0: 10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
clock: hz=100 stathz=0 profhz=0
boot device: <unknown>
root device:
```

Preparation for the Intel IQ80321

The IQ80321 uses RedBoot firmware from Red Hat, Inc. RedBoot supports loading the kernel via the serial port and the on-board Ethernet. These notes assume that the Ethernet will be used to load the kernel. RedBoot uses the Trivial File Transfer Protocol (TFTP) to load the kernel via the network.

These notes also assume that the IQ80321 will be run as a diskless system; that an NFS server will provide the root file system. The NetBSD kernel uses the Dynamic Host Configuration Protocol (DHCP) to obtain network address and root file system information.

The first step is to configure your DHCP, NFS, and TFTP server software. You will need the MAC address for the on-board Ethernet, which RedBoot can provide; see the RedBoot documentation for more information.

The next step is to configure RedBoot to obtain its IP address from your DHCP server; see the RedBoot documentation for more information.

Once your DHCP server and RedBoot are properly configured, you should see information similar to the following when the IQ80321 is reset or powered-up:

```
Ethernet eth0: MAC address 00:07:e9:03:38:40
IP: 192.168.0.11, Default server: 192.168.0.1

RedBoot(tm) bootstrap and debug environment [ROM]
Non-certified release, version UNKNOWN - built 11:21:56, Feb  1 2002

Platform: IQ80321 (XScale)
```

```
Copyright (C) 2000, 2001, 2002, Red Hat, Inc.

RAM: 0x00000000-0x08000000, 0x00017008-0x01ddd000 available
FLASH: 0xf0000000 - 0xf0800000, 64 blocks of 0x00020000 bytes each.
RedBoot>
```

A compressed kernel image and loader suitable for placing into flash is provided with the distribution. Use of this image is optional; a kernel that can be directly loaded over the network by RedBoot is also provided.

The image for the IQ80321 can be found under the NetBSD/evbarm 9.0 distribution directory in `evbarm/binary/gzimg/gzimg_IQ80321_flash_0xf0080000.gz`, and corresponds to the kernel in `evbarm/binary/kernel/netbsd-IQ80321.gz`. The following steps describe how to copy the compressed kernel image into flash.

1. Uncompress the gzimg file and place it into the TFTP server's download area (note, the kernel inside the gzimg is still compressed after this step).

   ```
   server# gzcat gzimg_IQ80321_flash_0xf0080000.gz > /tftpboot/gzimg_IQ80321_flash_0xf0080000
   ```

2. Load the gzimg from the TFTP server.

   ```
   RedBoot> load -r -b 0x00200000 gzimg_IQ80321_flash_0xf0080000
   Raw file loaded 0x00200000-0x00305fe4
   ```

3. Use the RedBoot **fis** command to copy the image into flash. For more information on the RedBoot Flash Image System, see the RedBoot documentation.

   ```
   RedBoot> fis create -b 0x00200000 -f 0xf0080000 -l 0x00200000 netbsd.gz
   * CAUTION * about to program 'netbsd.gz'
               at 0xf0080000..0xf027ffff from 0x00200000 - are you sure (y/n)? y
   ... Erase from 0xf0080000-0xf0280000: ....................
   ... Program from 0x00200000-0x00400000 at 0xf0080000: ....................
   ... Unlock from 0xf07e0000-0xf0800000:.
   ... Erase from 0xf07e0000-0xf0800000:.
   ... Program from 0x01ddf000-0x01dff000 at 0xf07e0000:.
   ... Lock from 0xf07e0000-0xf0800000:.
   ```

Once the compressed kernel image has been copied into flash, it may be started by jumping to the flash address of the image:

```
RedBoot> g 0xf0080000

>> NetBSD/IQ80321 Gzip Boot, Revision 1.1
>> (root@tgm, Thu Mar 28 18:32:45 PST 2002)
>> RAM 0xa0000000 - 0xa7ffffff, heap at 0xa7fd0000
>> Load address: 0xa0200000
>> Image size: 739495
Uncompressing image...done.
Jumping to image @ 0xa0200000...

NetBSD/evbarm (IQ80321) booting ...
initarm: Configuring system ...
physmemory: 32768 pages at 0xa0000000 -> 0xa7ffffff
init subsystems: stacks vectors undefined pmap irq done.
[ using 155076 bytes of netbsd ELF symbol table ]
Copyright (c) 1996, 1997, 1998, 1999, 2000, 2001, 2002
```

```
    The NetBSD Foundation, Inc.  All rights reserved.
Copyright (c) 1982, 1986, 1989, 1991, 1993
    The Regents of the University of California.  All rights reserved.

NetBSD 1.5ZC (IQ80321) #1: Thu Mar 28 18:31:58 PST 2002
    root@tgm:/u1/netbsd/src/sys/arch/evbarm/compile/IQ80321
total memory = 128 MB
avail memory = 113 MB
using 1228 buffers containing 6656 KB of memory
mainbus0 (root)
cpu0 at mainbus0: i80321 step A-0 (XScale core)
cpu0: DC enabled IC enabled WB enabled LABT
cpu0: 32KB/32B 32-way Instruction cache
cpu0: 32KB/32B 32-way write-back-locking Data cache
obio0 at mainbus0
com0 at obio0 addr 0xfe800000: ns16550a, working fifo
com0: console
iopxs0 at mainbus0: i80321 I/O Processor, acting as PCI host
iopxs0: configuring PCI bus
pci0 at iopxs0 bus 0
pci0: i/o space, memory space enabled, rd/line, rd/mult, wr/inv ok
wm0 at pci0 dev 4 function 0: Intel i82544 1000BASE-T Ethernet, rev. 2
wm0: interrupting at iop321 irq 27
wm0: Ethernet address 00:07:e9:03:38:40
makphy0 at wm0 phy 1: Marvell 88E1000 Gigabit PHY, rev. 0
makphy0: 10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, 1000baseT, 1000baseT-FDX, au
clock: hz=100 stathz=0 profhz=0
boot device: <unknown>
root device:
```

Installing the NetBSD System

Installation for the Technologic Systems TS-7200

The following steps describe how to set up a CompactFlash card for the TS-7200. The TS-7200 can use the interactive installation program **sysinst** to perform an internet installation from ftp.netbsd.org on the CompactFlash.

1. Place the netbsd-TS7200_INSTALL.bin kernel into the TFTP server's download area. It must be decompressed from the gzipped image in the release directory evbarm/installation/instkernel/netbsd-TS7200_INSTALL.srec.gz

   ```
   server# gunzip netbsd-TS7200_INSTALL.srec.gz
   server# cp netbsd-TS7200_INSTALL.srec
   /tftpboot/netbsd-TS7200_INSTALL
   ```

2. Load the kernel from the TFTP server.

   ```
   RedBoot> load netbsd-TS7200_INSTALL
   ```

3. Start the kernel running.

   ```
   RedBoot> go
   ```

   The kernel will display boot messages, probe for devices, and invoke the interactive sysinst installation tool. Once sysinst has completed installation, the system will reboot back into RedBoot. If the preparation instructions above have been followed, the post-installation NetBSD kernel will be in the FIS ready

to be jumped into. You can have RedBoot automatically start NetBSD by putting the command **go 0x60660000** into the RedBoot bootscript using the **fconfig** command.

Installation for the Intel IQ80310

The following steps describe how to set up an NFS root environment for the IQ80310 and how to boot the IQ80310.

1. Place the kernel into the TFTP server's download area.

   RedBoot only loads S-Records, so select a kernel with the ".srec" filename suffix. In this example, we will boot the kernel hard-wired to use the on-board Ethernet as the root device.

   ```
   server# cp netbsd-fxp0.srec /tftpboot/netbsd-iq80310-fxp0
   ```

2. Unpack the distribution sets into the target's root directory.

   ```
   server# cd /export/client/iq80310
   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.

   ```
   RedBoot> load netbsd-iq80310-fxp0
   Entry point: 0xa0200000, address range: 0xa0200000-0xa035e07c
   RedBoot>
   ```

4. Start the kernel running. The kernel will display boot message, probe for devices, and mount the root file system. If you used a kernel hard-wired to use the on-board Ethernet, it will attempt to mount the root file system automatically. Otherwise, it will prompt for the root device to use.

   Since the system's run-time environment has not yet been configured, the system should boot into single-user mode.

   ```
   RedBoot> go

   NetBSD/evbarm (IQ80310) booting ...
   Resetting secondary PCI bus...
   initarm: Configuring system ...
   physmemory: 65536 pages at 0xa0000000 -> 0xafffffff
   init subsystems: stacks vectors undefined pmap irq done.
   [ using 155084 bytes of netbsd ELF symbol table ]
   Copyright (c) 1996, 1997, 1998, 1999, 2000, 2001, 2002
       The NetBSD Foundation, Inc.  All rights reserved.
   Copyright (c) 1982, 1986, 1989, 1991, 1993
       The Regents of the University of California.  All rights reserved.

   NetBSD 1.5ZC (IQ80310) #20: Fri Mar 29 10:25:53 PST 2002
       root@tgm:/u1/netbsd/src/sys/arch/evbarm/compile/IQ80310
   total memory = 256 MB
   avail memory = 232 MB
   using 1228 buffers containing 13208 KB of memory
   mainbus0 (root)
   cpu0 at mainbus0: i80200 step A-0 (XScale core)
   cpu0: DC enabled IC enabled WB enabled LABT
   cpu0: 32KB/32B 32-way Instruction cache
   cpu0: 32KB/32B 32-way write-back-locking Data cache
   ```

```
obio0 at mainbus0: board rev. F, CPLD rev. D, backplane present
com0 at obio0 addr 0xfe810000: ns16550a, working fifo
com0: console
com1 at obio0 addr 0xfe800000: ns16550a, working fifo
iopxs0 at mainbus0: i80312 Companion I/O, acting as PCI host
iopxs0: configuring Secondary PCI bus
pci0 at iopxs0 bus 1
pci0: i/o space, memory space enabled
ppb0 at pci0 dev 7 function 0: Digital Equipment DECchip 21154 PCI-PCI Bridge (rev
pci1 at ppb0 bus 2
pci1: i/o space, memory space enabled
fxp0 at pci1 dev 0 function 0: i82559S Ethernet, rev 9
fxp0: interrupting at iq80310 irq 1
fxp0: Ethernet address 00:80:4d:46:0b:b9
inphy0 at fxp0 phy 1: i82555 10/100 media interface, rev. 4
inphy0: 10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
clock: hz=100 stathz=0 profhz=0
boot device: <unknown>
root on fxp0
nfs_boot: trying DHCP/BOOTP
nfs_boot: DHCP next-server: 192.168.0.1
nfs_boot: my_name=iq80310.lab.wasabisystems.com
nfs_boot: my_domain=wasabisystems.com
nfs_boot: my_addr=192.168.0.10
nfs_boot: my_mask=255.255.255.0
nfs_boot: gateway=192.168.0.254
root on 192.168.0.1:/export/client/iq80310
/etc/rc.conf is not configured.  Multiuser boot aborted.
Enter pathname of shell or RETURN for sh:
```

Installation for the Intel IQ80321

The following steps describe how to set up an NFS root environment for the IQ80321 and how to boot the IQ80321.

1. Place the kernel into the TFTP server's download area.

   RedBoot only loads S-Records, so select a kernel with the ".srec" filename suffix. In this example, we will boot the kernel hard-wired to use the on-board Ethernet as the root device.

   ```
   server# cp netbsd-wm0.srec /tftpboot/netbsd-iq80321-wm0
   ```

2. Unpack the distribution sets into the target's root directory.

   ```
   server# cd /export/client/iq80321
   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.

   ```
   RedBoot> load netbsd-iq80321-wm0
   Entry point: 0x00200000, address range: 0x00200000-0x00396a40
   RedBoot>
   ```

4. Start the kernel running. The kernel will display boot message, probe for devices, and mount the root file system. If you used a kernel hard-wired to use the on-board Ethernet, it will attempt to mount the root file system automatically. Otherwise, it will prompt for the root device to use.

   Since the system's run-time environment has not yet been configured, the system should boot into single-user mode.

```
RedBoot> go

NetBSD/evbarm (IQ80321) booting ...
initarm: Configuring system ...
physmemory: 32768 pages at 0xa0000000 -> 0xa7ffffff
init subsystems: stacks vectors undefined pmap irq done.
[ using 156468 bytes of netbsd ELF symbol table ]
Copyright (c) 1996, 1997, 1998, 1999, 2000, 2001, 2002
    The NetBSD Foundation, Inc.  All rights reserved.
Copyright (c) 1982, 1986, 1989, 1991, 1993
    The Regents of the University of California.  All rights reserved.

NetBSD 1.5ZC (IQ80321) #1: Thu Mar 28 18:20:34 PST 2002
    root@tgm:/u1/netbsd/src/sys/arch/evbarm/compile/IQ80321
total memory = 128 MB
avail memory = 113 MB
using 1228 buffers containing 6656 KB of memory
mainbus0 (root)
cpu0 at mainbus0: i80321 step A-0 (XScale core)
cpu0: DC enabled IC enabled WB enabled LABT
cpu0: 32KB/32B 32-way Instruction cache
cpu0: 32KB/32B 32-way write-back-locking Data cache
obio0 at mainbus0
com0 at obio0 addr 0xfe800000: ns16550a, working fifo
com0: console
iopxs0 at mainbus0: i80321 I/O Processor, acting as PCI host
iopxs0: configuring PCI bus
pci0 at iopxs0 bus 0
pci0: i/o space, memory space enabled, rd/line, rd/mult, wr/inv ok
wm0 at pci0 dev 4 function 0: Intel i82544 1000BASE-T Ethernet, rev. 2
wm0: interrupting at iop321 irq 27
wm0: Ethernet address 00:07:e9:03:38:40
makphy0 at wm0 phy 1: Marvell 88E1000 Gigabit PHY, rev. 0
makphy0: 10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, 1000baseT, 1000baseT-FDX,
clock: hz=100 stathz=0 profhz=0
boot device: <unknown>
root on wm0
nfs_boot: trying DHCP/BOOTP
nfs_boot: DHCP next-server: 192.168.0.1
nfs_boot: my_name=iq80321.lab.wasabisystems.com
nfs_boot: my_domain=wasabisystems.com
nfs_boot: my_addr=192.168.0.11
nfs_boot: my_mask=255.255.255.0
nfs_boot: gateway=192.168.0.254
root on 192.168.0.1:/export/client/iq80321
/etc/rc.conf is not configured.  Multiuser boot aborted.
```

```
Enter pathname of shell or RETURN for sh:
```

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_fxp0="inet 192.0.2.123 netmask 255.255.255.0"
   ```

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

   ```
   ifconfig_fxp0="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 evbarm/9.0/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
     # pkgin install bash
     # pkgin install perl
     # pkgin install apache
     # pkgin install kde
     # pkgin install firefox
     ...
     ```

     Note: Some mirror sites don't mirror the /pub/pkgsrc directory.

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

   - If you did not install it from the **sysinst** post-installation configuration menu, the pkgsrc(7) framework for compiling packages can be obtained by retrieving the file
     https://cdn.NetBSD.org/pub/pkgsrc/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 9.0 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 9.0 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 9.0.

Note that **sysinst** will automatically invoke

```
postinstall fix
```

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

A number of things have been removed from the NetBSD 9.0 release. See the "Components removed from NetBSD" section near the beginning of this document for a list.

Important note regarding ABI change on ARM ports

In NetBSD 7.0, most ARM ports (all but acorn32, and epoc32) have switched to the official standard ABI (EABI5) which is recommended by ARM for ELF binaries.

Backwards compatibility is provided for binaries using the previous ABI (oabi). A NetBSD 9.0 kernel with the `COMPAT_NETBSD32` option enabled will allow you to execute oabi binaries. This option is enabled in the kernels distributed with this release.

However, new binaries can not be mixed with old libraries, and shared libraries are incompatible.

sysinst does not provide an automatic mechanism to partlially upgrade an old installation. There are two ways to handle the transition:

1. Do a complete update.

   This means updating your system with **sysinst**, then deleting and recompiling all other binaries, whether they were installed locally or through pkgsrc. This is the preferred, cleanest approach.

2. Move your old binaries and libraries to `/compat/netbsd32` and replace them one by one.

   For example, move all of `/usr/pkg` to `/compat/netbsd32/usr/pkg` and add `/compat/netbsd32/usr/pkg/bin` to the end of your PATH. Most binaries should still run, and can be replaced over time with recompiled packages, which will install to `/usr/pkg` again.

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 Projet.
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 Ben Gray.
This product includes software developed by Berkeley Software Design, Inc.
This product includes software developed by Bill Paul.
This product includes software developed by Boris Popov.
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 Chuck Silvers.
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 Daan Vreeken.
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 Frank van der Linden for the NetBSD Project.
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 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 Yasushi Yamasaki
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.
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