

OpenBSD F.A.Q.

August 1, 2010

Contents

Foreword	iii
1 Introduction to OpenBSD	1
1.1 What is OpenBSD?	2
1.2 On what systems does OpenBSD run?	2
1.3 Why might I want to use OpenBSD?	3
1.4 Is OpenBSD really free?	3
1.5 How can I help support OpenBSD?	4
1.6 Who maintains OpenBSD?	5
1.7 When is the next release of OpenBSD?	5
1.8 What is included with OpenBSD?	5
1.9 What is new in OpenBSD 4.7?	7
1.10 Can I use OpenBSD as a desktop system?	7
1.11 Why is/isn't ProductX included?	7
2 Getting to know OpenBSD	9
2.1 Web Pages of Interest	10
2.2 Mailing Lists	10
2.3 Manual Pages	11
2.3.1 Getting started	11
2.3.2 For more advanced users	12
3 Getting started with OpenBSD	15
3.1 Buying an OpenBSD CD set	16
3.2 Buying OpenBSD T-Shirts	16
3.3 Does OpenBSD provide an ISO image for download?	16
3.4 Downloading via FTP, HTTP or AFS	17
3.5 Selecting Hardware	17
3.6 What is an appropriate "first system" to learn OpenBSD on?	19
4 OpenBSD 4.7 Installation Guide	21
4.1 Overview of the OpenBSD installation procedure	22
4.2 Pre-installation checklist	24
4.3 Creating bootable OpenBSD install media	24
4.3.1 Making a CD-ROM	25
4.3.2 Creating floppies on Unix	26
4.3.3 Creating floppies on Windows	26

Contents

4.4 Booting OpenBSD install media 26

Foreword

This FAQ is supplemental documentation to the man pages, available both in the installed system and online. The FAQ covers the active release of OpenBSD, currently v4.7. There are likely features and changes to features in the development version (-current) of OpenBSD that are not covered in this FAQ.

1 Introduction to OpenBSD

1.1 What is OpenBSD?

The OpenBSD project produces a freely available, multi-platform 4.4BSD-based UNIX-like operating system. Our goals place emphasis on correctness, security, standardization, and portability. OpenBSD supports binary emulation of most binaries from SVR4 (Solaris), FreeBSD, Linux, BSDI, SunOS, and HPUX.

This FAQ specifically covers only the most recent release of OpenBSD, version 4.7.

1.2 On what systems does OpenBSD run?

OpenBSD 4.7 runs on the following platforms:

- alpha - FTP only
- amd64 - Available on CD
- armish - FTP only
- hp300 - FTP only
- hppa - FTP only
- i386 - Available on CD
- landisk - FTP only
- loongson - FTP only (new)
- macppc - Available on CD
- mvme68k - FTP only
- mvme88k - FTP only
- sgi - FTP only
- socppc - FTP only
- sparc - FTP only
- sparc64 - Available on CD
- vax - FTP only
- zaurus - FTP only

Available on CD means the official CD set includes that platform and a number of packages. Base system CD ISO images can also be downloaded for most other platforms. More information on OpenBSD platforms can be found on the Platforms page.

1.3 Why might I want to use OpenBSD?

People sometimes ask why we support so many "odd" machines. The short answer is, "because we want to". If enough skilled people (sometimes, "enough" is only one really skilled person!) wish to maintain support for a platform, it will be supported. There are practical benefits to keeping OpenBSD multi-platform: when new platforms come out, the code tree is relatively free of portability-breaking bugs and design flaws. The OpenBSD platforms include 32 bit and 64 bit processors, little and big endian machines, and many different designs. And yes, supporting "unusual" platforms has helped produce a higher-quality code base for more "common" platforms.

1.3 Why might I want to use OpenBSD?

New users frequently want to know whether OpenBSD is superior to some other free UNIX-like operating system. That question is largely unanswerable and is the subject of countless (and useless) religious debates. Do not, under any circumstances, ask such a question on an OpenBSD mailing list.

Below are some reasons why we think OpenBSD is a useful operating system. Whether OpenBSD is right for you is a question that only you can answer.

- OpenBSD runs on many different hardware platforms.
- OpenBSD is thought of by many security professionals as the most secure UNIX-like operating system, as the result of a never-ending comprehensive source code security audit.
- OpenBSD is a full-featured UNIX-like operating system available in source form at no charge.
- OpenBSD integrates cutting-edge security technology suitable for building firewalls and private network services in a distributed environment.
- OpenBSD benefits from strong ongoing development in many areas, offering opportunities to work with emerging technologies with an international community of programmers and end-users.
- OpenBSD attempts to minimize the need for customization and tweaking. For the vast majority of users, OpenBSD "Just Works" on their hardware for their application. Not only is tweaking and customizing rarely needed, it is actively discouraged.

1.4 Is OpenBSD really free?

OpenBSD is all free. The binaries are free. The source is free. All parts of OpenBSD have reasonable copyright terms permitting free redistribution. This includes the ability to REUSE most parts of the OpenBSD source tree, either for personal or commercial purposes. OpenBSD includes NO further restrictions other than those implied by the

1 Introduction to OpenBSD

original BSD license. Software which is written under stricter licenses cannot be included in the regular distribution of OpenBSD. This is intended to safeguard the free use of OpenBSD. For example, OpenBSD can be freely used for personal use, for academic use, by government institutions, by non-profit making organizations and by commercial organizations. OpenBSD, or parts of it, can also be freely incorporated into commercial products.

People sometimes ask if it bothers us that our free work is put into commercial products. The answer is, we would prefer that our good code be widely used rather than have commercial software vendors reimplement and create badly coded or incompatible alternative solutions to already solved problems. For example, it is likely that SSH is a widely used protocol due to this freedom, much more widely used than if restrictions had been placed on how people used the OpenSSH code. If a free SSH solution was not available for vendors to use (in their multitude of rapidly developed products), they would have written or purchased some crummy off-the shelf version instead.

This isn't to say we would object to financial or hardware support in thanks. In fact, it is stunning how little support of any kind comes from companies that depend upon OpenBSD (or OpenSSH) for their products, but there is no requirement of compensation.

For further reading on other popular licenses read: [OpenBSD Copyright Policy](#).

The maintainers of OpenBSD support the project largely from their own pockets. This includes the time spent programming for the project, equipment used to support the many ports, the network resources used to distribute OpenBSD to you, and the time spent answering questions and investigating users' bug reports. The OpenBSD developers are not independently wealthy and even small contributions of time, equipment, and resources make a big difference.

1.5 How can I help support OpenBSD?

We are greatly indebted to the people and organizations that have contributed to the OpenBSD project. They are acknowledged by name on the donations page.

OpenBSD has a constant need for several types of support from the user community. If you find OpenBSD useful, you are strongly encouraged to find a way to contribute. If none of the suggestions below are right for you, feel free to propose an alternative by sending e-mail to donations@openbsd.org.

- Buy an OpenBSD CD set. It includes the current full release of OpenBSD, and is bootable on many platforms. It also generates revenue to support the OpenBSD project, and reduces the strain on network resources used to deliver the distribution via the Internet. This inexpensive three-CD set includes full source. Remember, your friends need their own copy!
- Donate money. The project has a constant need for cash to pay for equipment, network connectivity, and expenses relating to CD publishing. Manufacturing CDs requires an up-front out-of-pocket investment for the OpenBSD developers,

without guaranteed return. Send e-mail to donations@openbsd.org to find out how to contribute. Even small donations make a profound difference.

- Donate equipment and parts. The project has a constant need for general and specific hardware. Items such as IDE and SCSI disks, and various types of RAM are always welcome. For other types of hardware such as computer systems and motherboards, you should inquire as to current need. Write to donations@openbsd.org to arrange for shipment.
- Donate your time and skills. Programmers who enjoy writing operating systems are naturally always welcome, but there are literally dozens of other ways that people can be useful. Follow mailing lists and help answer new-user questions.
- Help maintain documentation by submitting new FAQ material (to faq@openbsd.org). Form a local user group and get your friends hooked on OpenBSD. Make a case to your employer for using OpenBSD at work. If you're a student, talk to your professors about using OpenBSD as a learning tool for Computer Science or Engineering courses. It's also worth mentioning one of the most important ways you should not try to "help" the OpenBSD project: do not waste your time engaging in operating system flame wars. It does not help the project to find new users and can cause substantial harm to important relationships that developers have with other developers.

1.6 Who maintains OpenBSD?

OpenBSD is maintained by a development team spread across many different countries. The project is coordinated by Theo de Raadt, located in Canada.

1.7 When is the next release of OpenBSD?

The OpenBSD team makes a new release every six months, with target release dates in May and November. More information on the development cycle can be found [here](#).

1.8 What is included with OpenBSD?

OpenBSD is distributed with a number of third-party software products, including:

- X.org 7.4, the X Window environment, with local patches. Installed with the `x*.tgz` install file sets.
- GCC versions 2.95.3 and 3.3.5. GNU C Compiler. The OpenBSD team has added the Propolice stack protection technology, enabled by default, and used throughout the OpenBSD userland and by default on applications compiled on OpenBSD. Installed as part of the `comp46.tgz` file set.

1 Introduction to OpenBSD

- Perl 5.10.1, with patches and improvements from the OpenBSD team.
- Our improved and secured version of the Apache 1.3 web server. The OpenBSD team has added default chrooting, privilege revocation, and other security-related improvements. Also includes mod_ssl and DSO support.
- OpenSSL 0.9.8k, with patches and improvements from the OpenBSD team.
- Groff 1.15 text processor.
- Sendmail 8.14.3 mail server, with libmilter.
- BIND 9.4.2-P2 (plus patches) DNS server. OpenBSD has implemented many improvements in chroot operation and other security-related issues.
- Lynx 2.8.6rel.5 text web browser. With HTTPS and IPv6 support added, plus patches from the OpenBSD team.
- Sudo v1.7.2, allowing users to run individual commands as root.
- Ncurses 5.7
- KAME IPv6
- Heimdal 0.7.2 with patches
- Arla 0.35.7
- Binutils 2.15 with patches
- gdb 6.3 with patches
- OpenSSH 5.5
- OpenNTPD Secure and simple Network Time Protocol implementation
- OpenBGPD and OpenOSPFD routing applications

As can be seen, the OpenBSD team often patches third-party products (typically) to improve the security or quality of the code. In some cases, the user will see no difference in operation, in other cases, there ARE operational differences which may impact some users. Keep these enhancements in mind before blindly adding different versions of the same software. You may get a bigger version number, but a less secure system.

Of course, additional applications can be added through the OpenBSD packages and ports system.

1.9 What is new in OpenBSD 4.7?

The complete list of changes made to OpenBSD 4.6 to create OpenBSD 4.7 can be found on [plus46.html](#), and highlights on the OpenBSD 4.7 Information page, however here are a few changes the OpenBSD team anticipate will require or warrant some special note to people upgrading or installing OpenBSD 4.7 who are familiar with older versions:

- OpenBSD/mac68k platform no longer maintained The code is still in the tree, but interest in actively maintaining the port and doing builds is gone.
- Multi-adapter X is broken While X can be run with multiple heads on ONE video card, the ability to use multiple video cards on one system was broken "upstream" by the X.org team.
- Bridge configuration is now done by `ifconfig(8)` `brconfig(8)` is no longer used.
- PF changes Detailed in the upgrade guide.
- `vether(4)` device added
- SSH protocol 1 is disabled by default This only impacts new installs. You have had a very long time to migrate to SSH2, most likely you have.

1.10 Can I use OpenBSD as a desktop system?

This question is often asked in exactly this manner – with no explanation of what the asker means by "desktop". The only person who can answer that question is you, as it depends on what your needs and expectations are.

While OpenBSD has a great reputation as a "server" operating system, it can be and is used on the desktop. Many "desktop" applications are available through packages and ports. As with all operating system decisions, the question is: can it do the job you desire in the way you wish? You must answer this question for yourself.

It might be worth noting that a large amount of OpenBSD development is done on laptops.

1.11 Why is/isn't ProductX included?

People often ask why a particular product is or isn't included with OpenBSD. The answer is based on two things: the wishes of the developers and compatibility with the goals of the project. A product will not be included simply because it is "nifty" – it must also be "free" for use, distribution and modification by our standards. A product must also be stable and secure – a bigger version number does not always mean a better product.

License is often the biggest problem: we want OpenBSD to remain usable by any person anywhere in the world for any purpose.

Another major consideration is the wishes of the developers. The OpenBSD developers are the ultimate judges of what does and doesn't go into the project. Just because an

1 Introduction to OpenBSD

application is "good" doesn't mean the OpenBSD project wishes to devote the resources needed to maintaining it, or that they will share other's enthusiasm about its place in OpenBSD.

Some commonly asked questions about third-party products:

- Why is Sendmail included, it is "known insecure"?! Sendmail has had an imperfect security record, however the Sendmail authors and maintainers have been very receptive to reworking their code to make it much more secure (and this is a sadly uncommon response). The recent security history of Sendmail is not much different than some of the supposedly "more secure" alternatives.
- Why isn't Postfix included? The license is not free, and thus can not be considered.
- Why isn't qmail or djbdns included? Neither program is what many Unix users "expect" out of a mail or DNS application.
- Why is Apache included? It isn't needed by many people! Because the developers want it.
- Why isn't a newer version of Apache included? The license on newer versions is unacceptable.
- Why isn't bzip2 included instead of gzip? Performance is horrible, and benefit is minimal. Impact on slower platforms, such as hp300 or VAX would be unacceptable.
- Why isn't there a graphical or curses(3) based installer? For a number of reasons, including the goal of keeping the installation boot media able to be a single floppy disk, the fact that one installer can be used on all platforms in all configurations, and the fact that after the second or third OpenBSD install, most users find the OpenBSD installation system among the fastest and easiest installers of any OS. Most developers and users greatly prefer the speed, power, and ease of use of the current installer to any of the more "colorful" or "pretty" installers on some other platforms.
- Will ZFS be added to OpenBSD? Not unless someone can convince Oracle to change the license for it to something compatible with OpenBSD policy.

In most cases, these topics have been discussed in painful detail on the mail lists, please see archives if you need more information.

Of course, If you wish to use one of these packages and your use is compatible with the license of the products, no one will stop you (that wouldn't be very free if we tried, would it?). However, your needs may change – you may not want to develop a "Killer Application" that you can't sell, distribute, or get rich from because you incorporated non-free software into it.

2 Getting to know OpenBSD

2.1 Web Pages of Interest

The official website for the OpenBSD project is located at: <http://www.OpenBSD.org>.

A lot of valuable information can be found here regarding all aspects of the OpenBSD project.

The OpenBSD Journal is an OpenBSD-focused news and opinion site.

OpenBSDsupport.org is a site collecting "user maintained" documentation of varying quality, but often covering topics not in this FAQ or other official documentation.

Many users have set up sites and pages with OpenBSD specific information. As with everything on the Internet, a good search engine is going to make your life easier, as will a healthy dose of skepticism. As always, do not blindly enter commands you do not understand into your computer.

2.2 Mailing Lists

The OpenBSD project maintains several popular mailing lists which users should subscribe to and follow. To subscribe to a mailing list, send an e-mail message to majordomo@openbsd.org. That address is an automated subscription service. In the body of your message, on a single line, you should include a subscribe command for the list you wish to join. For example: `subscribe announce`

The list processor will reply to you, asking for confirmation of your intent to join the list, so that others can not subscribe you to a flood of unwanted e-mail. The message will include instructions for several different ways to confirm, including a list server web page link, responding to the confirmation message or responding to majordomo@openbsd.org. Use whatever method is convenient to you. You will note that all three techniques involve a unique and time limited identifying number, such as `A56D-70D4-52C3`, again to make sure you are really the person who requested this mail list subscription (this is real "opt-in").

Once you have confirmed your intent to join, you will be immediately added to the list, and the list processor will notify you that you were successfully added.

To unsubscribe from a list, you will again send an e-mail message to majordomo@openbsd.org. It might look like this: `unsubscribe announce`

If you have any difficulties with the mailing list system, please first read the help file which can be obtained by sending an e-mail message to majordomo@openbsd.org with a message body of "help".

Your subscription to the OpenBSD mail lists can also be maintained through the web interface at <http://lists.openbsd.org>

Some of the more popular OpenBSD mailing lists are:

- `announce` - Important announcements. This is a low-volume list.
- `security-announce` - Announcements of security issues. This is a low volume list.
- `misc` - General user questions and answers. This is the most active list, and should be the "default" for most questions.

- bugs - Bugs received via sendbug(1) and discussions about them.
- source-changes - Automated mailing of CVS source tree changes. Every time a developer commits a change to the OpenBSD source tree, CVS will send out a copy of the (usually brief) commit message via this list.
- ports - Discussion of the OpenBSD Ports Tree.
- ports-changes - Automated mailing of ports-specific CVS source tree changes.
- advocacy - Discussion on advocating OpenBSD, and topics that are just too off-topic for misc.

Before posting a question on misc or any other mailing list, please check the archives, for most common questions have been asked repeatedly. While it might be the first time you have encountered the problem or question, others on the mailing lists may have seen the same question several times in the last week, and may not appreciate seeing it again. If asking a question possibly related to hardware, always include a dmesg(8)!

You can find several archives, other mailing list guidelines and more information on the mailing lists page.

An unofficial mailing list that may be of interest to new users of OpenBSD and Unix is the OpenBSD Newbies list.

2.3 Manual Pages

OpenBSD comes with extensive documentation in the form of manual pages, as well as longer documents relating to specific applications. Considerable effort is made to make sure the man pages are up-to-date and accurate. In all cases, the man pages are considered the authoritative source of information for OpenBSD.

To access the manual pages and other documentation, be sure that you installed the man47.tgz and misc47.tgz file sets.

Here is a list of some of the most useful manual pages for new users:

2.3.1 Getting started

- afterboot(8) - things to check after the first complete boot.
- help(1) - help for new users and administrators.
- hier(7) - layout of filesystems.
- man(1) - display the on-line manual pages.
- intro(1) - introduction to general commands, also see the intros to the other sections of the manual: intro(2), intro(3), intro(4) (note: intro(4) is platform specific), intro(5), intro(6), intro(7), intro(8), and intro(9).

2 Getting to know OpenBSD

- adduser(8) - command for adding new users.
- vipw(8) - edit the master password file.
- disklabel(8) - read and write disk pack label.
- reboot, halt(8) - stop and restart the system.
- shutdown(8) - close down the system at a given time.
- dmesg(8) - redisplay the kernel boot messages
- sudo(8) - don't log in as root, but run commands as root.
- mg(1) - emacs-like text editor.

2.3.2 For more advanced users

- boot(8) - system bootstrapping procedures.
- boot_config(8) - how to change kernel configuration at boot.
- gcc-local(1) - OpenBSD-specific modifications to gcc(1)
- ifconfig(8) - configure network interface parameters.
- login.conf(5) - format of the login class configuration file.
- netstat(1) - show network status.
- release(8) - build an OpenBSD release.
- sendbug(1) - send a problem report (PR) about OpenBSD to a central support site.
- style(9) - OpenBSD kernel source code style guide.
- sysctl(8) - get or set kernel state.

You can find all the OpenBSD man pages on the web at <http://www.openbsd.org/cgi-bin/man.cgi> as well as on your computer if you install the man47.tgz file set.

In general, if you know the name of a command or a manual page, you can read it by executing "man command". For example: "man vi" to read about the vi editor. If you don't know the name of the command, or if "man command" doesn't find the manual page, you can search the manual page database by executing "apropos something" or "man -k something", where "something" is a likely word that might appear in the title of the manual page you're looking for. For example: # apropos "time zone" tzfile (5) - time zone information zdump (8) - time zone dumper zic (8) - time zone compiler

The parenthetical numbers indicate the section of the manual in which that page can be found. In some cases, you may find manual pages with identical names living in separate sections of the manual. For example, assume that you want to know the format of the configuration files for the cron daemon. Once you know the section of the manual for the page you want, you would execute "man n command",

where *n* is the manual section number. # man -k cron cron (8) - clock daemon
crontab (1) - maintain crontab files for individual users crontab (5) - tables for
driving cron # man 5 crontab

In addition to the UNIX manual pages, there is a typesettable document set (included in the misc47.tgz file set). It lives in the /usr/share/doc directory. You can format each document set with a "make" in the appropriate subdirectory. The psd subdirectory is the Programmer's Supplementary Documents distribution. The smm subdirectory is the System Manager's Manual. The usd subdirectory is the UNIX User's Supplementary Documents distribution. You can perform your "make" in the three distribution subdirectories, or you can select a specific section of a distribution and do a 'make' in its subdirectory.

Some of the subdirectories are empty. By default, formatting the documents will result in PostScript output, suitable for printing. The PostScript output can be quite large – you should assume a 250-300% increase in volume. If you do not have access to a PostScript printer or display, you may also format the documents for reading on a terminal display. Each document subdirectory has a target for building ASCII copies of these papers (called 'paper.txt') which can be generated with make(1). For example: # cd /usr/share/doc/usd/06.bc # make paper.txt # more paper.txt

Note that superuser privileges may be required to build documents in these directories, and that issuing make clean will remove any papers generated by a previous make. See /usr/share/doc/README for more details about the documents in /usr/share/doc/.

The UNIX manual pages are generally more current and trustworthy than the typesettable documents. The typesettable documents sometimes explain complicated applications in more detail than the manual pages do.

For many, having a hardcopy of the man page can be useful. Here are the guidelines to making a printable copy of a man page.

3 Getting started with OpenBSD

3.1 Buying an OpenBSD CD set

Purchasing an OpenBSD CD set is generally the best way to get started. Visit the ordering page to purchase your copy: [OpenBSD ordering page](#).

There are many good reasons to own an OpenBSD CD set:

- CD sales support ongoing development of OpenBSD.
- Development of a multi-platform operating system requires constant investment in equipment.
- Your support in the form of a CD set purchase has a real impact on future development.
- The CDs contains binaries (and source) for the most popular supported platforms.
- The CDs are bootable on several platforms, and can be used to bootstrap a machine without a pre-existing installed operating system.
- The CDs are useful for bootstrapping even if you choose to install a snapshot.
- Installing from CD is faster! Installing from CD preserves network connectivity resources.
- OpenBSD CDs always come with very nice stickers. Your system isn't fully complete without these. You can only get these stickers by buying a CD set or donating hardware.
- OpenBSD CD sets come with an assortment of useful and popular packages. The CD set is complete enough to bring up a full work and development environment without any network connection at all.

If you're installing a release version of OpenBSD, you should use a official CD set.

3.2 Buying OpenBSD T-Shirts

Yes, OpenBSD has T-shirts for your wearing enjoyment. You can view these at the [OpenBSD T-shirts page](#). Enjoy :)

3.3 Does OpenBSD provide an ISO image for download?

For select platforms, yes!

Most platforms that are able to boot from CD-ROM have ISO images available for download which can be used to create bootable OpenBSD install CD-ROMs. Both "full install" (all files needed for a full OpenBSD install, named `install47.iso`) and "boot only" (for installing from a network file source, named `cd47.iso`) ISOs are provided for these platforms.

Note, these ISO files are not the same as the official CD set. These images are for single platforms, and do not include any of the pre-compiled packages, stickers, or artwork that the official CD set does.

However, ISO file installation is NOT the optimum installation method for many people. It is still usually faster and simpler to download the boot media and then install just the portions needed. However, for those who wish to do a number of installations, or can not figure out how to drop ten files on a CD-ROM or set up a local FTP server, ISOs are available.

The OpenBSD project does not make the ISO images used to master the official CDs available for download. The reason is simply that we would like you to buy the CD sets to help fund ongoing OpenBSD development. The official OpenBSD CD-ROM layout is copyright Theo de Raadt. Theo does not permit people to redistribute images of the official OpenBSD CDs. As an incentive for people to buy the CD set, some extras are included in the package as well (artwork, stickers etc).

Note that only the CD layout is copyrighted, OpenBSD itself is free. Nothing precludes someone else from downloading OpenBSD and making their own CD.

For those that need a bootable CD for their system, bootdisk ISO images (named `cd47.iso`) are available for a number of platforms which will then permit the rest of the system to be installed via FTP. These ISO images are only a few megabytes in size, and contain just the installation tools, not the actual file sets.

3.4 Downloading via FTP, HTTP or AFS

There are numerous international mirror sites offering FTP and HTTP access to OpenBSD releases and snapshots. AFS access is also available. You should always use the site nearest to you. Before you begin fetching a release or snapshot, you may wish to use `ping(8)` and `traceroute(8)` to determine which mirror site is nearest to you and whether that mirror is performing adequately. Of course, your OpenBSD release CD is always closer than any mirror. Access information is here:

OpenBSD FTP page.

3.5 Selecting Hardware

Selecting appropriate hardware to run your OpenBSD system on is important, as it can mean the difference between success and failure of a project.

If you are shopping for a new PC, whether you are buying it piece by piece or completely pre-built, you want to make sure first that you are buying reliable parts. In the PC world, this is not easy. Bad or otherwise unreliable or mismatched parts can make OpenBSD run poorly and crash often. The best advice we can give is to be careful, and buy brands and parts that have been reviewed by an authority you trust. Sometimes, a higher-price machine is a better quality machine. Other times, it is simply more expensive.

3 Getting started with OpenBSD

There are certain things that will help bring out the maximum performance of your system:

- Let the application chose the hardware: It is usually better to make an adjustment to the hardware you were planning on using rather than compromising on your application design because you have something you "really wanted to use".
- Identify your bottlenecks: Don't pay extra for the cutting-edge processor if your application is restricted by disk I/O. Don't pay for fast disk if your system is restricted by network speed. Don't pay for much of anything if your bottleneck is a 128kbps DSL line.
- Keep it simple: Simple hardware usually has simple problems. Complex hardware that isn't supposed to ever break may take you a long time to repair when it breaks anyway.
- Use hardware you understand, or learn the new hardware before you implement a production system: Regardless of the technical merits of the hardware, committing to use a particular type of hardware before you have become familiar with it, both how it works and how it fails, is foolish.
- Use multiple disks: Instead of buying one large disk, buy multiple smaller disks. While this may cost more, distributing the load over multiple spindles will decrease the amount of time necessary to access data on the disks. And, with more spindles, you can get more reliability and faster data access with RAID.
- Break up large blocks of storage: Many people make the mistake of designing a system that stores large amounts of data to have one Big Block of Storage. This is usually a bad design. You will usually find it much better to break up your storage into manageable blocks. This has many advantages, two of the biggest being that you can add more storage later when you need it easily (you weren't going to get the requirements estimate right, no one ever does), and you can buy a small amount of storage now, and add much more later when the price will have most likely dropped and the capacity will have increased.
- Avoid cheap network adapters: OpenBSD supports a plethora of cheap network adapters. These adapters work great in home systems, and low or moderate throughput business and research environments. But, if you need high throughput and low impact on your server, you are better off buying a quality network adapter. Unfortunately, some name-brand adapters are not much better than the cheap adapters, and some potentially good adapters do not have accurate documentation available to write good drivers. Gigabit adapters often perform better than 10Mbps/100Mbps adapters, even when used on slower speed networks, due to superior buffering.

3.6 What is an appropriate "first system" to learn OpenBSD on?

While OpenBSD will run on a smaller, older and less powerful computer than just about any other modern OS, if you are just getting started with OpenBSD, using too little machine can be frustrating. The following guidelines are ONLY guidelines, OpenBSD will run very well on much more modest equipment than is listed here, but it may be more frustrating than needed for a first-time user.

- Platform: Use a platform you are familiar with already. When you are learning a new operating system, it is a very bad time to also be learning a new platform. We'll assume you are going to be using the i386 platform here, as it is probably the one most people are familiar with.
- Processor: 100MHz Pentium or better processor. Yes, OpenBSD will run on a 25MHz 80486, but you won't want to do the experimenting and messing up and reloading you need to do to really get to know the system on a slow machine. The primary irritation you will first encounter with a processor slower than this is the time it takes to SSH into the box. If you wish to run X, you probably want to move up to at least 200MHz. X actually runs pretty well on a slower machine once loaded, but it takes a while to load and start.
- 64M RAM or better: If you wish to run X, 128M would be a better starting point.
- Hard disk: A 1G hard disk will give you an easy install of a simple system, such as a firewall, DNS server, or similar. If you wish to rebuild the system from source, you will probably want a 4G disk, and if you wish to rebuild X as well, you will want 6G or bigger. IDE is recommended to start with. If you have a much larger disk, don't feel the obligation to allocate all the disk initially – there is nothing wrong with leaving 72G of an 80G hard disk unallocated if all you need is 8G.
- Network adapter: Use a PCI adapter. While the urge to use that old ISA adapter you have may be strong, resist. You probably don't remember how to properly configure it. If you are planning on putting multiple network adapters in the machine, write the MAC address on the spine of the card before putting the cover on.
- Multibooting: For your first OpenBSD installation, don't attempt to multiboot with another OS. Multibooting is a difficult process to get right, and you should understand all the OSs involved well before attempting this, which is clearly not the case on your first installation. It is very possible you could accidentally delete all data on the system. Rather, use a dedicated computer, or at least, a dedicated disk on a machine.
- Laptops: While many laptops work very well with OpenBSD, they are sometimes not the easiest systems to get running well, so a laptop might not be the best

3 *Getting started with OpenBSD*

choice for your first OpenBSD install. However, once you are comfortable with OpenBSD, a laptop can be a very useful tool.

- New hardware: Brand new, cutting-edge hardware is sometimes not yet supported by OpenBSD, so for your first OpenBSD system, a slightly older machine is recommended.

Obviously, "more the better" to a point. Some popular applications, seemingly can use as much processor and memory as you can throw at the system.

4 OpenBSD 4.7 Installation Guide

4.1 Overview of the OpenBSD installation procedure

OpenBSD has long been respected for its simple and straight forward installation process, which is consistent across all platforms.

All platforms use a very similar installation procedure, however there are some minor differences in details on a few platforms. In all cases, you are urged to read the platform-specific INSTALL document in the platform directory on the CD-ROM or FTP sites (for example, `i386/INSTALL.i386`, `macppc/INSTALL.macppc` or `sparc/INSTALL.sparc`).

The OpenBSD installer is a special kernel with a number of utilities and install scripts embedded in a pre-loaded RAM disk. After this kernel is booted, the operating system is extracted from a number of compressed tar(1) (.tgz) files from a source other than this pre-loaded RAM disk. There are several ways to boot this install kernel:

- Floppy disk: OpenBSD can be installed on many platforms by booting an installer from a single floppy disk. However, due to space constraints, some larger platforms (`sparc64`, `amd64`, `alpha`) do not have some utilities which may be important to you, such as a DHCP client to configure the network. For these platforms, you may do better with the CD install. However, for platforms like `i386` and `sparc`, you will find the boot floppy very complete.

Floppy disk images are provided which can be used to create an install floppy on another Unix-like system, or on a Windows system. Typical file names are `floppy47.fs`, though several platforms have multiple floppy images available.

- CD-ROM: On several platforms a CD-ROM image (`cd47.iso` for just booting, or `install47.iso` for the entire install) is provided allowing creation of a bootable CD-ROM.
- Existing partition: The RAM disk kernel can be booted off an already existing partition for an upgrade or reinstall.
- Network: Some platforms support booting over a network (for example using PXE or other network boot).
- Writing a file system image to disk (miniroot): a filesystem image that can be written to an existing partition, and then can be booted.
- Bootable Tape: Some platforms support booting from tape. These tapes can be made following the `INSTALL.platform` instructions.

Not every platform supports all boot options:

- `alpha`: Floppy, CD-ROM, network, writing a floppy image to hard disk.
- `amd64`: Floppy, CD-ROM, network.
- `armish`: Varies by machine.
- `hp300`: CD-ROM, network.

4.1 Overview of the OpenBSD installation procedure

- hppa: Network.
- i386: Floppy, CD-ROM, network.
- landisk: miniroot, installed using another computer.
- macppc: CD-ROM, network.
- mvme68k: Network, bootable tape.
- mvme88k: Network, bootable tape.
- sparc: Floppy, CD-ROM, network, writing image to existing swap partition, bootable tape.
- sparc64: Floppy (U1/U2 only), CD-ROM, network, writing image to existing partition.
- vax: Floppy, network.
- zaurus: Boot `bsd.rd` from Linux partition. See `INSTALL.zaurus` for details.

All platforms can also use a `bsd.rd` to reinstall or upgrade. Once the install kernel is booted, you have several options of where to get the install file sets. Again, not every platform supports every option.

- CD-ROM: Of course, we prefer you use the Official CD-ROM set, but you can also use `install47.iso` or you can also make your own.
- FTP: Either one of the OpenBSD FTP mirror sites or your own local FTP server holding the file sets
- HTTP: Either one of the OpenBSD HTTP mirror sites or your own local web server holding the file sets.
- Local disk partition: In many cases, you can install file sets from another partition on a local hard disk. For example, on i386, you can install from a FAT partition or a CD-ROM formatted in ISO9660, Rock Ridge or Joliet format. In some cases, you will have to manually mount the file system before using it.
- NFS: Some platforms support using NFS mounts for the file sets.
- Tape: File sets can also be read from a supported tape. Details on creating the tape are in the `INSTALL.platform` document.

4.2 Pre-installation checklist

Before you start your install, you should have some idea what you want to end up with. You will want to know the following items, at least:

- Machine name
- Hardware installed and available
 - Verify compatibility with your platform's hardware compatibility page
 - If ISA, you also need to know hardware settings, and confirm they are as OpenBSD requires.
- Install method to be used (CD-ROM, FTP, etc.)
- Should an important bug be found, how will the system be patched?
 - If done locally, you will need to have sufficient space available for the source tree and building it.
 - Otherwise, you will need access to another machine to build a patched release on.
- Desired disk layout
 - Does existing data need to be saved elsewhere?
 - Will OpenBSD coexist on this system with another OS? If so, how both will be booted? Will you need to install a "boot manager"?
 - Will the entire disk be used for OpenBSD, or do you want to keep an existing partition/OS (or space for a future one)?
 - How do you wish to sub-partition the OpenBSD part of your disk?
- Network settings, if not using DHCP:
 - Domain name
 - Domain Name Server(s) (DNS) address
 - IP addresses and subnet masks for each NIC
 - Gateway address
- Will you be running the X Window System?

4.3 Creating bootable OpenBSD install media

As examples, we will look at the installation images available for the i386 and sparc platforms.

The i386 platform has six separate installation disk images to choose from:

4.3 Creating bootable OpenBSD install media

- `cd47.iso` is an ISO9660 image that can be used to create a bootable CD with most popular CD-ROM creation software on most platforms. This image has the widest selection of drivers, and is usually the recommended choice if your hardware can boot from a CDROM.
- `cdemu47.iso` is an ISO9660 image, using "floppy emulation" booting, using a 2.88M floppy image. It is hoped that few people will need this image – most people will use `cd47.iso`, only use `cdemu47.iso` if `cd47.iso` doesn't work for you.
- `install47.iso` is an ISO9660 image, containing all the standard install files. This file can be used to create a CD that can do a stand-alone OpenBSD install.
- `floppy47.fs` (Desktop PC) supports many PCI and ISA NICs, IDE and simple SCSI adapters and some PCMCIA support. Most users will use this image if booting from a floppy
- `floppyB47.fs` (Servers) supports many RAID controllers, and some of the less common SCSI adapters. However, support for many standard SCSI adapters and many EISA and ISA NICS has been removed.
- `floppyC47.fs` (Laptops) supports the CardBus and PCMCIA devices found in many laptops.

The sparc platform has four separate installation disk images to choose from:

- `floppy47.fs`: Supports systems with a floppy disk.
- `cd47.iso` An ISO image usable to make your own CD for booting SPARC systems with a CD-ROM.
- `miniroot47.fs` Can be written to a swap partition and booted.
- `install47.iso` is an ISO9660 image, containing all the standard install files. This file can be used to create a CD that can do a stand-alone OpenBSD install.

On modern platforms, you are best advised to use the CDROM boot images, as in some of the "bigger" platforms (such as `amd64`, `sparc64`), the floppy images have had to have a lot of drivers and utilities cut out, which can make installation much more difficult. Older platforms, such as `i386` and `sparc`, are still quite installable from floppy.

4.3.1 Making a CD-ROM

You can create a CD-ROM using the `cd47.iso` or `install47.iso` files. The exact details here are left to the reader to determine with the tools they have at their disposal.

In OpenBSD, you can create a CD from an ISO image using `cdio(1)`: `# cdio tao cd47.iso` Most CD recorders sold for Windows and Macintosh systems come with software that can burn ISO images to blank media. If yours does not, there are various no-cost applications that can do this for you. Other Unix-like systems use applications such as `cdrkit`.

4.3.2 Creating floppies on Unix

Precise details and device names may vary from Unix variant to Unix variant; use what is appropriate for your system. To create a formatted floppy, use the `fdformat(1)` command to both format and check for bad sectors.

```
# fdformat /dev/rfd0c Format 1440K floppy '/dev/rfd0c'? (y/n): y
Processing VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV done.
```

If your output is like the above example, then the disk is OK. However, if you do not see ALL "V"'s then the disk is most likely bad, and you should try a new one.

To write the image file to the disk, use `dd(1)`. An example usage of `dd(1)` is below:

```
# dd if=floppy47.fs of=/dev/rfd0c bs=32k
```

Once the image is written, check to make sure that the copied image is the same as the original with the `cmp(1)` command. If the diskette is identical to the image, you will just see another prompt.

```
# cmp /dev/rfd0c floppy47.fs
```

4.3.3 Creating floppies on Windows

To prepare a floppy in Windows, first use the native formatting tools to format the disk, verifying that the disk has no bad sectors.

To write the installation image to the prepared floppy you can use `ntrw.exe`, which can be downloaded from the tools directory on any of the FTP mirrors.

Example usage of `ntrw`:

```
C:\> ntrw floppy47.fs a:
3.5", 1.44MB, 512 bytes/sector
bufsize is 9216
1474560 bytes writte
```

4.4 Booting OpenBSD install media

Booting i386/amd64

Booting an install media on the i386 and amd64 PC platforms is nothing new to most people. Your system will have to be instructed to boot from whatever media you have chosen to use, usually either through a BIOS setup option. If you want to boot from CD, your system BIOS must be able to and be set to boot from CD. Some older systems do not have this option, and you must use a floppy for booting your installation image. Don't worry though; even if you boot from floppy you can still install from the CD if it is supported by OpenBSD (i.e., almost all IDE drives).

You can also install by booting `bsd.rd` from an existing OpenBSD partition, or over the network using the PXE boot process.

Booting sparc/sparc64

NOTE: On the sparc64 platform, only the SBus machines (Ultra 1, Ultra 2) are bootable from floppy.

4.4 Booting OpenBSD install media

You will need the system to be at a monitor ROM prompt, which typically looks like "ok ". If you are using a Sun keyboard, press and hold "STOP" while tapping "A". If using a serial console, a BREAK should return you to the monitor prompt.

Use the following command to boot from the floppy:

```
ok boot floppy
```

Usually, you can boot from the CDROM drive of a Sun system from the boot prompt by typing 'boot cdrom':

```
ok boot cdrom
```