Frequently Asked Questions for FreeBSD

Abstract

This is the Frequently Asked Questions (FAQ) for FreeBSD. Every effort has been made to make this FAQ as informative as possible.

Table of Contents

1. Introduction . . . 3
2. Installing FreeBSD . . . 7
3. Hardware . . . 9
4. Documentation and Support . . . 11
5. Community Questions . . . 13
6. Miscellaneous Questions . . . 14
7. Acknowledgments . . . 18

Chapter 1. Introduction

Welcome to the world of FreeBSD! In this section, we provide a comprehensive overview of what FreeBSD is, its origins, goals, use cases, and the community behind it.

Discover the reasons behind its name, learn how to contribute to this project, and explore the rich ecosystem of FreeBSD within the context of open-source operating systems.

1.1. What is FreeBSD?

FreeBSD is a versatile and open-source UNIX®-like operating system known for its exceptional stability, security, and performance. Developed by a dedicated community of volunteers, it's based on the Berkeley Software Distribution (BSD) UNIX operating system.

FreeBSD offers a powerful and customizable environment suitable for a wide range of applications, from servers and embedded systems to desktops and networking equipment. Its commitment to open source principles ensures a transparent and collaborative development process, making FreeBSD a trusted choice for those seeking a reliable and highly adaptable operating system.

1.2. Why is it called FreeBSD?

The name "FreeBSD" stems from its roots in the Berkeley Software Distribution (BSD) UNIX operating system, renowned for its contributions to the world of open-source software. The "Free" in FreeBSD signifies its commitment to the principles of free and open-source software, which grant users the freedom to study, modify, and distribute the code.

It is worth pointing out that the word "free" is being used in two ways here: one meaning "at no cost" and the other meaning "do whatever you like".

1.3. What is the goal of the FreeBSD Project?

The FreeBSD Project has a clear and unwavering goal: to provide a high-quality, open-source UNIX-like operating system that excels in terms of performance, security, and stability.

It aims to offer a versatile platform suitable for a wide range of computing needs, from servers and workstations to embedded systems and beyond. FreeBSD is developed with a strong commitment to open source principles, fostering a collaborative environment where contributions from a global community of developers help shape and refine the operating system. This dedication to quality, freedom, and reliability is what sets FreeBSD apart and drives its ongoing success as an open-source project.

1.4. What use cases is FreeBSD good for?

FreeBSD is a versatile operating system that excels in various use cases. It is particularly well-suited for server environments, where its stability and performance make it a popular choice for web hosting, databases, and networking applications. FreeBSD's robust security features also position it as a strong candidate for firewall and security appliance deployments. Beyond servers, FreeBSD

can be tailored to function in specialized environments, including embedded systems and game console devices. Its adaptability, reliability, and open-source nature make FreeBSD a compelling choice for a wide range of computing needs.

1.5. Who is responsible for FreeBSD?

FreeBSD is a community-driven open-source project with a decentralized structure. Its development and maintenance are carried out by a global community of dedicated volunteers, developers, and organizations who collaborate to enhance and expand the operating system.

The key decisions concerning the FreeBSD project, such as the overall direction of the project or who is allowed to add code to the source tree, are made by an elected Core Team of nine people.

This collaborative and community-driven approach has been fundamental to FreeBSD's success and longevity as a reliable and robust UNIX-like operating system.

1.6. How can I contribute to FreeBSD? What can I do to help?

We accept all types of contributions: documentation, code, and even art. See the article on Contributing to FreeBSD for specific advice on how to do this.

And thanks for the thought!

1.7. Does the FreeBSD license have any restrictions?

FreeBSD is distributed under the BSD License, which is known for its permissive nature.

This license places very few restrictions on how you can use FreeBSD:

- Do not claim that you wrote this.
- Do not sue us if it breaks.
- Do not remove or modify the license.

The license means you are free to modify, distribute, and even sell FreeBSD without **being required to release your modifications as open source**. However, there are some minimal conditions, such as retaining the original copyright notice and disclaimers when distributing FreeBSD. Overall, the BSD License provides a high degree of freedom and flexibility, making FreeBSD an attractive choice for a wide range of applications and projects.

Code in our source tree which falls under the GNU General Public License (GPL) or GNU Library General Public License (LGPL) comes with slightly more strings attached though, at least on the side of enforced access rather than the usual opposite.

1.8. Can FreeBSD replace my current operating system?

For many users and administrators, yes. But this question is not quite that cut-and-dried.

FreeBSD is a powerful and versatile operating system that can replace or coexist with many other operating systems, depending on users and administrators specific needs. However, whether FreeBSD can replace your current operating system depends on factors such as your hardware, software requirements, and familiarity with FreeBSD.

While it offers a robust and feature-rich alternative, it's essential to evaluate your specific use case and compatibility requirements before making the switch.

If an application is only available on one operating system, that operating system cannot just be replaced.

Users migrating to FreeBSD from another UNIX-like environment will find FreeBSD to be similar. Non-UNIX users, like Windows® users, should expect to invest some additional time learning the UNIX way of doing things.

1.9. Can FreeBSD run popular open source software?

Yes, FreeBSD is well-suited for running popular open source software. Its compatibility with a wide range of applications and libraries makes it a favorable choice for those looking to deploy and use open source software packages. FreeBSD provides a robust and stable environment that supports various programming languages, databases, web servers, and other software commonly used in the open source community. Its ports and packages system simplifies the installation and management of such software, ensuring that users can easily access and run their favorite open source tools and applications on FreeBSD with minimal hassle.

1.10. How can I install software in FreeBSD?

FreeBSD offers multiple methods for software installation. One of the most common methods involves using the built-in pkg(8) package manager, which simplifies the process by fetching and installing pre-built binary packages. Another approach is to compile and install software from source code using the ports(7) collection, providing a flexible and customizable way to install software.

FreeBSD's documentation offer detailed guidance on both methods, ensuring that users can easily expand their system's capabilities with the software they need.

1.11. What are the differences between FreeBSD, NetBSD, OpenBSD, and other open source BSD operating systems?

FreeBSD, NetBSD, OpenBSD and DragonFly BSD are all part of the open source BSD family, sharing

common UNIX-like foundations, but each has its own distinct focus and priorities. These differences reflect the unique goals of each project, and while they share similarities, their specific strengths and emphases cater to different use cases and preferences within the BSD ecosystem.

1.12. Is FreeBSD a Linux® distribution?

No, FreeBSD is **not** a Linux distribution.

While both FreeBSD and Linux are UNIX-like operating systems and share many similarities, they have distinct kernels. Linux uses the Linux kernel, whereas FreeBSD uses the FreeBSD kernel, which is based on the Berkeley Software Distribution (BSD) UNIX operating system.

FreeBSD and Linux each have their own unique development communities, release cycles, and system architectures, making them separate and independent operating systems.

FreeBSD provides its own set of advantages, including a different licensing model, system design, and userland utilities compared to Linux distributions.

1.13. Is it possible to run Linux programs on FreeBSD?

Yes, FreeBSD provides a compatibility layer known as *linuxulator* (linux(4)), allowing many Linux programs to function on FreeBSD. This feature facilitates the execution of a broad spectrum of Linux binaries without the necessity of a dedicated Linux environment.

Nevertheless, it's important to note that numerous widely used open-source software **have native FreeBSD versions available** through the ports and packages system.

Chapter 2. Installing FreeBSD

The process of installing FreeBSD is the initial step toward harnessing the power of this robust open-source operating system. This section provides essential information on where to obtain FreeBSD, detailed installation instructions, and insights into concepts like FreeBSD-CURRENT and FreeBSD-STABLE. It also delves into the release and snapshot schedules, post-installation configuration tools, package search methods, and addresses common questions regarding package updates.

2.1. Where can I get FreeBSD?

FreeBSD is freely available for download from the official FreeBSD website.

Additionally, FreeBSD may also be available through various mirror sites, ensuring accessibility to users worldwide. The official website is the primary and most reliable source for obtaining the latest FreeBSD releases and updates, making it the ideal starting point for a FreeBSD journey.

2.2. Where are the instructions for installing FreeBSD?

Installation instructions can be found at the Installing FreeBSD Chapter of the Handbook.

2.3. What is the FreeBSD-CURRENT concept?

FreeBSD-CURRENT represents the development branch of the FreeBSD operating system. It is the most cutting-edge version of FreeBSD, where active development takes place.

While it incorporates the latest features, improvements, and experimental changes, it may not always be as stable as the FreeBSD-STABLE or release versions.

FreeBSD-CURRENT serves as a platform for developers and enthusiasts who want to contribute to the future of FreeBSD and stay on the leading edge of its development, even though it may occasionally undergo significant changes and **is not recommended for production systems**.

2.4. What is the FreeBSD-STABLE concept?

The FreeBSD-STABLE branch is a more stable development branch compared to CURRENT. It contains code that is undergoing further testing and refinement before being merged into the RELEASE branch. STABLE is a good choice for those who want to track FreeBSD's development but prefer a more stable environment than CURRENT.

2.5. When are FreeBSD releases made?

The Release Engineering Team <re@FreeBSD.org> releases a new major version of FreeBSD about every 18 months and a new minor version about every 8 months, on average. Release dates are announced well in advance, so that the people working on the system know when their projects need to be finished and tested. A testing period precedes each release, to ensure that the addition of

new features does not compromise the stability of the release.

2.6. When are FreeBSD snapshots made?

FreeBSD snapshots are typically generated at regular intervals for all actively developed branches. These snapshots capture a moment in time of the FreeBSD source code and associated binary packages. The frequency of these snapshots may vary but is often done on a weekly or bi-weekly basis. These snapshots provide users with an opportunity to access the latest developments and changes in FreeBSD, helping them stay up-to-date with the project's progress.

2.7. Is there a tool to perform post-installation configuration tasks?

Yes. bsdconfig(8) provides a nice interface to configure FreeBSD post-installation.

2.8. How can I search for software to install in FreeBSD?

Searching for software to install in FreeBSD is made easy through the pkg(8) package manager and the `pkg search` command.

Users can utilize this command to search for available packages using keywords, package names, or descriptions. This feature helps users quickly locate the software they need within the extensive FreeBSD ports and packages collection, simplifying the process of adding new applications and tools to their FreeBSD systems.

2.9. Why am I not getting the latest packages in my FreeBSD system?

The availability of the latest packages in FreeBSD can be influenced by various factors, including the package repository's update frequency and the specific version of FreeBSD being used.

It should also be added that there are two branches for the FreeBSD ports. The most updated is called `latest` and the most stable is called `quarterly`. To use the latest packages it will be necessary to configure the `latest` branch. Information on how to configure branches can be found in the Ports chapter of the Handbook.

Chapter 3. Hardware

The Hardware section explores FreeBSD's compatibility with various hardware configurations. It covers a range of topics, including supported architectures, maximum RAM capacity, processor scalability, graphics card compatibility, Wi-Fi card support, and the inclusion of 10 Gigabit Ethernet network card support. Whether planning to deploy FreeBSD on servers, workstations, or specialized hardware, this section provides insights into FreeBSD's capabilities and limitations, offering information about hardware choices and configurations.

3.1. What architectures does FreeBSD support?

FreeBSD boasts a versatile range of supported architectures, making it a flexible choice for a variety of hardware environments. Its compatibility extends to architectures such as `amd64`, `arm64`, `riscv` and more.

A complete list of supported architectures can be found on the platforms page.

3.2. How much RAM does FreeBSD support?

FreeBSD as an operating system generally supports as much physical memory (RAM) as the platform it is running on does. Keep in mind that different platforms have different limits for memory.

For example, `amd64` platforms support up to 4TB of physical memory.

3.3. How many processors does FreeBSD support?

FreeBSD's processor support varies across different architectures. On amd64 and arm64 systems, FreeBSD can efficiently harness the power of up to 1024 processors. For powerpc-based systems, FreeBSD supports up to 256 processors, while on risc-v systems, it can effectively utilize up to 16 processors.

3.4. What graphics cards does FreeBSD support?

Popular graphics vendors like Intel®, AMD® or NVIDIA® are well-supported.

A list of supported graphics cards from Intel and AMD can be found in the FreeBSD Wiki.

3.5. What Wi-Fi cards does FreeBSD support?

This is a rapidly changing area as of 2025, so it's worth checking the Hardware Compatibility List for your chosen release.

The list of PCIe Wi-Fi devices that are known to work well on FreeBSD:

- Intel chipset devices covered by iwlwifi(4) (high-speed as of FreeBSD 14.3)

- Realtek RTL8188CE Mini PCIe

The list of USB Wi-Fi devices that are known to work well on FreeBSD:

- RT5370 USB dongles (supports hostap mode)
- TP-Link TL-WDN3200 (RT5592, if_run(4))
- TP-Link TL-WN725N v2 (RTL8188EU, rtwn(4))
- TP-Link Archer T4U (RTL8812AU, rtwn(4))
- D-Link DWA-131 (RTL8192CU, rtwn(4))
- D-Link DWA-171 rev A1 (RTL8821AU, rtwn(4))
- ASUS USB-N10 NANO (RTL8188CUS, rtwn(4))

3.6. Does FreeBSD support 10 Gigabit Ethernet network cards?

FreeBSD provides robust support for 10 Gigabit Ethernet network cards, making it a viable choice for high-speed networking environments. The FreeBSD kernel includes drivers for a wide range of 10 Gigabit Ethernet adapters from various manufacturers, ensuring seamless integration and optimal network performance.

More information can be obtained at the FreeBSD Wiki.

Chapter 4. Documentation and Support

This section provides valuable resources for those seeking to deepen their knowledge of FreeBSD. It covers a range of topics, including recommended books for learning FreeBSD, avenues for commercial training and support, insights into understanding FreeBSD's internals, and where to find assistance within the FreeBSD community.

4.1. What good books are there about FreeBSD?

FreeBSD enthusiasts and learners can explore a variety of informative books dedicated to the operating system. These books cover topics ranging from FreeBSD basics to advanced system administration and development.

Some notable titles include "Absolute FreeBSD: The Complete Guide To FreeBSD" by Michael W. Lucas, and the "FreeBSD Mastery series" also by Michael W. Lucas, among others. These resources offer valuable insights and knowledge to help users of all levels master FreeBSD and maximize its potential.

Apart from the books, The FreeBSD Project produces a wide range of documentation, available online in the Documentation Portal.

4.2. Where can I get commercial FreeBSD training and support?

For those seeking professional training and support for FreeBSD, several commercial providers offer tailored services. The FreeBSD project maintains a list of companies from which support can be requested.

4.3. How can I learn more about FreeBSD's internals?

To delve deeper into the inner workings of FreeBSD, interested individuals can explore resources like "The Design And Implementation Of The FreeBSD Operating System". The FreeBSD Architecture Handbook is another valuable resource that provides detailed information about FreeBSD's architecture, system organization, and design principles

Additionally, reviewing the FreeBSD source code, available through the FreeBSD source repository, provides valuable insights into the operating system's core components.

4.4. How can I get help in a FreeBSD system?

Getting assistance within a FreeBSD system is straightforward. FreeBSD offers a wealth of built-in documentation, accessible through the man(1) command, which provides comprehensive information about various commands and system components.

Additionally, the FreeBSD Handbook, available both online and locally on the system, serves as a valuable resource for in-depth guidance.

If specific issues arise, seeking help from the FreeBSD community via mailing lists, forums, or IRC can provide timely solutions and insights from experienced FreeBSD users and developers.

Chapter 5. Community Questions

Explore this section to discover ways to engage with the FreeBSD community.

Learn about FreeBSD IRC channels for real-time discussions, web-based forums for sharing insights, and access to FreeBSD mailing lists and news groups for in-depth discussions and news updates.

5.1. Are there FreeBSD IRC (Internet Relay Chat) channels?

Yes, most major IRC networks host a FreeBSD chat channel and the FreeBSD wiki holds an up to date list of IRC channels.

Each of these channels are distinct and are not connected to each other. Since their chat styles differ, try each to find one suited to your chat style.

5.2. Are there any web based forums to discuss FreeBSD?

The official FreeBSD forums are located at the FreeBSD Forums homepage.

5.3. Where do I find info on the FreeBSD mailing lists?

The public mailing lists can be found at the FreeBSD mailing lists.

Chapter 6. Miscellaneous Questions

Explore a variety of intriguing and practical queries about FreeBSD, covering topics from shell choices to system quirks and even the FreeBSD pet's name.

6.1. Why is /bin/sh so minimal? Why does FreeBSD not use bash or another shell?

Many people need to write shell scripts which will be portable across many systems. That is why POSIX® specifies the shell and utility commands in great detail. Most scripts are written in POSIX shell (sh(1)), and several important programming interfaces are specified to use the POSIX shell to interpret commands. As the POSIX shell is so often and widely used, it is important for it to be quick to start, be deterministic in its behavior, and have a small memory footprint.

The existing implementation is our best effort at meeting as many of these requirements simultaneously as we can. To keep `/bin/sh` small, we have not provided many of the convenience features that other shells have.

That is why other more featureful shells like `bash`, `scsh`, tcsh(1), and `zsh` are available.

6.2. How do I use my delete key in sh and csh?

For sh(1), add the following lines to ~/.shrc:

```
bind ^[[3~ ed-delete-next-char # for xterm
```

For the csh(1), add the following lines to ~/.cshrc:

```
bindkey ^[[3~ delete-char # for xterm
```

6.3. I have forgotten the root password! What do I do?

Do not panic! Restart the system and select `Boot Single User` from the boot loader menu, or press `Escape` at the menu and type `boot -s` at the loader prompt, to enter single-user mode. At the question about the shell to use, hit `Enter` which will display a # prompt. When there is a need to change the keyboard layout, run `kbdmap`, select the preferred one and confirm by pressing `Enter`. Enter `mount -urw /` to remount the root file system read/write. To remount all the remaining file systems, run `mount -a` for UFS filesystems and `zfs mount -a` when using ZFS. Run `passwd root` to change the `root` password then run `exit` to continue booting.

If you are still prompted to give the `root` password when entering the single-user mode, it means that the console has been marked as `insecure` in /etc/ttys. In this case, it will be required to boot from a FreeBSD installation disk, choose the Live CD or Shell at the beginning of the install process and issue the commands

mentioned above. Mount the specific partition in this case and then chroot to it. For example, replace `mount -urw /` with `mount /dev/ada0p1 /mnt; chroot /mnt` for a system on *ada0p1*.

If the root partition cannot be mounted from single-user mode, it is possible that the partitions are encrypted and it is impossible to mount them without the access keys. For more information see the section about encrypted disks in the FreeBSD Handbook.

6.4. I made a mistake in rc.conf, or another startup file, and now I cannot edit it because the file system is read-only. What should I do?

Restart the system and select `Boot Single User` from the boot loader menu, or press `Escape` at the menu and type `boot -s` at the loader prompt, to enter single-user mode. When prompted for a shell pathname, press `Enter` and run `mount -urw /` to re-mount the root file system in read/write mode. You may also need to run `mount -a -t ufs` to mount the file system where your favorite editor is defined. If that editor is on a network file system, either configure the network manually before mounting the network file systems, or use an editor which resides on a local file system, such as ed(1).

In order to use a full screen editor such as vi(1) or emacs(1), run `export TERM=xterm` so that these editors can load the correct data from the termcap(5) database.

After performing these steps, edit /etc/rc.conf to fix the syntax error. The error message displayed immediately after the kernel boot messages should indicate the number of the line in the file which is at fault.

6.5. How much free memory is available?

There are a couple of kinds of "free memory". The most common is the amount of memory immediately available without reclaiming memory already in use. That is the size of the free pages queue plus some other reserved pages. This amount is exported by the `vm.stats.vm.v_free_count` sysctl(8), shown, for instance, by top(1). Another kind of "free memory" is the total amount of virtual memory available to userland processes, which depends on the sum of swap space and usable memory. Other kinds of "free memory" descriptions are also possible, but it is relatively useless to define these, but rather it is important to make sure that the paging rate is kept low, and to avoid running out of swap space.

6.6. My time is wrong, how can I change the timezone?

Use tzsetup(8).

6.7. FreeBSD uses a lot of swap space even when the computer has free memory left. Why?

FreeBSD will proactively move entirely idle, unused pages of main memory into swap in order to make more main memory available for active use. This heavy use of swap is balanced by using the extra free memory for caching.

Note that while FreeBSD is proactive in this regard, it does not arbitrarily decide to swap pages when the system is truly idle. Thus, the system will not be all paged out after leaving it idle overnight.

6.8. Why does top show very little free memory even when I have very few programs running?

The simple answer is that free memory is wasted memory.

Any memory that programs do not actively allocate is used within the FreeBSD kernel as disk cache. The values shown by top(1) labeled as `Inact` and `Laundry` are cached data at different aging levels. This cached data means the system does not have to access a slow disk again for data it has accessed recently, thus increasing overall performance. In general, a low value shown for `Free` memory in top(1) is good, provided it is not *very* low.

6.9. What is the cute little red guy's name?

He does not have one, and is just called "the BSD daemon". If you insist upon using a name, call him "beastie". Note that "beastie" is pronounced "BSD".

More about the BSD daemon is available on his home page.

6.10. Can I use the BSD daemon image?

Perhaps. The BSD daemon is copyrighted by Marshall Kirk McKusick. Check his Statement on the Use of the BSD Daemon Figure for detailed usage terms.

In summary, the image can be used in a tasteful manner, for personal use, so long as appropriate credit is given. Before using the logo commercially, contact Kirk McKusick < mckusick@FreeBSD.org> for permission. More details are available on the BSD Daemon's home page.

6.11. Why should I care what color the bikeshed is?

The really, really short answer is that you should not. The somewhat longer answer is that just because you are capable of building a bikeshed does not mean you should stop others from building one just because you do not like the color they plan to paint it. This is a metaphor indicating that you need not argue about every little feature just because you know enough to do so. Some people have commented that the amount of noise generated by a change is inversely

proportional to the complexity of the change.

The longer and complete answer is that after a very long argument about whether sleep(1) should take fractional second arguments, Poul-Henning Kamp <phk@FreeBSD.org> posted a long message entitled A bike shed (any color will do) on greener grass….

Chapter 7. Acknowledgments

This FAQ has undergone countless revisions and improvements by a diverse group of contributors over the past decades.