Explaining BSD

Abstract

In the open source world, the word "Linux" is almost synonymous with "Operating System", but it is not the only open source UNIX® operating system.

So what is the secret? Why is BSD not better known? This white paper addresses these and other questions.

Throughout this paper, differences between BSD and Linux will be noted *like this*.

Table of Contents

1. What is BSD? 1
2. What, a real UNIX®? 2
3. Why is BSD not better known? 3
4. Comparing BSD and Linux 3

1. What is BSD?

BSD stands for "Berkeley Software Distribution". It is the name of distributions of source code from the University of California, Berkeley, which were originally extensions to AT&T's Research UNIX® operating system. Several open source operating system projects are based on a release of this source code known as 4.4BSD-Lite. In addition, they comprise a number of packages from other Open Source projects, including notably the GNU project. The overall operating system comprises:

- The BSD kernel, which handles process scheduling, memory management, symmetric multi-processing (SMP), device drivers, etc.
- The C library, the base API for the system.

  The BSD C library is based on code from Berkeley, not the GNU project.

- Utilities such as shells, file utilities, compilers and linkers.

  Some of the utilities are derived from the GNU project, others are not.

- The X Window system, which handles graphical display.

  The X Window system used in most versions of BSD is maintained by the X.Org project. FreeBSD allows the user to choose from a variety of desktop environments, such as Gnome, KDE, or Xfce; and lightweight window managers like Openbox, Fluxbox, or Awesome.

- Many other programs and utilities.

2. What, a real UNIX®?

The BSD operating systems are not clones, but open source derivatives of AT&T's Research UNIX® operating system, which is also the ancestor of the modern UNIX® System V. This may surprise you. How could that happen when AT&T has never released its code as open source?

It is true that AT&T UNIX® is not open source, and in a copyright sense BSD is very definitely *not* UNIX®, but on the other hand, AT&T has imported sources from other projects, noticeably the Computer Sciences Research Group (CSRG) of the University of California in Berkeley, CA. Starting in 1976, the CSRG started releasing tapes of their software, calling them *Berkeley Software Distribution* or *BSD*.

Initial BSD releases consisted mainly of user programs, but that changed dramatically when the CSRG landed a contract with the Defense Advanced Research Projects Agency (DARPA) to upgrade the communications protocols on their network, ARPANET. The new protocols were known as the *Internet Protocols*, later *TCP/IP* after the most important protocols. The first widely distributed implementation was part of 4.2BSD, in 1982.

In the course of the 1980s, a number of new workstation companies sprang up. Many preferred to license UNIX® rather than developing operating systems for themselves. In particular, Sun Microsystems licensed UNIX® and implemented a version of 4.2BSD, which they called SunOS™. When AT&T themselves were allowed to sell UNIX® commercially, they started with a somewhat bare-bones implementation called System III, to be quickly followed by System V. The System V code base did not include networking, so all implementations included additional software from the BSD, including the TCP/IP software, but also utilities such as the *csh* shell and the *vi* editor. Collectively, these enhancements were known as the *Berkeley Extensions*.

The BSD tapes contained AT&T source code and thus required a UNIX® source license. By 1990, the CSRG's funding was running out, and it faced closure. Some members of the group decided to release the BSD code, which was Open Source, without the AT&T proprietary code. This finally happened with the *Networking Tape 2*, usually known as *Net/2*. Net/2 was not a complete operating system: about 20% of the kernel code was missing. One of the CSRG members, William F. Jolitz, wrote the remaining code and released it in early 1992 as *386BSD*. At the same time, another group of ex-CSRG members formed a commercial company called Berkeley Software Design Inc. and released a beta version of an operating system called BSD/386, which was based on the same sources. The name of the operating system was later changed to BSD/OS.

386BSD never became a stable operating system. Instead, two other projects split off from it in 1993: NetBSD and FreeBSD. The two projects originally diverged due to differences in patience waiting for improvements to 386BSD: the NetBSD people started early in the year, and the first version of FreeBSD was not ready until the end of the year. In the meantime, the code base had diverged sufficiently to make it difficult to merge. In addition, the projects had different aims, as we will see below. In 1996, OpenBSD split off from NetBSD, and in 2003, DragonFlyBSD split off from FreeBSD.

3. Why is BSD not better known?

For a number of reasons, BSD is relatively unknown:

1. The BSD developers are often more interested in polishing their code than marketing it.
2. Much of Linux's popularity is due to factors external to the Linux projects, such as the press, and to companies formed to provide Linux services. Until recently, the open source BSDs had no such proponents.
3. In 1992, AT&T sued BSDI, the vendor of BSD/386, alleging that the product contained AT&T-copyrighted code. The case was settled out of court in 1994, but the spectre of the litigation continues to haunt people. In March 2000 an article published on the web claimed that the court case had been "recently settled".

   One detail that the lawsuit did clarify is the naming: in the 1980s, BSD was known as "BSD UNIX®". With the elimination of the last vestige of AT&T code from BSD, it also lost the right to the name UNIX®. Thus you will see references in book titles to "the 4.3BSD UNIX® operating system" and "the 4.4BSD operating system".

4. Comparing BSD and Linux

So what is really the difference between, say, Debian Linux and FreeBSD? For the average user, the difference is surprisingly small: Both are UNIX® like operating systems. Both are developed by non-commercial projects (this does not apply to many other Linux distributions, of course). In the following section, we will look at BSD and compare it to Linux. The description applies most closely to FreeBSD, which accounts for an estimated 80% of the BSD installations, but the differences from NetBSD, OpenBSD and DragonFlyBSD are small.

4.1. Who owns BSD?

No one person or corporation owns BSD. It is created and distributed by a community of highly technical and committed contributors all over the world. Some of the components of BSD are Open Source projects in their own right and managed by different project maintainers.

4.2. How is BSD developed and updated?

The BSD kernels are developed and updated following the Open Source development model. Each project maintains a publicly accessible *source tree* which contains all source files for the project, including documentation and other incidental files. Users can obtain a complete copy of any version.

A large number of developers worldwide contribute to improvements to BSD. They are divided into three kinds:

- *Contributors* write code or documentation. They are not permitted to commit (add code) directly to the source tree. For their code to be included in the system, it must be reviewed and checked in by a registered developer, known as a *committer*.

- *Committers* are developers with write access to the source tree. To become a committer, an individual must show ability in the area in which they are active.

  It is at the individual committer's discretion whether they should obtain authority before committing changes to the source tree. In general, an experienced committer may make changes which are obviously correct without obtaining consensus. For example, a documentation project committer may correct typographical or grammatical errors without review. On the other hand, developers making far-reaching or complicated changes are expected to submit their changes for review before committing them. In extreme cases, a core team member with a function such as Principal Architect may order that changes be removed from the tree, a process known as *backing out*. All committers receive mail describing each individual commit, so it is not possible to commit secretly.

- The *Core team*. FreeBSD and NetBSD each have a core team which manages the project. The core teams developed in the course of the projects, and their role is not always well-defined. It is not necessary to be a developer to be a core team member, though it is normal. The rules for the core team vary from one project to the other, but in general they have more say in the direction of the project than non-core team members have.

This arrangement differs from Linux in a number of ways:

1. No one person controls the content of the system. In practice, this difference is overrated, since the Principal Architect can require that code be backed out, and even in the Linux project several people are permitted to make changes.
2. On the other hand, there *is* a central repository, a single place where you can find the entire operating system sources, including all older versions.
3. BSD projects maintain the entire "Operating System", not only the kernel. This distinction is only marginally useful: neither BSD nor Linux is useful without applications. The applications used under BSD are frequently the same as the applications used under Linux.
4. As a result of the formalized maintenance of a single Git source tree, BSD development is clear, and it is possible to access any version of the system by release number or by date. Git also allows incremental updates to the system: for example, the FreeBSD repository is updated about 100 times a day. Most of these changes are small.

4.3. BSD releases

FreeBSD, NetBSD and OpenBSD provide the system in three different "releases". As with Linux, releases are assigned a number such as 1.4.1 or 3.5. In addition, the version number has a suffix indicating its purpose:

1. The development version of the system is called *CURRENT*. FreeBSD assigns a number to CURRENT, for example FreeBSD 5.0-CURRENT. NetBSD uses a slightly different naming scheme and appends a single-letter suffix which indicates changes in the internal interfaces, for example NetBSD 1.4.3G. OpenBSD does not assign a number ("OpenBSD-current"). All new development on the system goes into this branch.
2. At regular intervals, between two and four times a year, the projects bring out a *RELEASE* version of the system, which is available on CD-ROM and for free download from FTP sites, for

example OpenBSD 2.6-RELEASE or NetBSD 1.4-RELEASE. The RELEASE version is intended for end users and is the normal version of the system. NetBSD also provides *patch releases* with a third digit, for example NetBSD 1.4.2.

3. As bugs are found in a RELEASE version, they are fixed, and the fixes are added to the Git tree. In FreeBSD, the resultant version is called the *STABLE* version, while in NetBSD and OpenBSD it continues to be called the RELEASE version. Smaller new features can also be added to this branch after a period of test in the CURRENT branch. Security and other important bug fixes are also applied to all supported RELEASE versions.

By contrast, Linux maintains two separate code trees: the stable version and the development version. Stable versions have an even minor version number, such as 2.0, 2.2 or 2.4. Development versions have an odd minor version number, such as 2.1, 2.3 or 2.5. In each case, the number is followed by a further number designating the exact release. In addition, each vendor adds their own userland programs and utilities, so the name of the distribution is also important. Each distribution vendor also assigns version numbers to the distribution, so a complete description might be something like "TurboLinux 6.0 with kernel 2.2.14"

4.4. What versions of BSD are available?

In contrast to the numerous Linux distributions, there are only four major open source BSDs. Each BSD project maintains its own source tree and its own kernel. In practice, though, there appear to be fewer divergences between the userland code of the projects than there is in Linux.

It is difficult to categorize the goals of each project: the differences are very subjective. Basically,

- FreeBSD aims for high performance and ease of use by end users, and is a favourite of web content providers. It runs on a number of platforms and has significantly more users than the other projects.
- NetBSD aims for maximum portability: "of course it runs NetBSD". It runs on machines from palmtops to large servers, and has even been used on NASA space missions. It is a particularly good choice for running on old non-Intel® hardware.
- OpenBSD aims for security and code purity: it uses a combination of the open source concept and rigorous code reviews to create a system which is demonstrably correct, making it the choice of security-conscious organizations such as banks, stock exchanges and US Government departments. Like NetBSD, it runs on a number of platforms.
- DragonFlyBSD aims for high performance and scalability under everything from a single-node UP system to a massively clustered system. DragonFlyBSD has several long-range technical goals, but focus lies on providing a SMP-capable infrastructure that is easy to understand, maintain and develop for.

There are also two additional BSD UNIX® operating systems which are not open source, BSD/OS and Apple's Mac OS® X:

- BSD/OS was the oldest of the 4.4BSD derivatives. It was not open source, though source code licenses were available at relatively low cost. It resembled FreeBSD in many ways. Two years after the acquisition of BSDi by Wind River Systems, BSD/OS failed to survive as an independent product. Support and source code may still be available from Wind River, but all new

development is focused on the VxWorks embedded operating system.

- Mac OS® X is the latest version of the operating system for Apple®'s Mac® line. The BSD core of this operating system, Darwin, is available as a fully functional open source operating system for x86 and PPC computers. The Aqua/Quartz graphics system and many other proprietary aspects of Mac OS® X remain closed-source, however. Several Darwin developers are also FreeBSD committers, and vice-versa.

4.5. How does the BSD license differ from the GNU Public license?

Linux is available under the GNU General Public License (GPL), which is designed to eliminate closed source software. In particular, any derivative work of a product released under the GPL must also be supplied with source code if requested. By contrast, the BSD license is less restrictive: binary-only distributions are allowed. This is particularly attractive for embedded applications.

4.6. What else should I know?

Since fewer applications are available for BSD than Linux, the BSD developers created a Linux compatibility package, which allows Linux programs to run under BSD. The package includes both kernel modifications, to correctly perform Linux system calls, and Linux compatibility files such as the C library. There is no noticeable difference in execution speed between a Linux application running on a Linux machine and a Linux application running on a BSD machine of the same speed.

The "all from one supplier" nature of BSD means that upgrades are much easier to handle than is frequently the case with Linux. BSD handles library version upgrades by providing compatibility modules for earlier library versions, so it is possible to run binaries which are several years old with no problems.

4.7. Which should I use, BSD or Linux?

What does this all mean in practice? Who should use BSD, who should use Linux?

This is a very difficult question to answer. Here are some guidelines:

- "If it ain't broke, don't fix it": If you already use an open source operating system, and you are happy with it, there is probably no good reason to change.
- BSD systems, in particular FreeBSD, can have notably higher performance than Linux. But this is not across the board. In many cases, there is little or no difference in performance. In some cases, Linux may perform better than FreeBSD.
- In general, BSD systems have a better reputation for reliability, mainly as a result of the more mature code base.
- BSD projects have a better reputation for the quality and completeness of their documentation. The various documentation projects aim to provide actively updated documentation, in many languages, and covering all aspects of the system.
- The BSD license may be more attractive than the GPL.

- BSD can execute most Linux binaries, while Linux can not execute BSD binaries. Many BSD implementations can also execute binaries from other UNIX® like systems. As a result, BSD may present an easier migration route from other systems than Linux would.

4.8. Who provides support, service, and training for BSD?

Each of the projects has a list of consultants for hire: FreeBSD, NetBSD, and OpenBSD.