

Chapter 1

What Is OpenSolaris?

You probably wouldn't have picked up this book if you hadn't at least heard of OpenSolaris or Solaris, but even if you've poked around OpenSolaris or used Solaris for years, you might be confused about what, exactly, OpenSolaris is. Is it an operating system, an open source code base, an open source community, or a distribution? How is it different from Solaris? How is it different from Linux? Is it really open source?

This chapter answers those questions and more. Even if you're an experienced Solaris user, this chapter may be useful in helping you understand OpenSolaris and its differences from Solaris. On the other hand, if you're already involved in OpenSolaris, you might still want to skim this chapter to learn a bit about the history of OpenSolaris and Solaris with which you might not be familiar.

Introduction to OpenSolaris

OpenSolaris is an open source operating system, similar in scope to GNU/Linux and BSD, but descended from the proprietary Solaris operating system from Sun Microsystems. The authors of this book find it helpful to think of OpenSolaris as divided into three distinct but related aspects: the code, the distributions, and the community.

OpenSolaris code

OpenSolaris is the open source version of Sun Microsystems' Solaris operating system, but OpenSolaris consists of code for much more than

IN THIS CHAPTER

Introduction to OpenSolaris

OpenSolaris features

OpenSolaris licensing

History of OpenSolaris

Comparing OpenSolaris to other operating systems

Getting involved in OpenSolaris

OpenSolaris development process

just the core operating system — it includes source for installers, desktops, layered software such as Open High Availability Cluster, documentation, test frameworks and test suites, and much more. OpenSolaris is millions of lines of source code in tens of thousands of source files.

You can browse the source code online at <http://src.opensolaris.org>.

If you're familiar with the Linux world, you can think of this aspect of OpenSolaris as similar to `kernel.org`, but with source code for much more than just the operating system kernel.

NOTE

Some parts of Solaris are legally encumbered, such that they cannot be open sourced. Thus, OpenSolaris does not contain the source code for the complete Solaris operating system.

OpenSolaris distributions

Unless you're an operating systems developer, source code doesn't do you much good. Most people want a running operating system, not a bunch of code. While you can theoretically build a running system from the source, if all you want to do is run OpenSolaris, it's much easier to install one of the OpenSolaris binary distributions. Luckily, there are several of them, including Solaris Express, Shillix, BeleniX, NexentaCore, and MartUX. This book focuses on the OpenSolaris distribution from Sun Microsystems, confusingly also named OpenSolaris. (The OpenSolaris distribution from Sun is available from <http://opensolaris.com>.)

NOTE

Sun Microsystems owns the trademark for the term OpenSolaris. Thus, distributions from outside Sun are allowed to use the term only by following the OpenSolaris Trademark Policy. See <http://opensolaris.org/os/trademark>.

The various OpenSolaris distributions are comparable to the various Linux distributions such as Ubuntu, Red Hat, and SUSE Linux.

CROSS-REF

Chapter 2 describes OpenSolaris distributions in more detail.

OpenSolaris community

The OpenSolaris community consists of the activity around the OpenSolaris source code and distributions, including design and development of new features, bug fixes, advocacy and evangelism, distribution building, discussions, and much more. The development community, centered at <http://opensolaris.org>, hosts the source code and provides resources for projects such as web space, mailing lists, and source code repositories. This community supports active development, similar to the Apache community.

A more user-centered community built around the OpenSolaris binary distribution from Sun can be found at <http://opensolaris.com>.

Both of these OpenSolaris communities are sponsored by Sun Microsystems. However, non-Sun employees are encouraged to participate at all levels, from using the distributions to writing kernel code.

The section “Getting Involved in OpenSolaris” near the end of this chapter provides more information about the OpenSolaris communities.

OpenSolaris Features

OpenSolaris contains a rich feature-set that makes it suitable for a wide variety of uses, from running a personal desktop or laptop to providing web services to hosting enterprise-class databases with stringent availability requirements. OpenSolaris contains far too many features to list here, but an overview of the key differentiators can help you start to evaluate its usefulness. For more details on these features and many others, read the rest of this book! Here are some of the OpenSolaris highlights:

- Support for multiple hardware architectures, including both SPARC and 32-bit and 64-bit x86-based systems. OpenSolaris also performs well in many industry benchmarks.
- High scalability. OpenSolaris runs on both single processor machines and multiprocessor systems with hundreds of CPUs and terabytes of RAM.
- Innovative file system and volume manager support. Solaris uses a Virtual File System (VFS) layer so that different file systems can be plugged in on top of it relatively easily. In addition to the standard Unix File System (UFS), OpenSolaris includes the Solaris Volume Manager (SVM) and the new ZFS.
- Networking features, including a kernel-level, high-performance TCP/IP stack, IPv6 support, IPsec, Network Auto-Magic (NWAM) for automatic detection and configuration of network interfaces, and IP Network Multipathing (IPMP) for fault tolerance and load balancing.
- Complex resource management, including processor pools, physical memory controls, and a fair share scheduler.
- Sophisticated security, including role-based access control (RBAC), configurable privileges, and trusted extensions.
- Rich observability and debugging support, including myriad system monitoring tools, the Modular Debugger (MDB), and the dynamic tracing facility (DTrace).
- Predictive self-healing features in the form of the Fault Management Architecture (FMA) and Service Management Facility (SMF). They work together to detect hardware and software faults and take appropriate action.
- Multiple forms of virtualization. In addition to the operating-system-level virtualization of Solaris Zones, OpenSolaris offers support for xVM Hypervisor, Logical Domains (LDoms), and VirtualBox, and runs in VMware and other virtualization frameworks.
- Sophisticated 64-bit fully preemptable kernel. The OpenSolaris kernel is also modular — device drivers can be installed without requiring a system reboot, and features can be configured without recompiling the kernel. The virtual memory subsystem uses demand paging for greater performance and less memory usage. The process scheduling

system supports multiple scheduling classes, including timeshare, real time, interactive, fair share, and fixed priority.

- Full POSIX compliance with a rich application programming API, including support for 64-bit applications.
- Integrated AMP stack (Apache, MySQL, PHP) for running web services.

With all of these features in mind, let's take a look at open source software.

The “Open” in OpenSolaris

As implied by the “open” in the name, OpenSolaris is open source software. The general meaning of open source is that the source code is available for anyone to look at. However, the details vary, and in fact OpenSolaris is not open source in exactly the same way as Linux, Apache, MySQL, BSD, Perl, Java, or most other open source software with which you might be familiar. To understand the details of the OpenSolaris open source model, it's helpful to first review and define some open source software basics.

Open source software basics

In the traditional closed source model of software development, companies or developers distribute only running programs in the form of binaries. Users cannot look at the source code from which those binaries were compiled. In the open source model, as its name implies, anyone can view, modify, compile, and even redistribute the source code for the programs. More specifically, the Open Source Initiative, a respected authority and advocate for open source software, specifies 10 criteria that software must fulfill in order to be open source, including the following:

- **Free redistribution** — Anyone can sell or give away the software by itself or as part of an aggregate distribution.
- **Source code** — Source must be available for all distributions.
- **Derived works** — Anyone can modify the code and redistribute it.
- **No discrimination** — The code must be available to anyone for any “field of endeavor.”

The complete list is available at <http://opensource.org/docs/osd>. It's important to remember that open source software, despite sometimes being called free software, is not required to be free of charge. Think of the “free” in free software as referring to free speech, rather than free beer. Thus, companies or individuals can sell programs built from open source code.

NOTE

Other terms for open source software include *free software* and *free and open source software* (FOSS).

This book uses the term open source software not to emphasize any particular software philosophy but because the authors think it's the clearest term.

Open source licenses

All open source code is available under an open source license, which defines the terms of use. Different open source projects choose different licenses. Some licenses with which you might be familiar are the GNU General Public License version 2 (GPLv2), under which Linux is available, and the BSD license, under which OpenBSD, NetBSD, and other BSD variants are available.

The major difference between the licenses is their requirements regarding *derivative works*, or modifications to the source code. Specifically, if someone who is not the original author takes some open source code and changes it by adding newly written code, removing code, or combining it with other code, is she required to release the new code under the original license, or can she use a different license? Based on this criterion, there are three categories of open source licenses:

- **Strong copyleft licenses** require that any derived code stay under the original license. Therefore, if a developer adds some code to a file under a copyleft license, then that new file must also be released under the original license. A strong copyleft license is project-based, rather than file-based. That is, all source files in a project must be under the same license. This requirement generally means that code under a strong copyleft license cannot link (either statically or dynamically) with code under a non-strong copyleft license. Another way of looking at it is that every piece of code that strong copyleft licensed code touches must also be under that license. For this reason, strong copyleft licenses are sometimes called *viral licenses*. Thus, you cannot generally combine code under a strong copyleft license with code under other licenses. The best-known strong copyleft license is the GNU General Public License (GPL), both versions 2 and 3. The Linux kernel, the GNU tools, Java, and a multitude of other software projects use the GPL.
- **Weak copyleft licenses** are nearly identical to strong copyleft licenses except that they're file-based instead of project-based. That is, modifications to a file must be released under the original license, but that file can be combined in a project with code under a different license. As a result, weak copyleft licenses are not viral in the same way as strong copyleft licenses. The Mozilla Public License (MPL), under which the Mozilla Firefox browser is licensed, is a weak copyleft license.
- **Non-copyleft licenses** do not require derived works to stay under the original license. They do not even require derived code to be released under any open source license. Thus, someone could take an open source project under a non-copyleft license and use it as a basis for a proprietary product. The BSD license is the best-known example of a non-copyleft license.

While these differences can seem esoteric, a quick glance at the discussions in various open source communities shows that the debate can become quite passionate. Some people, particularly in the Linux and GNU communities, feel strongly that only strong copyleft licenses are “true” open source licenses because they best protect the original author. Others feel that non-copyleft licenses, such as the BSD license, are preferable because they give the most freedom to developers creating derived works. Still others find weak copyleft licenses to be a reasonable compromise.

Open source licenses generally include provisions to distribute binary executables built from the source. The licenses usually require that the source code be made available with the binary executables, or be made available upon request. For example, both the Linux source and the Linux distributions are available under the GPLv2.

OpenSolaris licenses

The OpenSolaris source code is heterogeneous in its open source licenses and the predominant license may be unfamiliar to you.

Common Development and Distribution License

The majority of the OpenSolaris source code is available under the Common Development and Distribution License (CDDL), pronounced “cuddle.” Written by Sun explicitly for OpenSolaris, this license has been officially approved by the Open Source Initiative (OSI) as a legitimate open source license. It’s a weak copyleft license like the MPL (which it resembles) in that it requires derivative works to maintain the same license on a per-file basis, but does not require all the files in a project to be under the CDDL.

Because the CDDL is copyleft, changes to the source code itself must be released under the CDDL as well. It is hoped that any such changes are contributed back to the OpenSolaris community itself, but that is not a requirement. However, because the CDDL is weak copyleft, instead of strong copyleft, whole pieces of it may be incorporated into projects under different licenses, including proprietary projects. This aspect of the license has allowed OpenSolaris features such as DTrace and ZFS to be ported to other operating systems such as Mac OS X 10.5. (DTrace and ZFS are covered in Chapters 15 and 8, respectively.)

NOTE

The GPLv2 is incompatible with the CDDL because the GPLv2 requires all code in the project to be under the GPL. Thus, porting OpenSolaris features to Linux is significantly more complicated than porting to other systems.

Because the Solaris code base contained some open source and third-party code even before it was open sourced by Sun, not all the OpenSolaris code is under the CDDL. Parts of it are licensed under the BSD license and other open source licenses. Each source file contains a header comment specifying the license for that file.

Binary distributions under the CDDL

It’s sometimes perplexing that some binary distributions of OpenSolaris are available under the CDDL. Isn’t the CDDL a source code license? Yes, it is. However, like many other open source licenses, the CDDL permits binary executables built from source code under the license to be distributed under the CDDL. Thus, distributions of OpenSolaris may be distributed under the CDDL.

As if things weren’t confusing enough with the source code licenses, OpenSolaris also uses another binary license called the OpenSolaris Binary License (OBL). Binaries under this license are freely redistributable, and can be used for running and developing OpenSolaris. Binaries

released under the OBL include build tools, parts of Solaris that cannot be open sourced (and so aren't under the CDDL), and binaries built from proprietary code.

Open development

Open source software is generally, but not always, developed as part of a community in an open development process. In open development, developers can collaborate in public forums, participants need not all work for the same company, and there is freedom to pursue projects that might not fit within the scope of a single company's business needs.

The opposite of an open development process is a proprietary development process, in which a company or individuals write the code on their own, with their particular business needs in mind, and without interacting with people outside their group.

NOTE

Eric S. Raymond's seminal article, "The Cathedral and the Bazaar," compares these two software development models. You can read the article at www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar.

Although the Solaris operating system was originally developed in a proprietary development model, the OpenSolaris community is intended to support an open development model. Consequently, you will find many active developers, discussions, and ongoing projects at www.opensolaris.org.

What open source OpenSolaris means to you

At this point you might be wondering what the open source and open development aspects of OpenSolaris mean for you. On the open source side, while the specific terms of the licenses and the legal requirements can be complicated, the important thing to remember is that you can always look at the OpenSolaris source code. That may not be too useful if you only want to run an OpenSolaris distribution, but if you're a developer or advanced system administrator, studying the OpenSolaris code can be a valuable proposition.

On the other hand, the open development aspects of OpenSolaris should interest everyone. The OpenSolaris community is a great place to ask for help, contribute suggestions, participate in discussions, and in general influence the direction of OpenSolaris!

The History of OpenSolaris

The history of OpenSolaris, and even some of the source code, dates back to 1969. In that year, Ken Thompson at AT&T Bell Laboratories wrote the first version of the UNIX operating system. UNIX was designed from the beginning to be multi-user and multi-tasking, with an interactive shell that would still look familiar to any UNIX or Linux user today. Over the next few years, Thompson and Dennis Ritchie continued refining UNIX, which was used mostly inside Bell Labs. However, in the mid to late 1970s, UNIX versions 6 and 7 were distributed fairly

widely, and used by various academic and government institutions, including the University of California at Berkeley.

Because of the lenient license terms in early versions of AT&T's UNIX, other organizations began significantly customizing and enhancing it. This work led to several major branches of UNIX, the most relevant of which to OpenSolaris was the Berkeley Software Distribution (BSD). In 1978, Bill Joy and others at Berkeley added virtual memory, demand paging, and other embellishments to UNIX Version 7 to create a version of UNIX called 3BSD. Joy and others continued enhancing BSD UNIX over the next few years, adding the familiar TCP/IP networking stack, the C shell, the VI editor, and other key features.

In 1982, Bill Joy co-founded Sun Microsystems and by 1984 had used BSD UNIX as the basis for the SunOS operating system that ran Sun's workstations. In the meantime, AT&T continued developing its line of UNIX, calling it System V, and other companies created their own branches, such as Microsoft's Xenix (which later became SCO UNIX).

In the late 1980s, Sun and AT&T began work on a joint project to remerge several popular variants of UNIX to create System V Release 4. The result, completed in 1990, contained the best features from AT&T's earlier System V Release 3, Sun's SunOS, 4.3BSD, and Xenix 5, including TCP/IP support, the Network File System (NFS), the Unix File System (UFS), and the Virtual File System (VFS) interface. Additionally, System V Release 4 (SVR4) fully complied with the Portable Operating System Interface (POSIX) standard, which defines an application programming interface, utilities, and other aspects of an operating system. Theoretically, a program written to POSIX interfaces can run on any POSIX-compliant operating system. In 1992, SVR4 became the basis of Sun's new operating system, Solaris 2.0.

NOTE

"Solaris" technically refers to the entire operating environment, including the graphical user interface. The kernel itself is still called SunOS. However, this book uses Solaris in the colloquial sense to refer to both the entire operating environment and the kernel.

In the years since, Sun has continually enhanced Solaris with features such as the kernel slab memory allocator, multithreaded kernel and multithreaded process support, 64-bit kernel and process support, Solaris doors inter-process communication, and many others. The most recent release of Solaris, Solaris 10, introduced several exciting new features such as a dynamic tracing facility (DTrace), the Service Management Facility (SMF), Zones, and the ZFS file system.

In 2005, Solaris became the first mature proprietary operating system to go open source when Sun released the source code as OpenSolaris. The open sourced code was basically the source for Solaris 10, which had been first released approximately five months earlier. Since then, some of the active development in OpenSolaris has been backported and released in Solaris 10 updates, but much of it is currently unique to OpenSolaris. It's important to note that backports of OpenSolaris features to Solaris 10 can only be done by Sun because the Solaris 10 source code is not open source.

In summary, OpenSolaris's development path has not been exactly straightforward. As an open source operating system based on a closed-source operating system that in turn is related to several other open source and closed source operating systems, OpenSolaris can be confusing. If

nothing else, this history should help you understand why there are so many AT&T and University of California Berkeley copyrights in the OpenSolaris source code.

Comparing OpenSolaris to Other Operating Systems

So how does OpenSolaris compare to other open source and proprietary operating systems? Let's take a look.

OpenSolaris and Solaris

OpenSolaris is an open source code base, community, and distribution. Solaris is a proprietary product from Sun Microsystems. The two are not synonymous, but they are intertwined. First of all, OpenSolaris was seeded from the Solaris code base around the time of Solaris 10. However, the OpenSolaris code base has subsequently diverged from the Solaris 10 code base, so the latest update of Solaris 10 is significantly different from OpenSolaris.

NOTE

Solaris is a product from Sun, whereas OpenSolaris is an open source code base, community, and distribution.

Confusingly, Sun does release a distribution of OpenSolaris called OpenSolaris. This distribution is not the same as the Solaris 10 product. For one thing, unlike Solaris 10, the OpenSolaris distribution is available free of charge and is fully redistributable. This book focuses primarily on the OpenSolaris distribution.

CROSS-REF

The OpenSolaris distribution and other distributions are described in Chapter 2.

In the future, Sun will likely release a version of the Solaris product with long-term support that is based on a more recent OpenSolaris snapshot. This model will be similar to the way Red Hat Enterprise Linux is based on the open source Fedora code base.

OpenSolaris and Linux

Linux and OpenSolaris are both open source UNIX-like operating systems. They can support identical user interfaces, such as GNOME, run many identical applications, such as Apache, MySQL, Mozilla Firefox, and OpenOffice, and support identical tools such as the GNU compiler tools, Java, Perl, Python, Ruby, and others. But the two operating systems have significant differences in their histories, licensing, distribution models, and underlying implementations.

History

Although UNIX-like, the Linux source code is not descended from the original AT&T or BSD UNIX code. Linus Torvalds and others created it independently in the early 1990s. Because

Linux is not based on the original AT&T UNIX, BSD, SVR4, or any other form of UNIX, it does not have any kernel code in common with OpenSolaris. Linux was open source from the beginning, and was developed following a community development model.

Conversely, OpenSolaris was open sourced in whole based on the mature Solaris operating system, which was developed in large part in a proprietary development model. Partly because of this history, Linux has a much larger development community than does OpenSolaris. Linux also has many more distributions, from several different vendors.

Licensing

The Linux kernel uses the GNU General Public License version 2 (GPLv2), which is incompatible with the CDDL used by OpenSolaris because of the GPL's viral nature, as described earlier in this chapter. Thus, code cannot be ported between the OpenSolaris kernel and the Linux kernel. However, both OpenSolaris and Linux can run userland programs distributed under the GPL and other licenses, which is why they can appear to be quite similar.

Distributions

The Linux kernel, user applications, tools, and libraries are developed separately and then packaged together into distributions, which some in the free software community refer to as GNU/Linux because of the combination of the GNU tools and the Linux kernel. Some of the well-known distributions include Ubuntu, Red Hat Enterprise Linux, SUSE Linux, and Debian GNU/Linux.

OpenSolaris is more of a monolithic model, in which many of the userland tools, libraries, and applications are part of OpenSolaris itself. However, OpenSolaris distributions also use a significant amount of third-party open source software such as GNOME, Firefox, OpenOffice, and more.

Technical differences

Some of the most apparent differences between Linux and OpenSolaris derive from the fact that OpenSolaris is a variant of UNIX System V Release 4, while Linux is not. One of the most noticeable results of Linux not being based on SVR4 is that it doesn't use SVR4 packaging. Linux packaging varies between distributions, but lately it has tended toward a model whereby packages can be easily downloaded and installed dynamically from a network package repository. Interestingly, OpenSolaris has recently introduced a more Linux-like packaging approach called the Image Packaging System (IPS).

NOTE

OpenSolaris contains many of the same GNU tools found on Linux. Historically, these were in `/usr/gnu/bin/` and `/usr/sfw/bin/`, but are moving to `/usr/bin/` when possible. Because of conflicts, some commands are still in `/usr/gnu/bin/` and `/usr/sfw/bin/`.

Additionally, the Linux and OpenSolaris kernels differ significantly in the areas of scheduling, virtual memory, file systems, and others. For more details, consult one of the references listed in the "Resources" section at the end of this chapter.

OpenSolaris and BSD

Because OpenSolaris is based on Solaris, which is based in part on BSD, OpenSolaris and BSD have significant similarities in their code. The main differences are threefold. First, BSD has been developed in the open, like Linux, and so has diverged substantially from Solaris. Interestingly, the BSD community has split into three main camps such that there are now three different BSD-based operating systems: OpenBSD, NetBSD, and FreeBSD.

Second, BSD is not based on SVR4. In fact, OpenSolaris is the only open source UNIX System V Release 4–based operating system.

NOTE

Historically, Solaris contains the SVR4-style tools in `/usr/bin`, while its BSD-style tools are in `/usr/ucb/`. However, in OpenSolaris the BSD-style tools are moving to `/usr/bin` when possible. Because of conflicts, a few tools remain in `/usr/ucb/`.

Finally, unlike the CDDL used for OpenSolaris and the GPL used for Linux, the BSD license is a non-copyleft license. This lenient license does not force modifications and enhancements to be contributed back to the “commons.”

Interestingly, Mac OS X is a variant of UNIX based on the Mach operating system, which itself is based on BSD. However, Mac OS contains a distinctive Macintosh user interface, hiding the details of the underlying UNIX operating system from most of its users. Apple has ported some OpenSolaris features, such as DTrace and ZFS to Mac OS X, and has released the source code of the core operating system as the Darwin Open Source Project.

Getting Involved in OpenSolaris

As you read this book and use OpenSolaris, we encourage you to become involved in the OpenSolaris community. There are a number of ways to do so, from trying out a distribution to contributing code. A good starting point is <http://opensolaris.org/os/participate>.

Running OpenSolaris

The best way to get started with the OpenSolaris community is to actually try out OpenSolaris. In fact, playing with OpenSolaris simultaneously with reading this book will significantly enhance your learning experience.

NOTE

The OpenSolaris distribution from Sun, related documentation, and user help forums are available from <http://opensolaris.com>. Chapter 2 contains more information about the OpenSolaris distribution from Sun as well as other distributions available.

While using OpenSolaris, you can enhance your community involvement in two ways. First, if you encounter a problem, ask questions on the community discussion lists and forums. Several relevant discussion lists are introduced in the next section. Second, if you find a bug, report it at <http://defect.opensolaris.org> (for problems with the OpenSolaris distribution from Sun) or <http://bugs.opensolaris.org> (for other issues) so that it can be tracked and fixed. You can also request enhancements.

Participating in discussion lists

The OpenSolaris communities feature a plethora of forums and mailing lists on a variety of topics. www.opensolaris.com contains user-oriented forums on the OpenSolaris distribution from Sun, while www.opensolaris.org contains developer-oriented mailing lists. If you're just searching for a particular piece of information, you can read the list archives online. However, to begin to get a feel for the OpenSolaris communities and the day-to-day issues and questions, consider subscribing to the mailing lists to receive e-mails directly. Some useful lists include the following:

- opensolaris-help@opensolaris.org — This list is a great resource for general questions about “getting, building, and installing OpenSolaris.”
- opensolaris-announce@opensolaris.org — This is a moderated list for general community announcements. It's useful for keeping track of the major OpenSolaris happenings.
- ogb-discuss@opensolaris.org — The public mailing list for the OpenSolaris Governing Board. Although theoretically for “governance” issues, the list seems to be a catch-all for any sort of controversy in the community, and is therefore a good way to track current “hot” issues.
- advocacy-discuss@opensolaris.org — The mailing list for the Advocacy community. This is useful to understand what sort of outreach and marketing efforts are going on for OpenSolaris.

TIP

Many new community members are tempted to subscribe to opensolaris-discuss@opensolaris.org. We do not recommend that list because it's high-traffic without much useful content.

Additionally, you can subscribe to more focused lists in your areas of interest. For example, if you are interested in Sun's OpenSolaris distribution, subscribe to indiana-discuss@opensolaris.org. If you are interested in high-availability clusters, subscribe to ha-clusters-discuss@opensolaris.org. If you are interested in DTrace, subscribe to dtrace-discuss@opensolaris.org.

CAUTION

You may have heard the phrase, “There are no stupid questions.” That's not entirely true in the OpenSolaris community. As in many online technical communities, some people have little patience for redundant, off-topic, or trivial questions. To avoid possible embarrassment, search this book, the mailing list archives, relevant FAQs, and other resources before asking anything on the discussion lists.

Finding OpenSolaris user groups

OpenSolaris user groups connect people in similar geographic areas for face-to-face meetings, from Adelaide, Australia, to New York City to Warangal, India. They are a good way to meet other OpenSolaris users and enthusiasts, and to learn more about cutting-edge topics. If you live

in a large metropolitan area, chances are good that you can find an OpenSolaris user group in your area.

Each user group is independently run, so check out the individual group for mailing lists, upcoming meetings, and other resources. The complete list can be found at <http://opensolaris.org/os/community/advocacy/usergroups/ug-leaders>.

If there is no OpenSolaris user group in your area, consider starting one! You can find instructions for doing so at <http://opensolaris.org/os/community/advocacy/usergroups>.

Contributing to OpenSolaris

The best way to increase your involvement in the OpenSolaris community is to start participating in relevant development discussions. These usually occur on mailing lists at opensolaris.org. You can find a complete list of mailing lists at <http://opensolaris.org/os/discussions>.

If you're interested in contributing code or other tangibles to the OpenSolaris community effort, consult the instructions at <http://opensolaris.org/os/communities/participation> for the current process.

OpenSolaris Development Process

Although you may not be interested in contributing code to OpenSolaris, it can be interesting and useful to understand how the operating system is developed. Before delving into the process, it helps to understand the OpenSolaris source-code layout. The OpenSolaris code base is divided into major areas, called *consolidations*, each of which has its own source-code repository. The core OpenSolaris consolidation is Operating System/Networking (ON), which contains the operating system kernel, userland libraries, and tools. Other consolidations include Developer Product Tools (Dev Pro), Documentation (Docs), and Globalization Support (G11N). You'll find a list of consolidations at <http://opensolaris.org/os/downloads>.

NOTE

Unlike some open source projects, OpenSolaris does not have a notion of “committers” or a select group of people who are permitted to integrate code. Anyone can integrate code into OpenSolaris as long as they follow the process and have submitted a signed Sun Contributor Agreement. See http://sun.com/software/opensource/contributor_agreement.jsp for details.

OpenSolaris allows two different paths for source code development, depending on whether the code is destined for an official consolidation such as ON.

If the code is not destined for an official consolidation, then no standard development process must be followed. Anyone can start an OpenSolaris project and create a code repository or post code as a tarball on the project page.

However, code developed in that way will not become part of the OpenSolaris code base. If you want your code to become part of the OpenSolaris code base, you must follow a rigorous development process to integrate it into a consolidation. This process evolved from the internal process that Sun Microsystems required for integration into Solaris. Although it varies by repository, the process generally includes the following:

- **Initiation** — Propose a project.
- **Architecture review** — The architecture is generally reviewed by an *Architecture Review Committee* (ARC). Projects destined for ON are mostly reviewed by the Platform Software Architecture Review Committee (PSARC), projects destined for Open HA Cluster are reviewed by the Cluster Architecture Review Committee (CLARC), and projects destined for the desktop area are generally reviewed by the Layered Software Architecture Review Committee (LSARC). See the Architecture Processes and Tools Community Group at <http://opensolaris.org/os/community/arc> for more information.
- **Design** — Prepare written documentation about the code design of your project.
- **Development** — Write, test, and debug the code.
- **Code reviews** — Each area has different requirements regarding the number of code reviewers, but a good rule of thumb is to obtain reviews from at least two people, at least one of whom is a known expert in that area.
- **Integration approval** — Every project must be approved for integration by the C-Team. Currently, the C-Team is Sun-internal only, but it is moving to become more open.
- **File request to integrate (RTI)** — This is the formal mechanism for obtaining the final integration approval.
- **Integrate**

As you can see, this process is not for the fainthearted; but it's the price to pay to keep OpenSolaris at the same level of quality as the Solaris product on which it was based, and it's not too extreme compared with the process in other open source projects. For example, the Linux kernel contribution process, although different in style, is similarly rigorous.

NOTE

Various parts of the development process moved from Sun internal to OpenSolaris at different times. To allow external contributions before the source code repository offered direct-commit access from outside Sun, OpenSolaris used a request/sponsor model in which a Sun employee sponsored an external contributor.

For more information on the OpenSolaris development process, see the complete process in the ON Community Group: <http://opensolaris.org/os/community/on>.

Resources

The user-oriented site on the OpenSolaris distribution from Sun is <http://opensolaris.com>. It provides binary downloads and contains documentation and help forums. The documentation is located at <http://opensolaris.com/learn>.

The developer site for the OpenSolaris community is at <http://opensolaris.org>. It contains useful mailing lists, user group details, and a plethora of information about past and current development projects. You can start with <http://opensolaris.org/os/participate>.

The Trademark Policy can be found at <http://opensolaris.org/os/trademark>.

The OpenSolaris source code can be browsed at <http://src.opensolaris.org>.

You can file bugs at <http://defect.opensolaris.org> and <http://bugs.opensolaris.org>.

The Open Source Initiative web page (<http://opensource.org>) contains much useful information on open source code, including the text of all the licenses mentioned in this chapter.

For general information on operating systems, consult *Operating System Concepts* by Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne (Wiley, 2005).

For details on the Solaris and OpenSolaris implementation, see *Solaris Internals: Solaris 10 and OpenSolaris Kernel Architecture* by Richard McDougall and Jim Mauro (Prentice Hall, 2006).

For more information on using and administering Linux, see the *Linux Bible* by Christopher Negus (Wiley, 2005).

For details on the Linux implementation, see *Understanding the Linux Kernel (Third Edition)* by Daniel Bovet and Marco Cesati (O'Reilly, 2006).

You can find the Sun Contributor Agreement at http://sun.com/software/opensource/contributor_agreement.jsp.

The Architecture Review Process and development process are documented at <http://opensolaris.org/os/community/arc/> and <http://opensolaris.org/os/community/on>.

Summary

This chapter introduced OpenSolaris, described its three main aspects, enumerated some of the salient OpenSolaris features, explained its licensing, related some of its history, contrasted OpenSolaris with several familiar operating systems, explained the OpenSolaris development process, and described how to get involved in the community. Now you're ready to learn more about the OpenSolaris distributions in Chapter 2 and to jump into a crash course on OpenSolaris in Chapter 3.

