Mac OS X for UNIX Users

The power of UNIX with the simplicity of Macintosh.

Mac OS X version 10.4 Tiger combines a robust and open UNIX-based foundation with the richness and usability of the Mac interface, bringing UNIX technology and 64-bit power to the mass market. Apple has made open source and standards a key part of its strategy to deliver an industrial-strength operating system that is both innovative and easy to use. There are over 15 million Mac OS X users—scientists, animators, developers, system administrators, and more—making Mac OS X the most widely used UNIX-based desktop operating system. In addition, Mac OS X is the only UNIX-based environment that natively runs Microsoft Office, Adobe Photoshop, and thousands of other consumer applications—all side by side with traditional command-line, X11, and Java applications. Mac OS X is also the foundation for Mac OS X Server, which makes open source software easy to administer. Tight integration with Apple hardware, from the sleek PowerBook G4 to the award-winning Xserve G5, is making Mac OS X the platform of choice for an emerging generation of UNIX users.

Mac OS X Architecture.

Mac OS X comes with more than three dozen high-quality graphical applications for file management, Internet access, system configuration, and much more. Aqua provides the elegantly functional look and feel of Mac OS X. The entire interface— including icons, menus, windows, and controls—represents an innovative continuation of the legendary Mac ease of use, using color, transparency, and animation to enhance the usability and consistency of the system and applications. Developers can create Aqua user interfaces for Cocoa, Carbon, and Java applications, as well as with several scripting frameworks. Mac OS X includes a variety of rich application frameworks, built on top of the traditional UNIX APIs, to support developers in many different communities. Cocoa is a set of object-oriented frameworks designed for rapid application development, making it easy to add rich Aqua interfaces to existing UNIX software or to create entirely new applications. Carbon provides a gentle migration path for developers using C++ and procedural application frameworks. Java 2 Standard Edition on Mac OS X is fully compliant, highly optimized, and tightly integrated with the native look and feel, making it easy to run standards-based Java applications right out of the box. The Mac OS X graphics system combines 2D, 3D, and time-based media standards using an industry-leading compositing window system for a rich yet seamless user experience. Quartz is the high-performance imaging model in Mac OS X, based on Adobe’s cross-platform Portable Document Format (PDF) standard. Quartz uses Core Image to leverage the graphics processor for efficient display and printing of high-quality, anti-aliased text and graphics. OpenGL is the industry standard for visualizing 3D shapes and textures. Mac OS X features a tightly integrated, highly optimized, and standards-compliant implementation that uses high-end 3D graphics cards to full advantage, for basic drawing primitives as well as real-time 3D modeling and rendering. QuickTime, Apple’s cutting-edge digital media software, provides a fully standardsbased environment for creating, playing, and delivering video (MPEG-4 and H.264), audio (AAC, or Advanced Audio Coding), and images (JPEG 2000, PNG, TIFF, and hundreds more). Core Image is a new system framework in Mac OS X Tiger for high-precision image processing that can utilize modern, high-performance graphics cards to provide realtime image processing capabilities using a systemwide API for image effects and transformations. The ability to search for information—using filenames, metadata, or the contents of files—is an integral part of the Mac OS X architecture. The user interface, system applications, file system, and interprocess communication all work together to ensure that users get up-to-date information about documents, messages, applications, and other resources on the system. Beneath the easy-to-use interface and rich graphics are powerful system services for directories, mobility, and security. Together, these services ensure that Mac OS X functions consistently, compatibly, and securely wherever users go. Powering all these capabilities is Darwin, an open source, UNIX-based foundation built on technologies such as FreeBSD, Mach, Apache, and GCC. Darwin provides a complete UNIX environment, with X11 and POSIX services comparable to Linux or FreeBSD, including the familiar kernel, libraries, network services, and command-line environment described in the following pages.

State-of-the-Art Foundation.

The Mac OS X kernel at the heart of Darwin is based on FreeBSD 5 and Mach 3.0. The Berkeley Standard Distribution (BSD, first developed at the University of California, Berkeley) is one of the most widely respected UNIX implementations. BSD provides Mac OS X with the stability, performance, and compatibility for which UNIX is justly famous. Apple has enhanced BSD by adding Mach 3.0 technology based on the OSF/mk microkernel from the Open Software Foundation, providing memory management, thread control, hardware abstraction, and interprocess communication services. Apple has built on top of this rich Mach/BSD heritage with a number of powerful innovations, including well-defined, future-proof kernel programming interfaces (KPIs) supporting dynamically loadable file systems, network extensions, and packet filters, as well as I/O Kit drivers. Such innovations enable Mac OS X to provide a wide range of services, which include the following.

Mac OS X has always provided full SMP support for user applications as well as within the Mach subsystems. Tiger includes optimized kernel resource locking at the level of individual interfaces and buffers, minimizing the chance of threads on different processors having to block for each other. This allows users to get maximum performance from multiprocessor systems such as the Xserve G5 or Power Mac G5. The Mac OS X kernel directly implements the pthreads API (from the POSIX 1003.1c standard) for efficiently handling multithreaded applications on one or more processors. Each thread is individually scheduled and migrated by the kernel, without the overhead of user-level thread libraries, minimizing CPU and memory overhead. Tiger includes full support for POSIX threads, including cancellation and synchronization. Each thread, even an unprivileged one, can specify its exact real-time requirements. For example, a thread can request 30 out of every 200 cycles to ensure that the write buffer is always full for maximum-speed DVD burning. The kernel then monitors threads to ensure that they stay within their stated allotment, enabling the system to safely perform normal tasks during time-sensitive operations. Tiger fully supports both POSIX (1003.1b) and SysV semaphores for managing resource sharing between threads and tasks, including message queues and semctl. Tiger also provides POSIX, Mach, and SysV shared memory APIs, as well as the ipcs and ipcrm command-line tools. Together, these features make it much easier to port certain classes of software from Linux or Solaris to Mac OS X. The modern kqueue/kevent APIs from FreeBSD 5 provide a scalable and flexible architecture for implementing the synchronous BSD select and SysV poll APIs, as well as the asynchronous aio APIs from POSIX.4. Mac OS X supports VNODE, PROC, SIGNAL, READ, WRITE, and now MACHPORT, and FS events, eliminating the need for timeconsuming polling and making it easy for tools and services to find out about such items as changed files and network disconnects. To support numerous GPU integration features, Tiger maps video RAM (VRAM) onto main memory. This gives normal applications easy access to data such as textures, which are automatically mapped back into VRAM for use by the GPU. Mapping VRAM into system memory also simplifies the use of data sets larger than physical VRAM. Tiger features an upgraded kernel tuned specifically for 64-bit computing and the PowerPC G5 processor. The PowerPC chip is designed to run both 32-bit and 64-bit programs in the same instruction mode, maximizing both compatibility and performance. In addition, a 32-bit application can exchange data with a 64-bit process using sockets and shared memory for interprocess communication. A single binary can even contain both 32-bit ppc and 64-bit ppc64 binaries, so a command-line utility shared over NFS would run as 32-bit on a PowerBook G4 but 64-bit on an Xserve G5. Tiger processes have the option of using full 64-bit pointers for addressing virtual memory; that’s equivalent to 16 exabytes, more than enough to individually address each square centimeter of Earth’s surface. This is particularly useful for scientific computing and multimedia applications that need to access huge data sets. The standard C runtime libraries, along with other components of the system framework, have been recompiled to support 64-bit pointers. This makes it easy to build cross-platform UNIX and open source code for Mac OS X. All math and vector libraries have been tuned to take maximum advantage of the new and faster math functions supported by the 64-bit G5, as well as to use multiple processors or the Velocity Engine. In addition, portions of the Accelerate framework have been extended to accept 64-bit as well as 32-bit pointers. I/O Kit is the device driver subsystem of Mac OS X. This powerful, object-oriented architecture in embedded C++ helps device manufacturers rapidly create drivers that run safely in a multiprocessing, preemptive environment. I/O Kit is specifically designed to support the dynamic plug-and-play capabilities expected by Mac users, as well as the low latencies required by video and audio applications. The driver model provides classes implementing abstractions common to all drivers, as well as specific high-level families such as IONetworkController and IOBlockStorageDevice. This makes it easy to implement SMP/real-time-safe drivers with a minimum of device-specific code. The I/O Kit user clients allow developers to manipulate drivers from application code, which in many cases avoids the need to write kernel drivers. Key device families supported by I/O Kit include the following. FireWire, based on Apple-developed technology, is an industry standard (IEEE 1394) for connecting peripheral devices to a computer or each other. By providing a highbandwidth, easy-to-use I/O technology, FireWire inspired a new generation of portable devices such as DV camcorders, external disk drives, and digital music players like the Apple iPod. Now even faster at up to 800 megabits per second (Mbps), FireWire is also a boon to professional audio and video production. Mac OS X provides drivers that take full advantage of the hot pluggability, daisy chaining, and power management capabilities of FireWire, so most devices just work right out of the box.