

QNX Neutrino RTOS Secure Kernel

Common-Criteria certified, memory-protected microkernel architecture for maximum reliability, unparalleled scalability, and realtime performance

Since 1980, manufacturers have relied on QNX® realtime OS technology to power mission-critical applications — everything from medical instruments and 9-1-1 call centers, through battlefield communications and nuclear-monitoring systems, to Internet routers and in-car information devices. Small or large, simple or distributed, these systems share an unmatched reputation for security and reliability, operating 24 hours a day, 365 days a year, nonstop.

Leverage multi-core processing

Not only is the QNX Neutrino® RTOS Secure Kernel the only full-featured RTOS to achieve Common Criteria certification, it is the only Common Criteria-certified RTOS to implement symmetric multiprocessing (SMP) and bound multiprocessing (BMP) for multi-core processors. As a result, it is the only certified RTOS to support the parallelism necessary for successful multi-core designs.

Build secure systems

Unique QNX Neutrino adaptive partitioning technology guarantees system resources for applications while preventing rogue software from denying resources to other parts of a system. During overload conditions, adaptive partitioning enforces hard resource guarantees, ensuring applications receive their budgeted share of resources.

Create fault tolerant applications

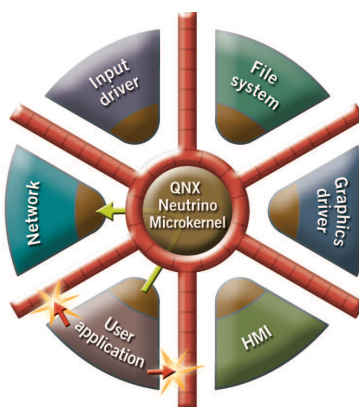
Time-tested and field-proven, the QNX Neutrino RTOS is built on a true microkernel architecture. Under this system, every driver, application, protocol stack, and file system runs outside the kernel in the safety of memory protected user space. Virtually any component can fail and be automatically restarted without affecting other components or the kernel. Further, QNX Neutrino provides an optional high-availability framework for ensuring critical software is monitored and kept running even after faults. No other commercial RTOS provides such a high level of fault containment and recovery.

Inherently modular, the QNX Neutrino lets you dynamically upgrade modules, introduce new features, or deploy bug fixes — without costly downtime or system outages.

Rely on standards for secure programming

Engineered to the POSIX standard (1003.1-2001 POSIX.1), the QNX Neutrino RTOS provides a well understood programming API with predictable and reliable behavior — a key ingredient to secure systems. This POSIX API prevents the use of proprietary interfaces with the potential for insecure behavior and misunderstood results. Moreover, POSIX demands proper adherence to user and group

Secure microkernel by design



A microkernel RTOS provides protective barriers between processes including networking and filesystems, which are typically within the kernel memory space in monolithic operating systems.

privileges and security defaults that prevent, if necessary, high privilege access to system resources. Using POSIX programming also gives developers the power to port legacy and open-source UNIX, Linux, and Internet code with just a simple recompile. With standard APIs, developers can reuse application code, avoid costly delays, and shorten their learning curve — accelerating development cycles and reducing time to market.

Scale large or small

Take advantage of the inherent multi-core capability and built-in transparent distributed processing of the QNX Neutrino RTOS to create large, highly fault-tolerant systems. Regardless of design requirements, developers can use the same RTOS, tools, APIs, and source code for all projects. The result? Longer product life cycles, shorter development time, and a significant return on investment.

Foundry27

This community portal for QNX software developers provides source repositories, forums, blogs, wikis, and, in the community spirit, a bazaar that lets developers share source code and binaries with one another. Whether the developer wants to download source, discuss ideas with fellow developers, or even create a new code project for the QNX community, Foundry27 offers the resources required.

About QNX Software Systems

QNX Software Systems, a Harman International company (NYSE: HAR), is the innovation leader in embedded technology with middleware, development tools, operating system software and services. Global leaders such as Cisco, Daimler, General Electric, Lockheed Martin, and Logitech depend on QNX technology for a wide range of OEM applications. QNX Software Systems is headquartered in Ottawa, Canada, and distributes products in over 100 countries worldwide.



QNX SOFTWARE SYSTEMS

H A Harman International Company

QNX Neutrino RTOS Secure Kernel

Common Criteria (ISO/IEC 15408) EAL4+ certification

- ▶ Stringent standard for security and development processes used in development
- ▶ First full-featured EAL4+-certified RTOS
- ▶ First RTOS to include SMP support for multi-core processors and adaptive partitioning technology in certification

Microkernel architecture

- ▶ Dynamically upgradable services and applications
- ▶ Fine-grained fault isolation and recovery
- ▶ Message-passing for modular, well-formed systems

POSIX compliance

- ▶ POSIX PSE 52 support of broadest range of POSIX API specifications
- ▶ Well understood programming API with predictable behavior
- ▶ Correct user and group privilege management and enforcement

High availability and fault tolerance

- ▶ Heartbeat for early fault detection
- ▶ Intelligent restart and transparent reconnection

Open Transparent Development

- ▶ Foundry27 community access to QNX developers, source, forums and project roadmaps
- ▶ Source code visibility and access to all key areas of the OS technology – no hidden, proprietary implementation

Adaptive partitioning

- ▶ Guaranteed system resources for secure, reliable systems without compromised performance and flexibility

Predictable realtime performance

- ▶ Preemptive scheduler with choice of scheduling methods
- ▶ Distributed priority inheritance

Multi-core

- ▶ Comprehensive multi-core support
- ▶ Asymmetric, symmetric, and bound multiprocessing
- ▶ Simple migration from uni- to multiprocessing

Transparent distributed processing

- ▶ Transparent network access to remote resources
- ▶ Simplified design of fault-tolerant clusters

File systems

- ▶ Image, RAM, Flash disk
- ▶ Linux, DOS, CD-ROM, DVD
- ▶ NFS, CIFS, compression
- ▶ Power safe mass storage file system

Resource manager framework

- ▶ Device drivers implemented in user, non-kernel, space
- ▶ Ability to start, stop, and debug drivers like any standard application

Device drivers

- ▶ Audio, character, disk, graphics, input, networking, parallel, printer, serial, USB

Processor support

- ▶ ARM, MIPS, PowerPC, SH-4, x86
- ▶ Extended memory support (beyond 4GB) for x86, MIPS, and PowerPC