QNX®/Neutrino® Realtime OS

Redefining OS technology for embedded systems

After 18 years of innovation in realtime OS microkernel development, QNX Software Systems introduces the next generation in microkernel design — QNX/Neutrino, a massively scalable, multi-threaded, fault-tolerant realtime OS for x86, PowerPC, and MIPS platforms.

Microkernel architecture at its best

The Neutrino microkernel delivers core realtime services for embedded applications, including message passing, POSIX thread services, mutexes, condition variables, semaphores, signals, and scheduling. What's more, it can be smoothly extended to support POSIX message queues, filesystems, networking, and other OS-level capabilities with simple plug-in, service-providing modules.

Neutrino's message-passing architecture forms a software bus that lets you plug in — or out — whatever OS modules you need, without rebooting your system. From tiny kernel-only embedded systems to powerful SMP boxes, it can scale any way you want.

Powerful kernel for massive scalability

Neutrino's architecture offers unprecedented scalability. Link your application code directly against the kernel to create a single multi-threaded image for small embedded systems — as you would with a realtime executive. Or run the Process Manager for all the advantages of a full process model and the ability to add thousands of applications — all running in MMU-protected memory. Or take it to the extreme and run your apps over a distributed network of SMP clusters for the ultimate in large-scale configurations! Whatever your configuration — tiny, medium, massive, or distributed — recoding is never an issue since the Neutrino API remains consistent throughout.

QNX®
The Leading Realtime OS for PCs
now multiplatform

POSIX means portable

Neutrino is the world's first microkernel with a POSIX personality. Unlike realtime executives and OS implementations that have proprietary APIs, Neutrino is engineered from the ground up for the latest POSIX 1003.1 standards and drafts, including realtime and thread options.

Neutrino's POSIX implementation means portability — not only of your application code, but also of your software developers! In fact, programmers familiar with UNIX won't need any training to feel right at home in this POSIX environment.

What's more, this built-in POSIX compatibility comes without the penalty of extra code. Even after the Process Manager is added to include services like process creation, pathname-space management, and memory protection, a Neutrino-based system is extremely small and efficient — crucial for ROMable systems. So you can count on better overall performance and reduced memory requirements with less code at the heart of your system — without sacrificing functionality or performance.

Superior memory protection

Conventional operating systems use a single flat memory architecture where hard-to-detect programming errors like corrupt C pointers can cause programs to overwrite each other (bad) or the kernel (worse). The inevitable result: system failure. A Neutrino-based system, however, can intelligently recover from software faults, even in drivers and other critical programs – without rebooting – because every OS component runs in its own MMU-protected address space.

Neutrino's full MMU support also simplifies testing since it identifies which module tried to perform an invalid memory access – at the exact instruction. What can often take weeks or months to identify in a conventional RTOS takes virtually no time with Neutrino.

Realtime executive architecture provides no protection.

Traditional monolithic architecture offers limited protection to application processes.

Neutrino's microkernel architecture provides full protection of all components.

Minimize hardware costs

Unlike some operating systems that try to squeeze monolithic designs or bulky windowing systems into embedded environments, QNX/Neutrino was designed from the ground up to reduce the cost and component count of your products. On custom x86 target systems, for example, you can eliminate the expense of a BIOS since Neutrino doesn't rely on BIOS calls. It was also designed to keep RAM requirements to an absolute minimum.

Neutrino supports execute-in-place (XIP), which allows applications to run directly out of ROM or flash. And, since its system image is actually a simple read-only filesystem, it allows applications to start without a separate filesystem manager or command interpreter.

A host of functionality and development options

QNX Software Systems has developed an efficient suite of embedding tools and runtime software components to provide everything from cross-platform connectivity to a full-featured embeddable windowing system.

Powerful development environment

Neutrino 1.1 development is supported under the award-winning CodeWarrior Integrated Development Environment for efficient multiplatform development. (Command-line GNU-compatible development tools are also available.)

Dynamically loadable functionality beyond DLLs

No other realtime OS scales so easily — just plug in the modules or drivers you need. Like other operating systems, QNX/Neutrino supports shared objects (also known as DLLs). But unlike other operating systems, Neutrino lets you add or remove entirely new OS functionality (via software modules) on the fly — safely and seamlessly — without rebooting your system.

Extend the functionality of your application with any of the following modules:

- **Embeddable Photon microGUI®**

Offers a highly functional windowing system for resource-constrained embedded environments. Running in an extremely small memory footprint, Photon delivers sophisticated functionality and connects seamlessly to Neutrino's message-passing architecture.

Photon also gives you exceptional connectivity between windowing systems. With Photon's remote user interface (RUI) technology you can view — and control — the GUI of a Neutrino embedded system from a window on a Windows NT/95/98 or UNIX desktop. RUIs are baud-rate aware and can run across a serial or network link. For embedded systems, this can give you a graphical interface into your consoleless black box.

With the optional ICA3 client for Photon, reverse connectivity is also available for situations where you need to run a Windows NT/95/98 session within a window on the Photon desktop.

- **Boot Modules**

Perform platform- and processor-dependent initialization for embedded systems at startup, releasing occupied memory after initialization. (Boot modules are supplied in full source for custom hardware adaptations.)

- **Process Manager**

Extends Neutrino's services to include support for processes (containing threads), protected memory, and pathname-space management. The pathname space can then be populated by other processes that are visible to application threads.

- **Embeddable QNX Filesystem Manager**

Provides essential services (including hard and soft symbolic links, long filenames, etc.) of a POSIX 1003.1 filesystem in a low-overhead implementation.

- **Flash Filesystem Manager**

Features wear-levelling, on-the-fly decompression, random writes, fault recovery, and other characteristics unique to implementing a filesystem in flash memory. In keeping with Neutrino itself, this filesystem is multithreaded (e.g. background reclaims).

- **CD-ROM Filesystem Manager**

Implements the ISO 9660/Rock Ridge media standard and allows CD-ROMs to be readily used.

- **CIFS Filesystem Manager**

Implements the Microsoft Common Internet File System standard and allows Windows NT/95/98 network file access.

- **Tiny TCP/IP Manager**

Provides a small implementation of TCP/IP; includes ftp, ftpd, telnet, telnetd, and more. Also supports PPP and 802.3 networking.

- **Serial Manager**

Provides standard POSIX TTY device control while supporting extensions for efficient realtime protocol management.

Device drivers made easy across platforms

From the beginning, drivers for Neutrino were designed to be source-code identical across CPUs and boards. In fact, even the same binaries for a CPU can run on different boards — no more BSP nightmares sorting out the complexity of custom board- and processor-specific code.

To reduce the time required to write your own device drivers, Neutrino provides a resource manager framework and C functions that handle the default behaviors common to most devices; all you need worry about are the low-level details specific to your device.

And because each Neutrino driver runs as a standard process (rather than as part of the kernel itself), you can test changes in driver code without having to go through the time-consuming task of rebuilding the kernel. Simply recompile and restart the driver on a running system.

Pick the best CPU for the job

With Neutrino, you can choose from a number of leading processors, including many highly integrated chips that save on space and cost:

- x86 — 386, i386 EX, Am386SE/DE, AMD ÉlanSC400/410, 486, Cyrix MediaGX, Pentium, Pentium Pro, Pentium II
- PowerPC — 401, 403, 603e, MPC860, MPC821, MPC823
- MIPS — R4000, R5000, NEC VR4300/4102/4111

Neutrino also supports many buses, including ISA, PCI, VME, STD, STD 32, and PC/104.

Please contact us for an up-to-date list of dynamically loadable software modules for MIPS, PowerPC, and x86 platforms.

Contact Information

Voice
Head Office ..... +1 613 591-0931
North America 1 800 676-0566
United Kingdom ..... +44 (0) 1223 364503
France ..... +33 1 646 15911
Germany ..... +49 (0) 511 94091-0
Fax ..... +1 613 591-3579

Email ..... info@qnx.com
Telnet ..... quics.qnx.com
Anonymous ftp ..... ftp.qnx.com
Dial up (ANSI, VT100) +1 613 591-0934

www.qnx.com

The Leading Realtime OS for PCs
now multiplatform

© QNX Software Systems Ltd. 1998
QNX, Neutrino, and Photon microGUI are registered trademarks, and PhAB, Phindows, Phinx, and Voyager are trademarks, of QNX Software Systems Ltd.
All other trademarks belong to their respective owners.
Printed in Canada. Part number: 301193