RTX 4.1™

Benefits of using RTX with Windows NT

- **Reduce System Complexity** by relying on a single operating system. RTX is an extension to Windows NT that provides robust deterministic performance capabilities, not a separate operating system.

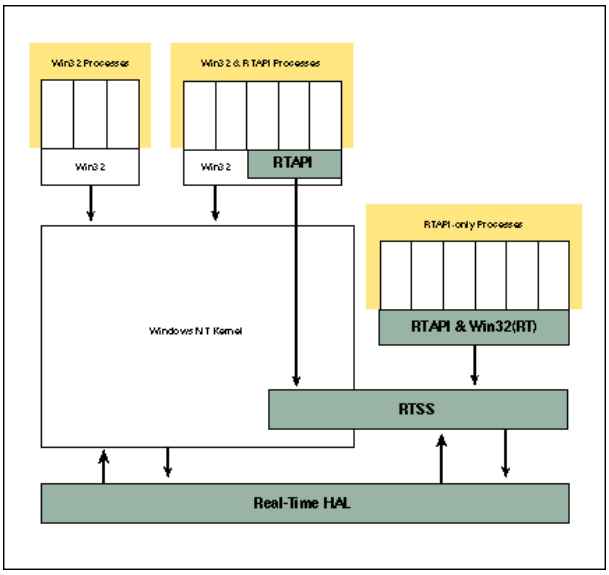
- **Reduce Development Costs** by developing RTX applications that utilize standard off-the-shelf Windows NT development tools, thus reducing development and training costs.

- **Reduce Time to Market** by leveraging the abundance of available Windows NT services, applications, and tools while providing the foundation for implementing RTX applications on a single platform.

- **Reduce Hardware Costs** by implementing real-time and Windows NT applications tightly coupled on a single hardware platform.

- **Ensure Application Compatibility** by utilizing the RTAPI and Win32(RT) — common to both Windows NT and RTX environments — which guarantees the portability of code between real-time and non real-time applications.

- **Ensure Forward Compatibility** of applications written to the RTAPI and Win32(RT). RTX is implemented with no changes to the Windows NT kernel, executive, or existing device drivers and is therefore portable to new versions of Windows NT.



Real-Time Extension (RTX) 4.1

VenturCom's RTX 4.1 is an extension to the Windows NT operating system that enables deterministic response characteristics for high-speed control applications. RTX 4.1 is tightly integrated with the Windows NT kernel and utilizes Windows NT services and the Win32 API to provide RTX services and functions. RTX 4.1 is comprised of several layers.

Real-Time HAL

The real-time HAL (Hardware Abstraction Layer) lays the foundation for RTX 4.1. The design of the real-time HAL originated with VenturCom and provides the underlying resources for the fast clock and timer services and the interrupt isolation between RTX and Windows NT that makes the high-speed and deterministic response characteristics of RTX possible. Additionally, the real-time HAL intercepts any Windows NT "blue screen" events and allows RTSS to continue until an orderly recovery of the Windows NT portion of the application can be managed.

Real-Time Subsystem (RTSS)

The RTSS provides the core functions and resource management of RTX. Implemented as a Windows NT device driver, the RTSS is an extension to Windows NT that utilizes the services of Windows NT and the real-time HAL to provide the services necessary for a real-time environment sub-system. RTSS utilizes the real-time HAL to provide the clock and timer services and interrupt management facilities to RTX processes. The RTSS utilizes the services available through Windows NT to provide the memory management facilities of RTX. The RTSS contains a thread manager and a fixed priority thread scheduler to provide the process and thread management facilities. The RTSS also provides the interface between RTX and Windows NT processes through the real-time inter-process communication (IPC) services.

Real-Time Application Programming Interface (RTAPI)

The RTAPI provides an essential set of programming extensions to Win32. The RTAPI, combined with Win32(RT) — a sub-set of Win32 functions — provides a complete and fully functional real-time application programming environment. The RTAPI may be utilized by both RTX and Windows NT processes. Since the RTAPI operates with both the Windows NT and RTX run-time environments, applications can be developed that are portable between the two environments. Since the RTAPI operates on Windows NT, the user's choice of Windows NT development tools can be utilized to develop applications for both environments.

Applications can take advantage of both the Windows NT and RTX run-time environments simultaneously. Applications can be partitioned into time-critical functions that run on the Real-Time Sub-System and non time-critical functions that run on Windows

NT. RTX and Windows NT processes communicate and share information through the Real-Time Sub-System using Inter-Process Communication functions.

RTAPI 2.0 Features

Process and Thread Management

The RTAPI provides interfaces for creating, prioritizing, controlling, profiling, and terminating RTX threads.

Fixed Priority Scheduling

RTX scheduler provides 128 fixed, non-degrading priorities that guarantee the execution order of RTX processes. RTX provides prioritizing, inversion management and preemption capabilities to give exclusive use of the CPU to the highest-priority real-time process.

Memory Management

The RTAPI memory management facilities allow a process to allocate both contiguous and locked memory and to lock both process and Windows NT kernel memory. All of the RTSS and RTX processes are automatically locked in memory. However, portions of the Windows NT kernel and all of Windows NT processes are normally not locked in memory, but are paged in from disk as required. The memory locking functions will eliminate paging latencies for Windows NT processes.

Inter-Process Communication

The RTAPI provides **semaphores, mailslots,** and **shared memory** inter-process communication mechanisms to allow RTX and Windows NT processes to communicate with each other and share information.

High Speed Clocks and Timers

RTX provides clocks with 1 microsecond resolution and timers with periods of 100 microsecond increments. These features provide the capability to measure time accurately and to schedule tasks at precise times and intervals.

Port I/O

The RTAPI port input and output programming interfaces allow data movement in the I/O space of a processor by a user-level process. This eliminates the need for creating a device driver for every device that must be accessed and eliminates the latencies associated with requesting driver services for device access.

Physical Memory Mapping

The RTAPI physical memory mapping functions allow a process to access physical memory with a virtual address. Physical memory mapping gives a user-process access to physical memory address ranges, typically to memory on controllers and other devices.

Interrupt Management

The RTAPI memory management functions allow a user process to attach a function handler to an interrupt and to enable and disable interrupt processing. This ability, along with the port I/O and physical memory mapping functions, allows for a device to be controlled entirely from a user process, thus eliminating the need to write a device driver.

VenturCom, Inc.
617/ 661-1230 tel
617/ 577-1607 fax
info@vci.com
http://www.vci.com