Martin Timmerman and Bart Van Beneden,
Director and Computer Engineer,
Real-Time Consult.

VxWorks/x86 5.3.1 evaluation

Following article is an executive summary of the evaluation report of VxWorks/x86 5.3.1 from WindRiver Systems, Inc.

INTRODUCTION

During the summer of 1998, Real-Time Consult officially started an RTOS evaluation program. First, Windows NT and the real-time extensions to Windows NT were studied. The evaluation reports for the following products are currently available:

- RTX 4.2 from VenturCom, Inc.
- INtime 1.20 from Radisys Corporation Ltd.
- Hyperkernel 4.3 from Imagination Systems, Inc.
- VxWorks/x86 5.3.1 from WindRiver Systems Inc.
- pSOSystem/x86 2.2.6 from Integrated Systems Inc.
- QNX 4.25 from QNX Software Systems Ltd.

The evaluation reports, as well as comparison reports highlighting the decision critical information are available on our website (http://shop.realinter.net/rtshop/).

This article presents an executive summary of the evaluation report of VxWorks 5.3.1 from WindRiver Systems, Inc.

ARCHITECTURE

VxWorks was initially a development and network environment for VRTX and pSOSystem. Only later on did Wind River Systems develop their own microkernel. The net results is that VxWorks was "client-server" from the beginning.

At the heart of the VxWorks run-time system is the wind microkernel. This microkernel supports a full range of real-time features including multi-tasking, scheduling, intertask synchronization/communication and memory management. All the other functionality is implemented as processes.

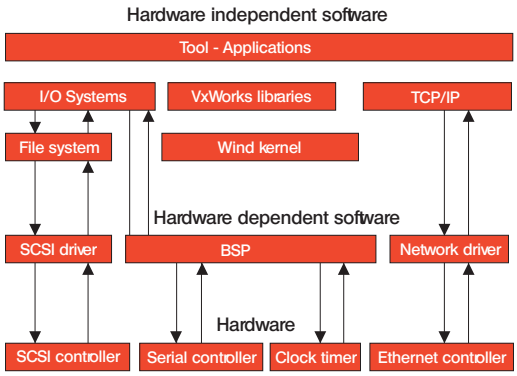

Figure 1. VxWorks system architecture

VxWorks 5.3.1 is very scalable. By including or excluding various modules, VxWorks can be configured for the use in small embedded system with tough memory constraints to complex systems where more functions are needed. Furthermore, individual modules themselves are scalable. Individual functions may be removed from the library, or specific kernel synchronization objects may be omitted if they are not required by the application.

Tasks

The VxWorks 5.3.1 real-time kernel (wind) provides a basic multitasking environment. VxWorks 5.3.1 offers both POSIX and a proprietary scheduling mechanism (wind scheduling). Both a preemptive priority and a

VxWorks 5.3.1	
Model	Tasks containing single thread of execution. All tasks run in a single address space without any protection. VxVMI (optional component) allows each task to have its own address space.
Priority levels	256 levels.
Max. number of tasks	Limited by the amount of memory available
Scheduling policies	Both POSIX and Wind scheduling - Preemptive priority. - Round-robin.
Number of documented states	9

Table 1. VxWorks 5.3.1 Task handling properties

Copyright 1999 by Real-Time Magazine 99-4 (http://www.realtime-info.com)

round-robin scheduling mechanism are available.

The differences between POSIX and wind scheduling is that wind scheduling applies the scheduling algorithm on a system wide basis, whereas POSIX scheduling algorithms are applied on a process-by-process basis.

Memory

In VxWorks 5.3.1, all systems and all application tasks share the same address space. This means that faulty applications could accidentally access system resources and compromise the stability of the entire system. However, WindRiver Systems does provide an additional component (VxVMI) that needs to be purchased separately and that allows every process to have its own private virtual memory.

VxWorks 5.3.1	
MMU	Not required, but supported (need extra component VxVMI).
Physical page size	8KB by default.
Paging/Swapping	Never.
Virtual memory	Supported (need extra component VxVMI).
Memory protection models	No protection: all tasks run in the same address space. User/user private VM: every user process has its own private virtual memory (need extra component VxVMI).

Table 2. VxWorks 5.3.1 Memory Management properties

VxWorks 5.3.1 also does not offer privilege protection, the privilege level is always 0 (supervisor mode).

Interrupts

To achieve the fastest possible response to external interrupts, interrupt service routines (ISRs) in VxWorks 5.3.1 run in a special context outside of any thread's context, so that there are no thread context switches involved.

It should be pointed out that the C function that the user attaches to a interrupt vector is not the actual ISR. Interrupts cannot directly vector to C functions. The ISR's address is stored in the interrupt vector table and is called directly from the hardware. The ISR performs some initial work (e.g. saving registers and setting up stack) and then calls the C function that was attached by the user. For this reason, we use the term interrupt handler (instead of ISR) to designate the user installed C handler function.

API RICHNESS

To assess the API richness, we created a list of features for the most common system calls and compared it with the available system calls in VxWorks 5.3.1. Table 4 gives an overview of all the categories and the score (in percentage points) that was obtained. For a breakdown of the categories into individual features and system calls, the reader is referred to the evaluation report.

VxWorks 5.3.1	
Handling	Nested and prioritized
Context	Interrupt handlers run in a special context, outside any task's context.
Stack	Special interrupt stack. If the architecture doesn't allow a separate stack for ISRs, then the stack of the interrupted task is used
Interrupt-to-task communication	• Shared memory and ring buffers. • Semaphores (release only). • Message queues (send only) • Pipes (write only) • Signals (send only)
Min. RAM	Less than 100 bytes.

Table 3. VxWorks 5.3.1 Interrupt handling properties

This table should not be misunderstood. VxWorks 5.3.1 has system calls that are not in our list, and are therefore not represented in Table 4.

An average percentage of 63 % was obtained. The average percentage does not include any weight factors, it is simply the average of each category's score. Table 4 shows that the VxWorks API is missing conditional variables and event flags.

VxWorks 5.3.1 has by far the richest API of the RTOSs evaluated by Real-Time Consult so far. pSOSystem/x86 2.2.6 from Integrated Systems Inc for example scored an average percentage of 44%.

PERFORMANCE TESTS

Interrupt latencies

For this test, we measured two latencies:

- Interrupt Latency (task to interrupt handler): The time elapsed between the execution of the last instruction of the interrupted thread and the first instruction in the interrupt handler.
- Interrupt Dispatch Latency (interrupt handler to task): The time needed to go from the last instruction in

Copyright 1999 by Real-Time Magazine 99-4 (http://www.realtime-info.com)

Mechanism	Richness
Thread Management	94 %
Clock	57 %
Interval Timer	100 %
Fixed block size memory partition	73 %
Non-fixed block size memory pool	82 %
Interrupt Handling	38 %
Counting Semaphore	70 %
Binary Semaphore	100 %
Mutex	92 %
Conditional Variable	0 %
Event Flags	0 %
POSIX Signals	100 %
Message Queue	81 %
Mailbox	0 %
AVERAGE PERCENTAGE	63 %

Table 4. API Richness

the interrupt handler to the next task scheduled to run.

Figure 2 and Figure 3 display the minimum, maximum and average values for both interrupt latency and interrupt dispatch latency. The results for QNX/Neutrino 1.0 from QNX Software Systems, Ltd are included for comparison.

Priority inheritance

VxWorks 5.3.1 has a priority inheritance mechanism, which is essential for an RTOS. We tested this by creating a situation with 3 threads where the priority inversion problem occurs: a high priority thread wants to acquire a mutex that is owned by a low priority thread. A medium priority thread keeps the low priority thread from running and releasing the mutex so that the high priority thread can't acquire it.

For a detailed description and flow chart of the test, the reader is referred to the document "report definition and test plan", which can be downloaded from our website (www.realtime-info.be/eval).

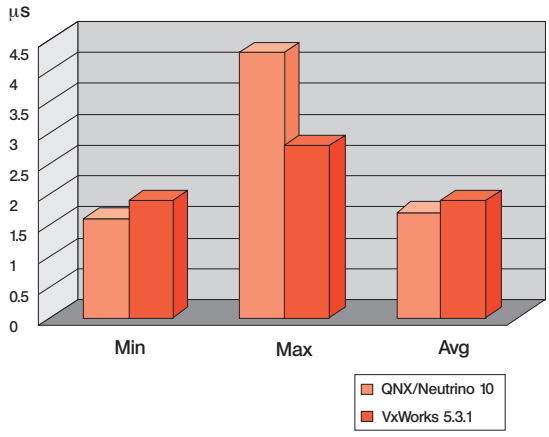

Figure 2. Interrupt Latency - VxWorks 5.3.1

In this test, we measured the time it takes for the highest priority thread to acquire the mutex. That time includes the time it takes to boost the priority of the lowest priority thread, have it release the mutex, and switch back to the highest priority thread so it can acquire the mutex. The results for the test are displayed in Figure 4.

The high maximum measurement was caused by a clock interrupt interrupting the measurement. The clock ISR in VxWorks apparently can get pretty lengthy under certain circumstances, which makes the system less responsive to other external interrupts. This is demonstrated and explained in more detail in the full evaluation report.

TOOLS & DEVELOPMENT METHOD

Wind River has an integrated development environment for embedded applications called "Tornado". Tornado is a completely open environment designed to be customized and extended by the developer. Its open interfaces make it possible to integrate other development tools from third parties.

Tornado comes with a very extensive set of tools. Aside from the traditional tools like compilers, debuggers and profilers, Tornado also provides tools to perform real-time data analysis (Stethescope), tools to detect memory leaks and code coverage tools.

A nice feature of VxWorks/Tornado is dynamic linking and loading of modules. This feature can reduce the edit-test-debug cycle. The user can download an individual object module into the target without having to reboot. The module is dynamically linked into the target. There is no need for the user to compile and link the complete executable on the host and download it to the target.

To assist developers of embedded systems using custom hardware, WindRiver Systems also offers VxSim, which is a prototyping and simulation tool for Tornado/VxWorks. VxSim provides a simulation of VxWorks on the host. With this tool, application development can begin before the hardware becomes available.

DOCUMENTATION & SUPPORT

Tornado 1.0.1 comes with an extensive set of manuals. Unfortunately, the manuals are not always clear and comprehensive. The reader may find himself browsing through various manuals before finding the information he was looking for.

The Wind River web site (http://www.wrs.com) has a large database of online publications, FAQs and more. Customers with a valid maintenance contract can get a username and password that provides access to this database.

Customer support is very professional. Support is best requested via email. Tornado support request (TSR) messages can be sent from within the Tornado environment. Before sending the email to Wind River tech-

Copyright 1999 by Real-Time Magazine 99-4 (http://www.realtime-info.com)

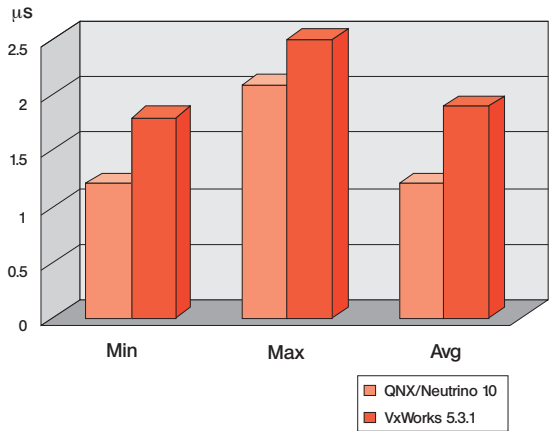

Figure 3. Interrupt Dispatch Latency - VxWorks 5.3.1

nical support, the TSR tool adds specific information about your host and target system to your message that might be helpful to the technical support representative.

CONCLUSION

VxWorks 5.3.1 has a "client-server" architecture in the sense that it has a small microkernel which only handles the basic real-time features. All the other functionality is implemented as processes. VxWorks is not a message based operating system.

During all of our tests, VxWorks/x86 5.3.1 exhibited predictable behavior. The clock ISR execution time can become pretty lengthy in certain cases, which reduces the responsiveness of the system to other external interrupts (the clock interrupt is by default the highest priority interrupt in PC-like targets, and VxWorks/x86 cannot be configured otherwise).

The Tornado development environment is good and comes with a very extensive set of tools which facilitate development. Developers can also count on a professional technical support team.

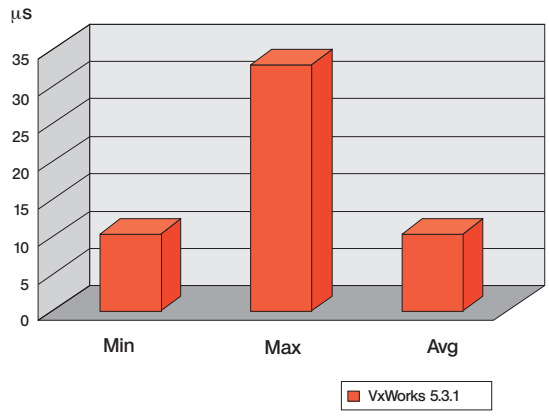

Figure 4. Priority Inheritance

OTHER PUBLICATIONS AND SERVICES

As mentioned at the outset, evaluation reports for Windows NT 4.0, RTX 4.2, Hyperkernel 4.3 and INtime 1.20 have been for sale on our website since the beginning of 1999.

Real-Time Consult has also evaluated VxWorks/x86 5.3.1, QNX 4.25 and pSOSystem/x86 2.2.6. Evaluation reports for these products have been available since April 20, 1999.

The evaluation reports are intended for everyone who is in one way or another involved with dedicated systems technology. This obviously includes the system design engineers and application developers, who need to have a detailed understanding of how the product behaves in a real-time environment, but the audience also includes managers and project leaders who need to make strategic decisions like which RTOS to use, and how it will affect the overall execution of the project.

Finally, Real-Time Consult also performs feasibility studies and product validations on customer demand. Please contact our offices for additional information ■

Dr. Martin Timmerman has a degree in Telecommunications Engineering from the Royal Military Academy (RMA) Brussels and received a Doctorate in Applied Science from the Gent State University (1982) in Belgium. In 1983 he transferred to Computer Engineering and set up the System Development Centre (SDC) at RMA. He gives general courses on Computer Platforms and more specific courses on System Development Methodologies. He is a consultant to the Joint Staff of the Belgian Armed Forces in areas concerning Information System Methodologies and CASE tools and he is the Belgian representative in some NATO technical commissions. Outside the RMA, Martin is known for his audits, reviews and seminars, and for his two companies Real-Time Consult and R.T.U.S.I., where he makes use of his considerable knowledge of the Real-Time world. Real-Time Consult is the publishing house responsible for Real-Time Magazine, an International magazine about Real-Time system development. Real-Time User's Support International (R.T.U.S.I.) provides hardware and software support services and is involved in project engineering for real-time systems.
Bart Van Beneden has been with Real-Time Consult since 1998 where he is involved in the RTOS evaluation program of Real-Time Magazine as a project manager. He received his degree in computer science at the Free University of Brussels. Before joining Real-Time Consult, he designed multi-media applications with LaserMedia Inc.

Copyright 1999 by Real-Time Magazine 99-4 (http://www.realtime-info.com)