QNX NEUTRINO 6.2, VXWORKS AE 1.1, WINDOWS CE .NET AND ELDS 1.1 COMPARED

http://www.dedicated-systems.com

E-mail: info@dedicated-systems.com

http://www.dedicated-systems.com
email: info@dedicated-systems.com

EVALUATION REPORT LICENSE

This is a legal agreement between you (the downloader of this document) and/or your company and the company DEDICATED SYSTEMS EXPERTS NV, Gerechtsstraat 21, 1070 Brussels, Belgium.
It is not possible to download this document without registering and accepting this agreement on-line.

1. **GRANT**. Subject to the provisions contained herein, Dedicated Systems Experts hereby grants you a non-exclusive license to use its accompanying proprietary evaluation report for projects where you or your company are involved as major contractor or subcontractor. You are not entitled to support or telephone assistance in connection with this license.
2. **PRODUCT**. Dedicated Systems Experts shall furnish the evaluation report to you electronically via Internet. This license does not grant you any right to any enhancement or update to the document.
3. **TITLE**. Title, ownership rights, and intellectual property rights in and to the document shall remain in Dedicated Systems Experts and/or its suppliers or evaluated product manufacturers. The copyright laws of Belgium and all international copyright treaties protect the documents.
4. **CONTENT**. Title, ownership rights, and an intellectual property right in and to the content accessed through the document is the property of the applicable content owner and may be protected by applicable copyright or other law. This License gives you no rights to such content.
5. **YOU CAN NOT**:
   - You can not, make (or allow anyone else make) copies, whether digital, printed, photographic or others, except for backup reasons. The number of copies should be limited to 2. The copies should be exact replicates of the original (in paper or electronic format) with all copyright notices and logos.
   - You can not, place (or allow anyone else place) the evaluation report on an electronic board or other form of on line service without authorisation.
6. **INDEMNIFICATION**. You agree to indemnify and hold harmless Dedicated Systems Experts against any damages or liability of any kind arising from any use of this product other than the permitted uses specified in this agreement.
7. **DISCLAIMER OF WARRANTY**. All documents published by Dedicated Systems Experts on the World Wide Web Server or by any other means are provided "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. This disclaimer of warranty constitutes an essential part of the agreement.
8. **LIMITATION OF LIABILITY**. Neither Dedicated Systems Experts nor any of its directors, employees, partners or agents shall, under any circumstances, be liable to any person for any special, incidental, indirect or consequential damages, including, without limitation, damages resulting from use of OR RELIANCE ON the INFORMATION presented, loss of profits or revenues or costs of replacement goods, even if informed in advance of the possibility of such damages.
9. **ACCURACY OF INFORMATION**. Every effort has been made to ensure the accuracy of the information presented herein. However Dedicated Systems Experts assumes no responsibility for the accuracy of the information. Product information is subject to change without notice. Changes, if any, will be incorporated in new editions of these publications. Dedicated Systems Experts may make improvements and/or changes in the products and/or the programs described in these publications at any time without notice. Mention of non-Dedicated Systems Experts products or services is for information purposes only and constitutes neither an endorsement nor a recommendation.
10. **JURISDICTION**. In case of any problems, the court of BRUSSELS-BELGIUM will have exclusive jurisdiction.

Agreed by downloading the document via the internet.

http://www.dedicated-systems.com
email: info@dedicated-systems.com

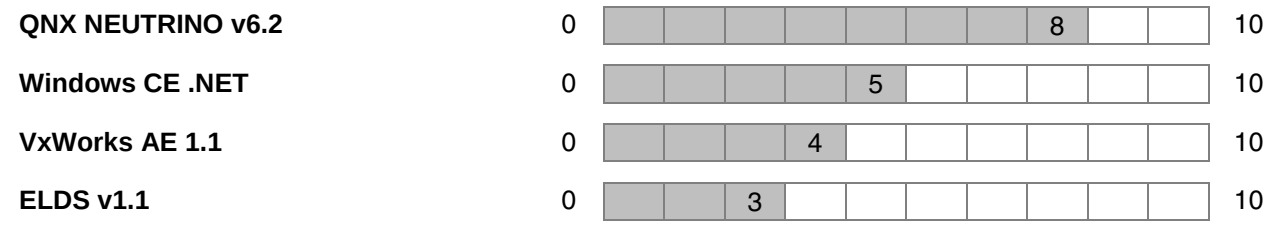
1 Introduction

Recently, Dedicated Systems Experts evaluated the following real-time operating systems (RTOS):

- The QNX NEUTRINO RTOS v6.2 from QNX Software Systems Ltd.
- The VxWorks AE 1.1 RTOS from Wind River Systems, Inc.
- Windows CE .NET from Microsoft Corporation, Inc. Version 4.0.0708 with QFE number 020607_Q323461, a fix that fixes a known scheduler problem in Windows CE .NET was applied to the RTOS. The results reported in this paper can only be obtained when this fix is applied to the system.
- The Red Hat Embedded Linux Developer's Suite v1.1 (ELDS) from Red Hat, Inc. This uses the Red Hat Linux kernel 7.2 (derived from the Linux kernel 2.4.5).

This report summarizes and compares the key elements of the full evaluation reports of both RTOS.

2 Installation and Configuration



2.1 QNX NEUTRINO RTOS v6.2

The QNX NEUTRINO RTOS v6.2 is quick and easy to install. After only a few minutes the basic modules are installed i.e., the kernel and a user interface (Photon Windowing System). Additional packages like compilers etc, can be installed by means of the package manager.

Configuring the QNX NEUTRINO RTOS v6.2 is reasonably straightforward. When installing the full environment, the most critical components like storage devices and network cards are detected automatically. If further configuration is necessary, it can be done through the graphical user interface.

Creating and configuring a custom QNX image is done in the IDE or by text-based build files. Modules can be added, removed and configured by the System Builder Tool or manually editing text-based files. The documentation contains plenty of examples of such build files.

New in the QNX NEUTRTINO RTOS v6.2 is the graphical System Builder Tool which enables the management of QNX images. It replaces the build files with a graphical tool to create images (both boot images and flash images) and allows the importing of existing build files. The system builder tool features dependency analysis (tells you which libraries might be missing), as well as a “dietician”, which creates smaller versions of the shared libraries you’re using that only contain the functions you need.

2.2 VxWorks AE 1.1

The Tornado tool suite is easy to install. You do need to make sure you have the correct installation keys. Depending on the key, different parts and modules will be installed.

After the Tornado tool suite is properly installed, developers can start using it to create custom VxWorks AE images to run on their targets. As is usually the case with embedded operating systems, the VxWorks AE RTOS is highly configurable. All this flexibility makes the configuration process intricate, but the Tornado 3.0 tool suite provides a pretty good interface to this purpose to make the task more manageable.

Nevertheless, we experienced many configuration problems during this evaluation. In fact, the evaluation of VxWorks AE 1.1 has taken longer than any other RTOS product evaluated by us before. But this was more due to the lack of (good) documentation.

2.3 Windows CE .NET

The first step to using Windows CE .NET is to install the platform builder software. Platform builder 4.0 is the set of tools that is used to create a custom Windows CE .NET platform. The platform builder comes on a DVD and is capable of creating ARM, MIPS, SH or Intel x86 based platforms. For this evaluation, only the Intel x86 component was installed. Installing platform builder is similar to installing any other Microsoft software application, and is pretty straightforward and user-friendly.

Although the platform builder integrated development environment (IDE) includes wizards for creating platforms and components, most of the configuration work will happen through manually editing registry files, manipulating a set of environment variables and modifying various other configuration scripts. This makes the configuration process a difficult task to newcomers. We were under the impression that Platform Builder 4.0 has made some progress compared to previous versions in terms of ease of configuration.

2.4 ELDS v1.1

The Red Hat Embedded Linux Developer Suite version 1.1 is a tool-set from which you can configure and build a target kernel, define the packages that are to be included on the target, and finally build the target file-system. On the host, the standard Red Hat 7.2 Linux distribution is used. Installation of the host (on a x86) is straightforward and worked on our hardware without a glitch. The only manual intervention that was required was the addition of the I/O address and the IRQ level of our NE2000 compatible ISA NIC to /etc/modules.conf.

The installation of the Embedded toolset, performed by an automatic install script, essentially involves unpacking a large number of rpm packages from the CDROM to the hard disk of the host. The script furthermore installs specific versions of Perl and Python, required by ELDS, when these should not be present on your host.

The purpose of ELDS is to have a GUI for generating a custom (embedded) target kernel. However, the use of a GUI for creating a project in ELDS, which only allows you to enter a single string, clearly presents no added value whatsoever. In combination with the command-line instructions the developer has to issue immediately thereafter, it gives the impression of being a first, incomplete attempt at what may one day become a useful GUI-based project creation tool.

The Red Hat ELDS provides no documentation whatsoever regarding the configuration of the kernel, nor does it try to verify the validity of your combination of configuration choices. Yet this is precisely a fundamental and very complex step in the development of the target embedded system.

The ELDS provides no assistance in checking the necessary dependencies to ensure that the embedded system you are configuring contains all the required libraries in order to be able to boot.

Suppose you finally have, notwithstanding the exotic combination of command-line and poor graphic tools of the Red Hat Embedded Linux Developer Suite, managed to configure and build a kernel, as well as configure and build a root file-system. You are then confronted with the problem of transferring the target kernel and target file-system to the target host.

On this topic, the ELDS "Getting Started Guide", which is the very thin and only piece of documentation you receive, only gives information for Intel's StrongARM Assabet board. Developers for other platforms are

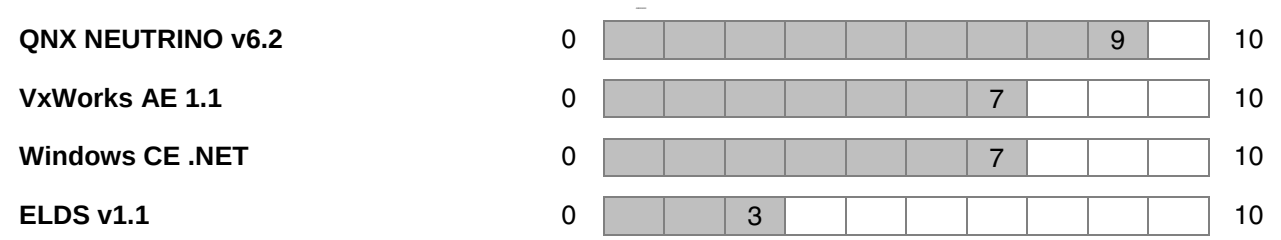
http://www.dedicated-systems.com
email: info@dedicated-systems.com

simply on their own. This information is not usable as such for other platforms, and to get it running on our x86 platform we finally had to use the good old boot/root disk combination to transfer the kernel and the file-system to our target.

From all the products that passed through our evaluation process, the Red Hat Embedded Linux Developer Suite is clearly the most difficult one to configure and install on our basic x86 target.

3 RTOS Architecture

For a description of the ratings, the reader is referred to appendix D in the document “report definition and test plan”, which can be downloaded from our website (http://www.dedicated-systems.com/encyc)



3.1 System Architecture

3.1.1 QNX NEUTRINO RTOS v6.2

The QNX NEUTRINO RTOS v6.2 has a true client-server architecture, consisting of a microkernel and optional cooperating processes. The microkernel implements only the core services, like threads, signals, message passing, synchronization, scheduling and timer services. Additional functionality is implemented in cooperative processes, which act as server processes and respond to the request of client processes (e.g. an application process). Examples of such server processes are the file system manager, process manager, device manager, network manager, etc. While the kernel runs at privilege level 0 of the Intel processor, the managers and device drivers run at levels 1 and 2 (to perform IO operations). Application processes on the other hand run at privilege level 3, and can therefore only execute general instructions of the processor.

☺ Every process runs in its own virtual memory space. The QNX NEUTRINO RTOS is a message based OS, and can seamlessly be distributed over multiple nodes. The RTOS supports SMP, and implements several HA (High Availability) features.

3.1.2 VxWorks AE 1.1

The VxWorks AE 1.1 operating system was derived from VxWorks 5.x, hence their architectures are very similar. The reader is referred to the VxWorks 5.3.1 evaluation report for more details.

Nevertheless, a new feature was added to AE: the concept of *protection domains*. A protection domain provides a logical resource “container” that defines an execution environment: each protection domain has its own virtual address space and, depending on the configuration, may or may not be visible to other domains. Whereas VxWorks 5.x uses a single flat address space shared by the system and its applications, system designers can divide their AE system in as many virtual address partitions as they see fit. To maximize robustness and system reliability, the operating system kernel should run in its own private protection domain.

Several remarks can be made about AE’s architecture and protection domains in general:

- ☺ Protection domains are an improvement compared to previous versions of VxWorks (versions 5.x) that operated in a single flat memory space. Protection domains now provide memory protection resulting in a more robust system.
- ☺ Compared to traditional processes, protection domains include some new (and nice) parameters that dictate the priority range that the domain's threads are allowed to have, and against which libraries it can link. These features could be very useful. For example, consider a team that is working on the GUI application. By putting the GUI code in a separate protection domain, you can restrict the GUI thread priorities to a range where they could never starve the system or even impact the real-time performance of the more critical threads.
- ☹ Whereas traditional processes are automatically created by the system, protection domains need to be configured by the user. This configuration is tricky and puts extra responsibility in the hands of the developer; increasing the chances errors are made.
- ☹ The protection domains did not change anything to the overall (high-level) RTOS architecture. The kernel still doesn't have intrinsic message passing capabilities, poor support for inter-processor communication in systems without shared memory, etc.
- ? An interesting question is how HA-capable (High Availability) VxWorks AE is, and if these capabilities are affected by the introduction of these protection domains. However, this issue is out of the scope of this study.

3.1.3 Windows CE .NET

Windows CE .NET is very scalable. The system is built from a set of discrete modules, each providing specific functionality. Several of these modules are divided into components, which can be individually selected. In its most compact configuration, CE .NET requires roughly 200K of ROM.

The prime modules are the kernel, the object store, the graphics subsystem and communications components. In addition to these primary modules, other modules are available and provide support for multimedia, COM (Component Object Model), Windows CE shell and device manager.

3.1.4 ELDS v1.1

ELDS v1.1 uses the Red Hat Linux 7.2 kernel which is based on the general linux kernel v2.4.5.

Linux has its roots in the Unix General Purpose Operating System (GPOS), it has a traditional monolithic kernel and is clearly not built for real-time purposes. Fortunately, Red Hat does not claim real-time behavior for the Linux kernel used by the ELDS.

But let's start with the good things: Linux is process based and has virtual memory protection between the different user processes and between a user and kernel process. The kernel can be extended by so called "modules" (device drivers). The kernel is well protected against faults in such modules. Swapping (virtual memory swapping to disk) can be avoided by not mounting a swap file-system.

On the other side, it is clear that Linux is not built for real-time purposes. So what's wrong in the Linux system architecture to make it unsuitable for real-time?

- The kernel is not pre-emptive (there is now a patch available to make it pre-emptive)

- Monolithic OS: Linux is still a monolithic kernel, using large and complex shared data structures. As a result the critical sections are large (and mostly used more than needed) and can lock the processor for a considerable amount of time.
- Primitive kernel (threading/process support): threading and process supported by the rather primitive “spawn()” and “clone()” call. Other APIs are build on these calls.
- No priority inversion avoidance mechanism: This is a problem in all GPOS. As these OS are not built for real-time systems, there is no need for the rather complex protection against priority inversion.

3.2 Basic System Facilities

3.2.1 Task Handling Method

All operating systems are multi-threaded. The QNX RTOS, Windows CE and ELDS use traditional processes, while VxWorks AE has its protection domains.

	QNX NEUTRINO v6.2	VxWorks AE 1.1	Windows CE .NET	ELDS v1.1
Model	Threads and processes	Threads and protected domains.	Processes and Threads	Threads and processes
Priority levels	64	256	256 levels	99
Maximum number of tasks	4095 processes Every process can have 32767 threads	Limited by the amount of memory available.	The maximum number of *threads* in a process is only limited by the amount of memory available. Windows CE can run a maximum of 32 *processes* simultaneously.	Limited by the amount of memory available and configuration settings
Scheduling policy	Prioritized FIFO Round-robin scheduling Adaptive Sporadic	Preemptive priority. Round-robin.	Round-robin with adjustable time-slice (quantum) When the quantum is set to zero, the thread runs to completion	Prioritized FIFO round-robin scheduling. Regular non real-time.
Number of documented states	14		5	From sched.h : 6 (not in documentation)

3.2.2 Memory Management Method

All operating systems use full virtual memory protection, significantly increasing the robustness and reliability of the system.

	QNX NEUTRINO v6.2	VxWorks AE 1.1	Windows CE .NET	ELDS v1.1
MMU support	Yes	Yes	Yes	Yes
Physical page size	Depends on architecture	Depends on architecture	Depends on architecture	Depends on architecture
Swapping/Demand Paging	Yes/No	Never	Supported, but can be disabled to achieve real-time performance.	Yes/No
Virtual memory	Yes	Yes	Yes	Yes
Memory protection models	Full virtual memory protection	Full virtual memory protection.	Full virtual memory protection.	Full virtual memory protection.

3.2.3 Interrupt Handling Method

The interrupt handling is prioritized, and interrupt handlers can be nested.

For the linux kernel, there is no documentation available (with ELDS) to understand the interrupt handling internals. The only way is to look into the source code…

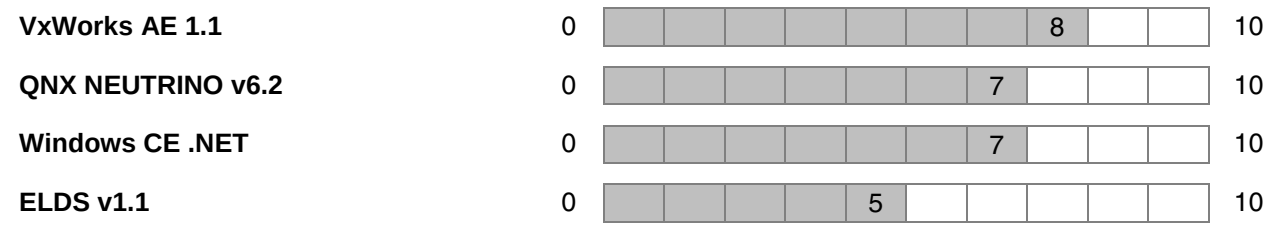
	QNX NEUTRINO v6.2	VxWorks AE 1.1	Windows CE .NET	ELDS v1.1
Handling	Nested, prioritized	Nested, prioritized[1].	Nested, prioritized	Nested, prioritized
Context	The ISR runs in the context of the thread that attached it	Interrupt handlers run in a special context, outside any task's context.	The ISR runs in a special context and uses virtual addresses statically mapped by the OEM. The IST is a normal application thread	Not documented

[1] This is what the documentation claims. Our tests however revealed a bug in the x86 BSP that caused interrupts between IRQ8 and IRQ15 not to be nestable. In the means time, Wind River has a cumulative patch available with a fix for this problem.

	QNX NEUTRINO v6.2	VxWorks AE 1.1	Windows CE .NET	ELDS v1.1
			and has its own context.	
Stack	The ISR has its own stack	Special interrupt stack. Only one system wide interrupt stack allocated from the kernel's memory	The IST is a normal application thread and has its own stack.	Not documented
Interrupt to task communication	Signals and pulses	Shared memory and ring buffers (cannot create or delete ring buffers) Semaphores (release only), Message queues (send only), Pipes (write only), Signals (send only)	Only an event can be used from within the ISR to signal the IST. No other API is accessible from within the ISR. OEM can create a shared memory region by statically mapping a memory region into the ISR's address space	Not documented

4 API Richness

For a description of the ratings, the reader is referred to appendix D in the document “report definition and test plan”, which can be downloaded from our website (http://www.dedicated-systems.com/encyc)



While interpreting these results, the reader should keep in mind that these tables cover a strictly defined set of the most commonly used system calls. All OS have system calls that are not covered by the table below. For more details on what features the different categories in the table below encompass, the reader is referred to the evaluation reports.

	QNX NEUTRINO v6.2	VxWorks AE 1.1	Windows CE .NET	ELDS v1.1
Task management	82%	88%	71%	47%
Clock	100%	71%	85%	43%
Timer	100%	100%	67%	0%
Memory management	27%	77%	35%	0%
Interrupt handling	88%	50%	50%	63%
Semaphore	35%	90%	40%	31%
Mutex	67%	92%	67%	31%
Conditional Variables	60%	60%	0%	0%
Event flags	0%	0%	75%	0%
POSIX signals	100%	78%	0%	78%
Message queue and Mailbox	41%	81%	75%	12%

The API of the QNX NEUTRINO RTOS, VxWorks AE 1.1 and Windows CE .NET are sufficiently rich APIs, but they have a different emphasis. As the QNX NEUTRINO RTOS has a message based architecture, its API also focuses on messaging. Mechanisms like messages and signals are preferred over semaphores or

http://www.dedicated-systems.com
email: info@dedicated-systems.com

other classic synchronization objects. This is less the case for the VxWorks AE and Windows CE .NET APIs.

On the other hand the Linux API is not elaborated enough to use in real-time systems. For instance, there are no timer nor memory support functions (except the traditional alloc() call).

5 Internet support

For a description of the ratings, the reader is referred to appendix D in the document “report definition and test plan”, which can be downloaded from our website (http://www.dedicated-systems.com/encyc)

VxWorks AE 1.1	0	9	10
Windows CE .NET	0	9	10
QNX NEUTRINO v6.2	0	8	10
ELDS v1.1	0	8	10

5.1 QNX NEUTRINO RTOS v6.2

QNX Neutrino’s Internet Technology Suite contains the following products and tools:

- Voyager Web Server: an HTTP server to serve web pages. It has support for dynamic HTML pages via SSI (Server Side Includes).
- A Voyager Web Browser for viewing information. It has full HTML 3.2 support, frames support, javascript, cookies, etc. The Mozilla and Opera browsers have recently also been ported to run on the QNX NEUTRINO RTOS v6.2.

A Voyager SDK (Software Development Kit) to build internet-enabled applications into an embedded system.

5.2 VxWorks AE 1.1

VxWorks AE 1.1 offers the same internet and network support as VxWorks 5.x, except some graphics options. As such, VxWorks AE has very extensive networking and protocol support. The reader is referred to the Wind River website for detailed information.

5.3 Windows CE .NET

Windows CE .NET has very extensive Internet support

Windows CE .NET comes with an extensive set of Internet products and tools. It includes, among others:

- An HTTP server to post information. The server supports active server pages, ISAPI extensions and filters.
- A web browser for viewing information. The browser is a miniature version of Internet Explorer. It supports frames, tables and Javascript, as well as JPEG, static and animated GIF and WAV files.
- A telnet server to remotely administer devices, or to administer devices that do not have displays.
- Networking protocol support for communicating across the internet/intranet.

Many other tools and utilities are available from third-party vendors.

http://www.dedicated-systems.com
email: info@dedicated-systems.com

5.4 ELDS v1.1

There are a lot of servers CGI script engines, browsers and so one available for the Linux Platform.

6 Tools

For a description of the ratings, the reader is referred to appendix D in the document “report definition and test plan”, which can be downloaded from our website (http://www.dedicated-systems.com/encyc)

QNX NEUTRINO v6.2	0	8	10
ELDS v1.1	0	8	10
VxWorks AE 1.1	0	8	10
Windows CE .NET	0	8	10

6.1 QNX NEUTRINO RTOS v6.2

QNX has its own Momentics IDE toolsuite. There are also two other sets of tools available for the QNX NEUTRINO RTOS v6.2: the Metrowerks Codewarrior IDE and the GCC toolkit. Tools for both self-hosted and cross development are available. These toolkits contain the most commonly used tools.

Although the Momentics IDE has all features needed and is pretty intuitive to use, you will need a lot of processing power and ram to get it running at a comfortable speed!

6.2 VxWorks AE 1.1

The VxWorks AE RTOS comes with the Tornado 3 IDE. Tornado 3 comprises an extensive suite of tools and utilities that can be used during the development and debugging phase. The full evaluation report contains an inventory of the most commonly used tools available.

There is in fact only one negative aspect to the Tornado 3 IDE: its “clunkiness”. Huge TCL scripts need to be parsed whenever an action is taken, which requires a lot of processing power. Make sure to run Tornado on the latest and greatest hardware only.

6.3 Windows CE .NET

Platform Builder 4.0 (PB) has a few new features that make configuring a Windows CE .NET image somewhat easier. A new platform wizard that assists you while creating a new platform is one example. Still, we encountered some very annoying bugs in the tool. Quite often the PB crashes when one disconnects the host from the target. On one occasion, the crash apparently destroyed some key files on our host that rendered it inoperable. Our platform builder needed to be reinstalled for this reason.

6.4 ELDS v1.1

Embedded Linux developers have a wide range of development tools at their disposal. There is the wide range of GNU tools (editors, compilers for different languages such as gcc and g++, the gdb debugger, the gas assembler, de ld linker, ...) as well as all the other tools that have been developed for the Linux/GNU system. For a number of GNU tools, graphical front-ends have been developed (Source Navigator or

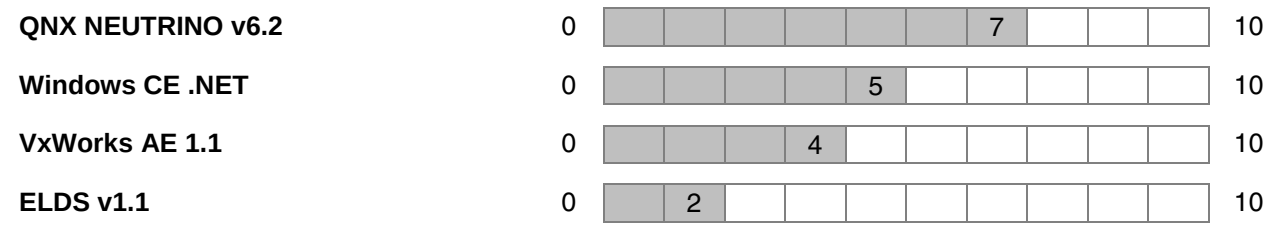
http://www.dedicated-systems.com
email: info@dedicated-systems.com

KDevelop C/C++ IDE, DDD or Insight front-end for gdb, ...). Most of these tools are integrated in the Red Hat distribution and the developer has plenty of choice. Installation of the tools is easy thanks to the rpm package management system.

http://www.dedicated-systems.com
email: info@dedicated-systems.com

7 Documentation and Support

For a description of the ratings, the reader is referred to appendix D in the document “report definition and test plan”, which can be downloaded from our website (http://www.dedicated-systems.com/encyc)



7.1 QNX NEUTRINO RTOS v6.2

The documentation does a decent job giving a general overview of the system and its architecture. The documentation has improved compared with the QNX v6.1 documentation: the meaning of API parameters are now well explained.

7.2 VxWorks AE 1.1

The VxWorks/Tornado documentation doesn’t provide a good overview of the inner workings of the system. It also doesn’t contain all the information to guide users through the complex configuration process. We had to seek assistance of Wind River’s technical support staff to help us out with a variety of configuration chores (configuring network drivers, etc), simply because we could find no reference in the documentation of how to go about it.

7.3 Windows CE .NET

Windows CE comes with an online documentation set. This documentation set contains a lot of information, but it is not presented in a very structured way. The documentation can easily be used as a reference, but is less appropriate as a tutorial. Newcomers will have a hard time acquiring an overview of the system if this is the only documentation they have at their disposal. It also lacks in-depth information about the inner workings of the system. Documenting the APIs and available features is not enough to provide the reader with a sufficient understanding of a complex system like Windows CE .NET.

7.4 ELDS v1.1

Linux is a free open source operating system, built by a virtual community of enthusiast programmers. Unfortunately, programmers tend to prefer writing code over writing documentation.

In the Linux development model, there are no resources that are dedicated to writing the documentation that goes along with the kernel code and the tools that are developed by the volunteer programmers. Apparently, there are no - or at least very few - technical writers that feel the urge to participate in the Linux development effort.

http://www.dedicated-systems.com
email: info@dedicated-systems.com

There is of course the Linux Documentation Project. There is the documentation that is available in the "/usr/src/linux/Documentation" directory of the kernel source, together with (sometimes) comments in the source files. There are the numerous web pages, created by enthusiasts all over the world. There are the archives of newsgroups and mailing lists. Is all this not enough then?

Actually, no it isn't. The Internet is a possible source of information when you use your Linux system for one of the things a lot of other Linux users use their system for as well, for instance a Linux web server or firewall. However, when developing an embedded Linux system, the community of information providers via web pages and participants via newsgroups is a lot smaller, so the probability that the information you need is available somewhere on the Internet becomes relatively low.

At present, the Red Hat Embedded Linux Developer Suite provides very little or no documentation, neither in digital form nor on paper. As a result, the learning curve when using Linux in an embedded project is a steep one.

8 Development methodology

8.1 QNX NEUTRINO RTOS v6.2

The QNX NEUTRINO RTOS originally used the host = target approach only i.e., host and target are the same machine. As was mentioned earlier, the QNX NEUTRINO RTOS v6.2 can be configured with only a microkernel, as well as with many other modules turning it into a fully fledged multi-user operating system capable of serving as a development environment. The advantage of this approach is that the user has the option to do it all on one machine: the application can be tested on the same machine as it was developed on, debugging can be done locally, etc. There are no problems with communication between host and target.

Developers that prefer a standard MS-Windows desktop to the QNX desktop can use cross-development tools. The Metrowerks IDE for MS-Windows can be used to allow the user to do the compiling and debugging from the Windows based host machine.

There's often a lot of discussion about which development method is the better one: self-hosted or cross development. It all really depends on the quality of the tools. If the quality of the cross-development tools is poor, it is better to opt for self-hosted development, and vice versa. Unfortunately, evaluating the quality of the development tools is not within the scope of this report. Nonetheless, it is important advantage that an OS support both methods.

8.2 VxWorks AE 1.1

Wind River Systems uses the *host* ≠ *target* approach. Host and target are two different machines linked together (serial, LAN, bus, etc) for communication. The host is the machine on which the development environment (Tornado 3.0) runs. Tornado is available for both Windows and UNIX based hosts. The target is the machine on which the dedicated RTOS (VxWorks AE 1.1) runs with the real-time application.

The advantage of this approach over the host = target approach is that a separate host is more suitable as a system development host. Indeed, all the features of the GPOS (Windows NT or UNIX) can be used, which allows for a better and more complete development environment.

In this configuration however, the debugger is on the host, while the real-time application is executed on the target. To make interactive debugging possible, VxWorks installs so called "debug agents" on the target that communicate debug information to the host. These debug agents provide a virtual seamless integration of host and target while debugging i.e., the user is under the impression that he is debugging a local application.

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, Wind River 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. VxSim was not tested for this evaluation.

8.3 Windows CE .NET

Microsoft provides development tools to cater the need of two categories of developers: the platform developers and the application developers.

Platform developers use an integrated development environment called Platform Builder on the Windows NT based host, while the target runs Windows CE with the (real-time) applications. Aside from this platform development tool, Microsoft also provides the Embedded Visual Tools for application development.

The Platform Builder can be used to create a custom SDK (Software Development Kit) based on the Windows CE OS to allow developers to write applications that run on the target platform. An SDK is a set of library, header, and Help files that developers use to write applications for a specific platform. The SDK is used in conjunction with the Embedded Visual Tools to create, debug and run custom applications.

8.4 ELDS v1.1

Red Hat uses the host ≠ target approach. The development environment (ELDS) runs on a standard x86 Red Hat 7.1 or 7.2 host system. The target can be any of the major targets: ARM, MIPS, PowerPC, SuperH or x86.

On the host, the developer can use the wide range of well-known and proven GNU development tools that are available, in combination with a large number of graphical tools and graphical front-ends that are packaged with the Red Hat distribution.

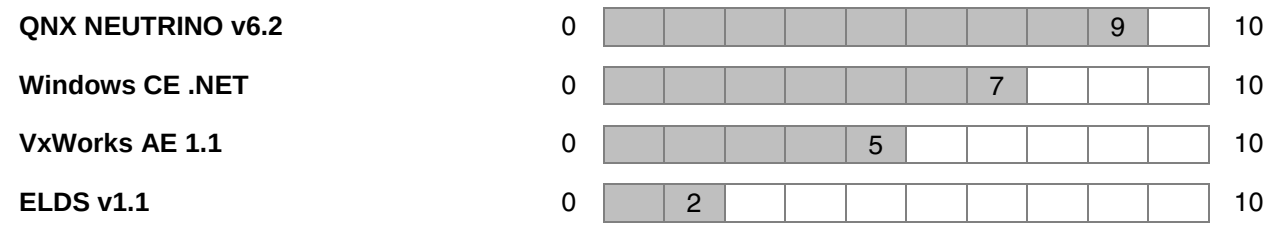
Red Hat provides RedBoot as an embedded debug and bootstrap solution on all the supported targets. RedBoot supports downloading and debugging of applications, flash and network booting of the Linux kernel, and downloading and updating of flash images remotely via serial or Ethernet connections.

RedBoot furthermore provides a target-side hook (a "stub") that enables gdb, the GNU debugger, to communicate with applications built with gcc or g++ running on the target board.

The Linux kernel can be compiled with support for dynamically loadable on demand kernel modules. In this way, during development, different development versions of a driver can be loaded and unloaded.

9 Test results

For a description of the ratings, the reader is referred to appendix D in the document “report definition and test plan”, which can be downloaded from our website (http://www.dedicated-systems.com/encyc).



9.1 QNX NEUTRINO RTOS v6.2

The QNX NEUTRINO RTOS exhibited fast and predictable behaviour during all phases of our testing. None of the tests revealed any problems of any kind. Hence the QNX NEUTRINO RTOS was given a score of 9 for this category.

9.2 VxWorks AE 1.1

In general, VxWorks AE 1.1 is slower than the QNX NEUTRINO RTOS v6.2 (on a x86 platform). However, our tests also revealed some (serious) issues, hence VxWorks was rated a score of 5. The issues that were discovered are:

- A bug in the Pentium BSP that prevents interrupt nesting. This is a fundamental problem in any real-time system. Wind River Systems sent us a quick fix that corrected this problem. The vendor has informed us that this quick fix has developed into a cumulative patch in the mean time.
- The execution of the clock ISR can become pretty lengthy (>60µs under certain circumstances), making the system less responsive to other external interrupts. This was confirmed in our stress tests.
- A socket-related bug in the TCPIP stack that can severely hamper data transmission throughput.

9.3 Windows CE .NET

While Windows CE .NET performs slower than the QNX NEUTRINO RTOS v6.2, it exhibited true real-time behavior during all of our tests.

9.4 ELDS v1.1

As expected, our tests revealed that the Red Hat Linux kernel is not foreseen for real-time purposes. This is normal, as it is a GPOS optimised for General Purpose applications. It has to be said that Red Hat Inc. does not claim any real-time behaviour.

If you would compare the tested Red Hat Linux kernel (7.2) with another GPOS we tested some years ago (Windows NT 4.0) then the results are similar (some are better on the linux platform others are better on Windows NT). Both OSs are not capable of keeping time constraints.

For Windows NT we therefore gave a score of 2 on the real time behaviour found in the test results. For the Red Hat Linux kernel, the score had to be less as we found major flaws in the behaviour of some POSIX API calls. Although Windows NT 4.0 is not real-time we didn't found API flaws in it… Finally we decided to give it the same score as Windows NT 4.0 due to the better interrupt handling.

The main problems detected in the Red Hat Linux kernel are:

- A Semaphore release does not imply a rescheduling of ready threads (even if a higher priority thread was blocked on the semaphore).

  It has to be said that this behaviour is stated as such in the linux manpages! But this makes this API call not compliant with the behaviour as defined in the POSIX standards! The behaviour is surely not what a real-time programmer expects.

- Both thread yield calls (as defined in POSIX 1003.1b and 1003.1c: sched_yield() and pthread_yield()) do not work. They do something, but the behaviour is not like it is defined in the POSIX standards (put the thread at the back of the ready queue on it's priority level)

This bug is known and documented already since the late 1990's, however it is still not solved in the Red Hat Linux 7.2 kernel.

9.5 Comparison test results

The same test suite was applied to all operating systems.

Table 1 presents most of the performance test results for all OS. It does not include the results of the stress tests or the TCP/IP tests. When a test for a particular RTOS is marked "TNE", it means this test was not executed. For a summary of the test identification codes, please refer to Appendix B: Summary test identification codes.

Remarks for the ELDS v1.1:

- The semaphore implementation is not correct: therefore the creation/deletion is fast, but when releasing a semaphore the kernel does not reschedule. We had to make a workaround to have comparable results with QNX.
- As most GPOS, the average time is good. However, for real-time performance the worst case and dependence of the worst case with system load is far more important.

	QNX NEUTRINO v6.2		VxWorks AE 1.1		Windows CE .NET		ELDS v1.1	
Test identification	Average (µs)	Max. (µs)	Average (µs)	Max. (µs)	Average (µs)	Max. (µs)	Average (µs)	Max. (µs)
IL-a-1_ISR	1.7	4.3	1.7	6.8	2.4	5.6	3.2	4.0
IDL-a-1_ISR	1.9	2.7	1.9	8.4	TNE	TNE	1.2	2.0
IL-a-1_IST	2.3	7.7	6.5	14.2	7.7	12.9	TNE	TNE
SI-a-1_ISR_HI	1.6	2.5	1.8	4.7	2.5	5.8	3.0	5.3
SI-a-ISR_LO	4.1	4.9	4.3	8.3	7.2	10.4	7.0	10.9
TF-a-1	175	2880	260	299	102	1006	231	296
TF-b-1	78	102	85	103	280	922	225	884
TSL-a-2	2.6	8.3	2.9	15.5	2.6	35.1	1.7	101
TSL-a-10	2.8	7.9	3.4	16.5	3.3	37.4	2.8	123
TSL-a-128	3.6	9.5	6.5	29.6	5.3	63.9	25	536
TSL-b-128	8.8	21.8	6.8	46.8	9.6	16.7	TNE	TNE
SEO-a-1	3.4	8.5	7.4	19.4	3.6	8.3	0.18	12.3
SEO-b-1	3.2	10.2	9.0	20.4	3.6	11.1	0.18	8.5
SEO-d-1	2.5	9.1	0.2	7.5	3.9	8.4	1.2	9.7
SEO-e-1	2.6	6.4	0.2	7.1	4.2	11.3	1.0	27.7
SEO-f-max	6.6	12.7	5.9	19.1	9.1	13.5	22	61
SEO-g-3 (mutex)	7.4	13.7	9.3	18.0	19.4	34.5	0.6	33.4
FS-a-1	6575	11596	TNE[2]	TNE	48[3]	162[3]	74	8902
FS-b-1	578	759	TNE	TNE	23.1	28.4	79	127
FS-c-1 (1 byte)	93	116	TNE	TNE	17	58	3.6	33
FS-c-1 (1 block)	123	440	TNE	TNE	38	154	9.7	41.3

[2] We were not able to mount a Hard Drive on the system

[3] As opposed to QNX 6.2 where a IDE hard drive was used, the file system tests on Windows CE .NET were executed with a RAM-based file system.

	QNX NEUTRINO v6.2		VxWorks AE 1.1		Windows CE .NET		ELDS v1.1	
Test identification	Average (µs)	Max. (µs)	Average (µs)	Max. (µs)	Average (µs)	Max. (µs)	Average (µs)	Max. (µs)
FS-c-1 (10 blocks)	326	710	TNE	TNE	238	390	100	224
FS-d-1 (1 byte)	2340	29998	TNE	TNE	21	59	2380	21180
FS-d-1 (1 block)	20936	54369	TNE	TNE	47	229	6880	46531
FS-d-1 (10 blocks)	24519	58778	TNE	TNE	312	600	26310	60738

Table 1: Performance results for the different RTOS

Aside from the performance tests in Table 1, we also executed stress tests. These stress test try to detect memory leaks or performance degradation when the system is loaded. No such problems were detected in any OS. Also Red Hat Linux 7.2 did not have any memory leaks, however it did exhibit performance degradation when the system is loaded.

Another stress test is the “billion interrupt” test. It generates a billion (10^9) interrupts (at the same IRQ level) at a programmable frequency, and we count how many interrupts were serviced, and how many of them were lost. The test is considered successful if not one single interrupt is lost. An interrupt is “lost” when by the time the next interrupt needs to be generated, the previous one is still being serviced or masked out by a higher priority interrupt (e.g. the clock interrupt). Therefore, this test gives a pretty good idea of the worst-case interrupt latency of the RTOS.

Table 2 presents the results. The QNX NEUTRINO RTOS v6.2 served every single interrupt when they were generated every 9µs. Windows CE .NET only needed 2µs more. In VxWorks AE 1.1 however, this period needed to be increased to 25µs in order for the test to succeed. The Red Hat Linux kernel could only handle them all when they were generated with a 60µs interval. Here clearly the difference between a RTOS and a GPOS is illustrated!

	QNX Neutrino v6.2	Windows CE .NET	VxWorks AE 1/1	ELDS v1.1
Maximum Sustainable Interrupt Frequency	9µs	11µs	25µs	60µs

Table 2: Maximum sustainable interrupt frequency – Endurance Test

10 Conclusion

The QNX NEUTRINO RTOS v6.2, Windows CE .NET, VxWorks AE 1.1 and Red Hat Linux 7.2 Operating Systems were evaluated against the same criteria and test suite.

The QNX NEUTRINO RTOS v6.2 performed very well during this evaluation. None of the performance or stress tests revealed any problems and the RTOS was fast, predictable and reliable at all times. The QNX NEUTRINO RTOS is also the only RTOS that has a true message-based client-server architecture well equipped to handle today's requirements concerning distributed processing, high availability, etc.

VxWorks AE 1.1 is Wind River Systems its latest RTOS product. It introduces the concept of protection domains. Protection domains provide a memory protection scheme and therefore present a big step forward compared to previous version of VxWorks, which operated in a single flat memory space. Aside from these protection domains, no other advancements were made to the RTOS architecture. Our test suite did expose some acute problems, the most serious one being a bug in the x86 Pentium BSP that prevented interrupt nesting. We found another problem in the TCPIP stack that causes the transmission throughput to collapse under certain circumstances. Aside from these issues, VxWorks AE's (real-time) performance is acceptable, but no match for the QNX NEUTRINO RTOS.

Windows CE .NET is the successor of Windows CE 3.0. Windows CE .NET exhibited real-time behavior during our tests. None of the stress tests exposed any problems concerning stability and robustness either.

The Red Hat Embedded Linux Developer's Suite v1.1 (Red Hat Linux kernel 7.2 based on the Linux kernel 2.4.5) is clearly not foreseen to be used in a real-time environment. Linux is made as a GPOS and the test results illustrate this. The added value of the Embedded Linux Developer's Suite is questionable: it does not make it easier to generate a custom target platform.

Although the Linux kernel is royalty free, it comes with a price: documentation is poor and the API is not compatible with (POSIX) standards. The learning curve to get the kernel up and running on your custom target platform is steep.

! Finally, the reader should bear in mind that we tested all OS on an Intel x86 platform only. The results in this report apply to the x86 platform only, not to any of the other platforms these products support.

http://www.dedicated-systems.com
email: info@dedicated-systems.com

11 References

[1] The QNX NEUTRINO RTOS v6.2 evaluation report, Dedicated Systems Experts, 2002. (http://www.dedicated-systems.com)

[2] The VxWorks AE 1.1 RTOS evaluation report, Dedicated Systems Experts, 2002. (http://www.dedicated-systems.com)

[3] The Red Hat ELDS v1.1 evaluation report, Dedicated Systems Experts, 2002. (http://www.dedicated-systems.com)

http://www.dedicated-systems.com
email: info@dedicated-systems.com

12 Appendix A: Product ratings

This appendix contains the evaluation report summary pages for QNX NEUTRINO RTOS v6.2, VxWorks AE 1.1, Windows CE .NET and Red Hat ELDS v1.1.

12.1 QNX NEUTRINO RTOS v6.2

Product

The QNX NEUTRINO RTOS v6.2.0, from QNX Software Systems Ltd.

Positive points

- Fast performance
- Excellent architecture for a distributed and robust system
- Good platform support

Negative points

- Slow Integrated Development Environment

Ratings

For a description of the ratings, the reader is referred to appendix D in the document “report definition and test plan”, which can be downloaded from our website (http://www.dedicated-systems.com/encyc)

Installation and Configuration	0	8	10
RTOS Architecture	0	9	10
API Richness	0	7	10
Internet support	0	8	10
Tools	0	8	10
Documentation and Support	0	7	10
Test Results	0	9	10

Pricing

Please contact the vendor for detailed and up-to-date pricing information.

12.2 VxWorks AE 1.1

Product

VxWorks/x86 AE 1.1 from Wind River Systems, Inc. (http://www.windriver.com)

Corporate headquarters: 500 Wind River Way. Alameda, CA 94501 Tel: (800) 545 WIND

Positive points

– VxWorks AE now also provides memory protection by means of the protection domains

Negative points

– Quality of documentation is less than mediocre

– Major bug in x86 BSP that prevents interrupt nesting (cumulative patch available)

– Slow worst-case reaction to external interrupts (x86 platform)

– Poor TCP/IP network performance due to socket-related bug in IP-stack (cumulative patch available).

Ratings

For a description of the ratings, the reader is referred to appendix D in the document “report definition and test plan”, which can be downloaded from our website (http://www.dedicated-systems.com/encyc)

Installation and Configuration	0	4	10
RTOS Architecture	0	7	10
API Richness	0	8	10
Internet support	0	9	10
Tools	0	8	10
Documentation and Support	0	4	10
Test Results	0	5	10

12.3 Windows CE .NET

Product

Windows CE .NET from Microsoft Corporation, Inc.

Positive points

- Extensive platform support
- Generally good real-time performance

Negative points

- Documentation insufficient for such a complex system

Ratings

For a description of the ratings, the reader is referred to appendix D in the document “report definition and test plan”, which can be downloaded from our website (http://www.dedicated-systems.com/encyc)

Installation and Configuration	0	5	10
RTOS Architecture	0	7	10
API Richness	0	7	10
Internet support	0	9	10
Tools	0	8	10
Documentation and Support	0	5	10
Test Results	0	7	10

12.4 Red Hat ELDS v1.1

Why did we test Red Hat Embedded Linux 7.2 in our RTOS evaluation program, although Red Hat and the Linux community clearly states that Linux is not meant to be used in real-time environments?

A lot of our readers explicitly asked for it. However Linux is not an RTOS! The reader should keep this in mind when going through the results published in this report.

Product

Red Hat Embedded Linux Developer Suite (ELDS) version 1.1 from Red Hat Inc. This is based on the Red Hat 7.2 kernel (derived from the linux 2.4.5 kernel).

Positive points

- No license fee
- Used at many universities: chance of finding graduates with a (basic) knowledge of Linux

Negative points

- ELDS provides little or no assistance to configure the embedded target's kernel.
- There is no documentation provided with the ELDS, apart from a very thin booklet.
- No real-time behaviour
- API not compliant with POSIX standards

Ratings

For a description of the ratings, the reader is referred to appendix D in the document "report definition and test plan", which can be downloaded from our website (http://www.dedicated-systems.com/encyc)

Installation and Configuration	0	3	10
RTOS Architecture	0	3	10
API Richness	0	5	10
Internet support	0	8	10
Tools	0	8	10
Documentation and Support	0	2	10
Test Results	0	2	10

Pricing

Although (Red Hat) Linux itself is a free operating system, the embedded toolset is not. Contact the vendor for pricing information.

However, you do not need the embedded toolset to build a custom (tiny) kernel and build an embedded Linux system.

13 Appendix B: Summary test identification codes

Test identification	Description
IL-a-1(_IST or _ISR)	Interrupt Latency (task to interrupt handler) – no rescheduling – 1 thread. Measured on IST or ISR level.
IDL-a-1(_IST or _ISR)	Interrupt Dispatch Latency (interrupt handler to task) – no rescheduling – 1 thread. Measured on IST or ISR level.
IDL-b-1	Interrupt Dispatch Latency (interrupt handler to task) – with rescheduling – 1 thread.
IDL-b-10	Interrupt Dispatch Latency (interrupt handler to task) – with rescheduling – 10 threads.
IDL-b-128	Interrupt Dispatch Latency (interrupt handler to task) – with rescheduling – 128 threads.
SI-a-1_ISR_HI	Simultaneous interrupts. Interrupt latency of the high priority ISR
SI-a-1_ISR_LO	Simultaneous interrupts. Interrupt latency of the low priority ISR
TF-a-1	Thread Creation time
TF-b-1	Thread Deletion time, thread did not execute
TF-c-2	Thread Deletion time, thread has executed.
TSL-a-2	Thread Switch Latency – 2 threads in the same process
TSL-a-10	Thread Switch Latency – 10 threads in the same process
TSL-a-128	Thread Switch Latency – 128 threads in the same process
TSL-b-128	Thread Switch Latency – 128 threads in different processes
SEO-a-1	Synchronization & Exclusion Object (semaphore) Creation time
SEO-b-1	Synchronization & Exclusion Object (semaphore) Deletion time – semaphore was not used.
SEO-c-1	Synchronization & Exclusion Object (semaphore) Deletion time – semaphore was used.
SEO-d-1	Synchronization & Exclusion Object (semaphore) Acquisition time – no contention.
SEO-e-1	Synchronization & Exclusion Object (semaphore) Release time – no contention.
SEO-f-max	Synchronization test executed with a number of threads equal to the number of task priority levels.

http://www.dedicated-systems.com
email: info@dedicated-systems.com

Test identification	Description
SEO-g-3	Synchronization & Exclusion object (Mutex) – Priority inversion prevention time.
FS-a-1	File creation.
FS-b-1	File deletion.
FS-c-1	File synchronous read.
FS-d-1	File synchronous write.

http://www.dedicated-systems.com
email: info@dedicated-systems.com

14 Appendix C: Document revision history

14.1 Issue 2.50 (November 7, 2002)

Initial issue.

Remark that the main version number increases if tests are added to our test suite. The extended test suite (Version 2) contains also network, disks and memory leak tests.