Chimera 3.2

A real-time operating system developed at CMU which is far better than anything else you'll find on the market.

Table of Contents

Overview

The Chimera 3.2 Real-Time Operating System has its own HTML page which more than adequately covers what it does, and how to use it. Let me therefore take this opportunity to answer the question, "Why use Chimera instead of VxWorks?" (Besides the fact that it has a wonderful graphical interface called Onika , I mean.)

Here's what Chimera inventor David B. Stewart has to say on the subject:

Frequently Asked Questions about Chimera vs. VxWorks

Q: What are the similarities and differences between the VxWorks and Chimera real-time kernels?

A: As with most other commercial real-time operating systems, VxWorks only provides a basic real-time kernel which gives you task management, low-overhead context switching and local semaphores, static highest priority first scheduling, and an interface to program the hardware timers for periodic events.

Chimera provides an extended real-time kernel. In addition to all the basic functions that VxWorks provides, it also gives you both static and dynamic real-time scheduling, virtual timers which eliminate the need to explicitly program hardware timers, the novel deadline failure handling and global error handling mechanisms, and two-level device driver support.

Q: Chimera claims it is a "multiprocessor operating system", while VxWorks only [illegible] "multiprocessor support". What is the difference?

Chimera 3.2

A real-time operating system developed at CMU which is far better than anything else you'll find on the market.

Table of Contents

Overview

The Chimera 3.2 Real-Time Operating System has its own HTML page which more than adequately covers what it does, and how to use it. Let me therefore take this opportunity to answer the question, "Why use Chimera instead of VxWorks?" (Besides the fact that it has a wonderful graphical interface called Onika , I mean.)

Here's what Chimera inventor David B. Stewart has to say on the subject:

Frequently Asked Questions about Chimera vs. VxWorks

Q: What are the similarities and differences between the VxWorks and Chimera real-time kernels?

A: As with most other commercial real-time operating systems, VxWorks only provides a basic real-time kernel which gives you task management, low-overhead context switching and local semaphores, static highest priority first scheduling, and an interface to program the hardware timers for periodic events.

Chimera provides an extended real-time kernel. In addition to all the basic functions that VxWorks provides, it also gives you both static and dynamic real-time scheduling, virtual timers which eliminate the need to explicitly program hardware timers, the novel deadline failure handling and global error handling mechanisms, and two-level device driver support.

Q: Chimera claims it is a "multiprocessor operating system", while VxWorks only claims it has "multiprocessor support". What is the difference?

extensions involve networking multiple VxWorks systems over the backplane using the TCP/IP protocol. The only difference is that the communication medium is the backplane, and not ethernet. This setup is no more a multiprocessor setup than two Sun workstations on the same local area network.

Chimera is a multiprocessor operating system because the kernels on multiple Real-Time Processing Units (RTPUs -- i.e. single board computers) can communicate with each other via Chimera's high performance system-level communication, called express mail. Without such kernel-level interprocessor communication, an operating system cannot make interprocessor communication transparent to the user's programs running on each RTPU, and limit both the types of higher-level communication mechanisms that can be designed and their performance.

Q: We have written a communications layer above VxWorks? Will this work with Chimera?

A: Because Chimera is a multiprocessor operating system, there is no need for such a communications layer. Almost anyone using multiple processors and VxWorks is required to write such a layer, because of VxWorks' lack of multiprocessor support. In Chimera, all the communication mechanisms are built-in to the operating system, so you don't have to write them. Because they are built-in, everyone uses the same ones; as opposed to what happens today with projects using VxWorks, where every project or institution has their own custom layer, thus limiting any potential software reuse or technology transfer.

Q: What are the communication mechanisms available in Chimera, and how does each compare to VxWorks?

A: Here is a quick rundown of most of them:

- Express Mail: this is a high-performance real-time non-blocking communication mechnism between the kernels on each RTPU and the host workstation. This mechanism allows the other interprocessor communication and synchronization mechanism to operate transparently across multiple processors.

- Global Shared Memory: tasks can dynamically create and attach to shared memory on any processor, and access that shared memory. Because of the inter-kernel communication in Chimera, all the VMEbus address offsets can be calculated automatically during the initialization of the shared memory, and all operations are transparent to which processor the segment is actually on. In VxWorks, addresses of shared memory segments must be hard-coded into user's code, with the offsets (with are RTPU dependent) also added manually. This makes the code non-portable, and difficult to maintain if the hardware configurations are changed, as each hardware change requires a modification of the source code. The only reason this is available in VxWorks is because the VMEbus allows it; i.e. VxWorks makes no special provisions for global shared memory. In Chimera, no such modifications are required, and the same executable can be run even if the base addresses of any of the RTPUs in the system are changed.

- Spin Locks: This is a front end to the lowest-level of interprocessor communication using the atomic test-and-set (TAS) instructions. VxWorks lets you use the TAS instructions, but do not provide a convenient front-end for using them and automatically timing out if the lock cannot be obtained.

- Remote Semaphores: In addition to local semaphores, Chimera provides remote semaphores which allow tasks on all RTPUs to use a semaphore. VxWorks has only local semaphores; interrupts must explicitly be sent by the user if multiprocessor synchronization is required.

- Prioritized Message Passing: Chimera's message passing is multiprocessor. Any task on any RTPU can send or receive a message to/from any queue. Messages can be retrieved either first-in-first-out, last-in-first-out, or highest-priority-first. In VxWorks, message passing is local to an RTPU. To send a message to a remote RTPU, either sockets must be used, or a communications layer must implement the message passing on top of shared memory with interrupts for signaling. It is not clear from VxWorks literature, but queues might only be first-in-first-out.

- Global State Variable Table: Chimera provides this communication mechanism for predictable real-time communication and making efficient use of the VMEbus bandwidth. One global table exists, and every task which attaches to the table has a local copy of the parts of the table it needs. Updates of the local and global tables are done periodically and predictably. This is a 'must' communication mechanism for developing reconfigurable software. VxWorks does not provide anything comparable.

- Subsystem Multiprocessor Task Control: Chimera allows a task on one RTPU to control tasks (i.e. spawn, block, etc.) on multiple RTPUs. This is another feature required when developing reconfigurable software which must execute on multiple processors. VxWorks does not provide anything comparable.

- Triple Buffer Communication Mechanism: This predictable real-time communication mechanism allows Chimera to communicate with other subsystems which may or may not be running Chimera, or to communicate with intelligent memory mapped I/O devices or special purpose processors. The code for the mechanism is extremely portable, so that it can be compiled to execute on any non-Chimera platform. VxWorks does not provide any such mechanisms.

- Host Workstation Interface: Chimera allows processes on the host workstation to attach to the real-time environment by appearing to the RTPUs as just another RTPU. This allows communication between the host and RTPUs to occur in the same transparent manner that RTPUs communicate with each other. For example, a graphics application on the host can display real-time data by reading from a shared memory segment on one of the RTPUs. In VxWorks, any host-to-RTPU communication must be done through the ethernet, which is slow, less predictable, and much more difficult to program.

- Transparent Host Procedure Calls: RTPUs can execute remote procedure calls on the host workstation, either to execute some non-real-time code or to obtain information that is otherwise not available on the RTPUs. For example, RTPUs have a physical clock, but they don't have a time-of-day clock. So calling the function 'gettimeofday()' is automatically transfered into a remote procedure call, and the time-of-day from the host is returned. An RTPU can also execute programs like 'ls', 'emacs', etc. These programs "appear" to run on the RTPU, but in fact execute on the host workstation, but the results are sent to the RTPU's 'stdout', and input is received from the RTPU's 'stdin'.

- Special Purpose Processors (SPP): Chimera treats special purpose processors, such as floating point

accelerators, LISP machines, and image processors, as slaves, which execute a short, efficient, non-preemptive executive. A programming running under Chimera can then execute a procedure remotely. The SPP drivers automatically handle the communication and synchronization between the RTPU and the SPP. VxWorks has no equivalent interfaces.

- Interrupts: Chimera provides a hardware indepedent interface for generating and handling both VMEbus and mailbox interrupts. Chimera also gives you 256 mailbox interrupts per RTPU, even if the RTPU only supports one or two mailbox interrupts (as is the case with most RTPUs). VxWorks does not provide any special support for interrupts; you are at the mercy of whatever the hardware gives you, and must program the hardware using non-portable code.

Q: How does the hardware support compare between Chimera and VxWorks?

Chimera has concentrated on providing as much functionality as possible along a single line of RTPUs, initially. However nothing precludes poviding support for multiple RTPU's. We expect that this will happen in due time. At present, though, VxWorks supports several different models of RTPUs.

Chimera provides several device drivers for off-the-shelf I/O hardware, such as DAC, ADC, parallel I/O and serial I/O. VxWorks does not provide such drivers.

Chimera also provides device drivers for some special purpose processors, allowing user's to access the SPP's through the Chimera generic SPP interface. VxWorks does not have any such drivers. Several SPP hardware vendors do sell VxWorks drivers; however, those drivers always have custom interfaces, thus making them non-portable with the interfaces of other SPPs.

Q: How does the technical support for Chimera software compare to the support given for VxWorks by Wind River Systems?

Chimera II did not have any official support; however, those using Chimera II can usually get an answer or bug fixes within a couple of days directly from the authors. With Chimera 3.0, similar support will be available from the authors. However, when the software is commercialized, it will be possible to buy a software support contract, in which case there will be official support.

Q: What is the performance of Chimera vs. other RTOS?

A: The biggest strength of Chimera is in all the added features which are not available in other real-time operating systems (RTOS). However, despite the additional functionality, Chimera's performance is at par with VxWorks and other RTOS. Some benchmarks for real-time kernels were published comparing a few commercial RTOS (A. Topper, "A computing architecture for a multiple robot controller, M.S. Thesis, McGill University). Those same benchmarks were executed on Chimera, following are the results:

[NOTE: The authors are providing these results without any explicit or implied warranties.]

Test name	pSOS+	VRTX32	LynxOS	VxWorks	Chimera
Create Delete Task	[illegible]	[illegible]	--	[illegible]	[illegible]
Ping Suspend Resume	[illegible]	[illegible]		[illegible]	[illegible]
Ping Semaphore	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.