

Virtualization: Keeping Embedded Software safe and Secure in an Unsafe World

Robert Day provides some advice on techniques for software migration in embedded software platforms, the impact of security policies, and how to maintain real-time performance and determinism in your apps.



Author: Robert Day, LynuxWorks™, Inc.

Traditional military embedded systems have relied upon the physical separation of devices to ensure that information and communications at different levels of security is not compromised.

However, as in commercial systems, with the increase in performance of microprocessors and the desire to reduce the size, weight and power (SWaP) in confined spaces, there is a drive within modern military programs to consolidate hardware platforms.

This means that systems running at different levels of security will now be running on hardware platforms where resources (processor, memory, disks, devices and screens) are shared.

A new methodology is therefore needed to separate these systems and the information that they contain in a secure way, without compromising performance, and offering a cost-effective way to migrate applications from the legacy systems.

Another interesting shift is the desire to bring Commercial Off The Shelf (COTS) applications and operating systems into this next generation of military systems. These OSes and applications will generally be operating at an unclassified level of security, but will need to co-reside on the same physical platforms as the applications and OSes that will be dealing with secret or top-secret information.

The solution to these requirements is found in a collection of technologies that are integrated together and provided by embedded software and hardware companies that service the defense market.

To enable the secure sharing of applications over multiple security domains on the same physical hardware, a separation kernel that is based around a Multiple Independent Levels of Security (MILS) architecture can be used.

A separation kernel is a small, lightweight operating system that is the lowest-level connection to the processor. It provides secure partitions in which applications can run, where each partition is given access to both physical and virtual resources that the other partitions cannot access.

The separation kernel itself does not offer traditional OS features such as disk or network access, but it does manage scheduling and memory functions, as well as any physical devices (disks, graphics, network etc) available to each partition.

It also enforces security policies between the partitions, defining the allowed information flows between the different security levels. So, a user in a secret partition can access unclassified information, but has no access to the top-secret partition.

In the defense world, this small separation kernel is the foundation of high-assurance systems, offering security policy enforcement and strict partitioning. Also using a

specific protection profile, the separation kernel can be taken to the highest level of Common Criteria evaluation (currently EAL 7), which is used in systems that require the strictest security across multiple levels of clearance all running on the same physical hardware.

Many separation kernels are derived from partitioned operating systems by removing OS functionality and adding security features. However, to achieve the highest levels of evaluation, the software must also be proven secure by using formal-methods analysis, which is best achieved by designing the separation kernel from the ground up.

With the separation kernel being kept small and efficient, it lacks many of the features found in traditional operating systems. So, for military systems that require complex graphics or networking functionality, another component is required to enable applications and operating systems to also exist in this secure environment.

This next component is called a hypervisor, and is a piece of software that allows different guest operating systems to reside on a single hardware platform by offering a virtualized environment for the guests to run on.

This technology has been traditionally used in the enterprise or data-center world to allow the IT departments to run all their required applications across

multiple versions of server-based operating systems, and is now starting to gain traction in the embedded world.

Used in tandem, a hypervisor and a separation kernel allow multiple operating systems to run on the same hardware platform, while maintaining secure separation between them.

The requirement of modern military systems to use traditional desktop OSes has traditionally been a problem if security (and especially if multiple levels of security) is required.

With the introduction of a secure separation kernel and hypervisor, the traditional desktop OSs and applications can be run in their own unclassified partition, thus allowing for the functionality of a known user interface and applications, without compromising the security of the rest of the system.

Anything that enters into the desktop partition (such as targeted malware attacks) cannot breach the secure separation kernel and hence will be contained in the unclassified part of the system. The secure OSes and applications run in their own secure partitions and the separation kernel maintains the strict security policies to allow or inhibit communication and interaction between them.

Hypervisor technology can often offer different schemes for the virtualization of guest operating systems. A para-virtualized guest OS is modified to work more closely with the underlying hypervisor, and is typically the scheme that is used to allow hypervisors to operate on hardware with no virtualization support.

It can also offer better performance on processors with virtualization support as it is optimizing how the OS and the hypervisor will work in concert with one another and with the underlying processor.

This software partitioning and hypervisor virtualization helps in the consolidation of hardware and the reduction of SWaP, which is of particular interest in many military scenarios. By running separate systems in their own partitions, and allowing for different OSs and applications to be run in those partitions, there can be a true consolidation of physically separate systems to a single physical piece of hardware.

The use of Intel® multicore, virtualized processors allows the merging of a Windows® or Linux® desktop system with a more traditional RTOS system, but allows the same performance and functionality of applications as if they were still running on their own dedicated hardware platforms.

An additional feature that is very compelling in regard to this approach is virtual networking. Here, the guest OSs and applications can communicate virtually with other guest OSs and applications, even though they are residing in separate partitions.

The virtual network looks to the applications as though it is a real network port, and so these applications can communicate as if they were two physically separate networked devices, even though the communication is internal. A secure separation kernel can also enforce security policies to this virtual networking, and dictate which partitions can communicate with each other and in which direction.

This gives a secure partitioned environment with the ability to run multiple guest OSs and applications separated from one another on the same hardware. To allow near-native performance while maintaining real-time determinism and security, hardware virtualization support for both execution and memory can be utilized by the separation kernel and hypervisor.

For example, independent studies performed on the LynxSecure separation kernel and hypervisor have shown that running benchmark applications on a virtualized Linux OS yields less than a 5 percent performance degradation as compared to the same applications running on a native implementation of the same Linux on the same hardware.



Robert Day is Vice President, Marketing at LynuxWorks, Inc. and has more than 20 years of experience in the embedded industry. Based in San José, California, Robert is a graduate of the University of Brighton, England, where he earned a Bachelor of Science degree in computer science.



1.800.255.5969



LynuxWorks, Inc.
855 Embedded Way
San José, CA 95138-1018
408.979.3900
408.979.3920 fax
www.lynuxworks.com

LynuxWorks Europe
50 Broadway
London SW1H 0RG
United Kingdom
+44 208 906 9506
+44 208 906 2338 fax

©2011 LynuxWorks, Inc. LynuxWorks and the LynuxWorks logo are trademarks, and LynxOS is a registered trademark of LynuxWorks, Inc. Linux is a registered trademark of Linus Torvalds. All other trademarks are the trademarks and registered trademarks of their respective owners.

All rights reserved. Printed in the USA.