

Be my guest to find out what I am!

By David Kleidermacher, Green Hills Software

Virtualization can provide compelling efficiency and flexibility advantages for computing. However, instead of reinventing the wheel with a hypervisor, the microkernel operating system-based approach provides improvements in resource management and security. Want to add Linux or Windows to that? Be my guest!

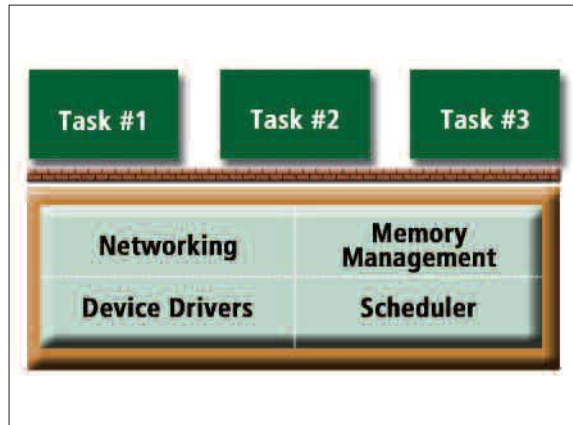


Figure 1. What am I?

Let's play a game: can you guess what I am?

- I schedule tasks on a computer.*
- I partition memory and disk space, ensuring that unrelated tasks are unable to steal memory or corrupt each other.*
- I manage peripherals, providing interrupt handling and device driver services.*
- I provide efficient interprocess communications between tasks.*
- I am trusted to do all these things in mission-critical environments, ensuring high availability, reliability, and security.*
- Figure 1 shows a picture of me.*
- Any guesses?*
- Are you thinking of an operating system?*
- Wrong! I'm a hypervisor.*

That's right, a hypervisor. Hypervisors are like operating systems whose managed tasks are guest operating systems. Of course, the hypervisor must also incorporate system virtualization logic that enables a general purpose operating system to believe it is executing on hardware (figure 2). Someone forgot to tell the hypervisor vendors that there already exists software that is really, really good at providing the scheduling, memory management, IPC, and device driver services needed by system virtualization: real-time operating systems. Sadly, hypervisor vendors have tried to reinvent the wheel. But as so often happens when wheels are reinvented, hypervisor manufacturers have yet to learn the hard-fought lessons that operating system technology has assimilated across

decades, vendors, and industries. Let's look at just a few examples. Some hypervisors statically dedicate a virtual machine to each core in a multi-core system. Symmetric multiprocessing (SMP) capable operating systems, on the other hand, are very good at dynamically scheduling virtual machines across cores. While an SMP operating system will allow the developer to bind a virtual machine to a core, it is almost always better to let the operating system perform the scheduling. An SMP operating system uses natural core affinity to reduce virtual machine migrations while still permitting virtual machines to float across cores as needed.

Many hypervisors lack support for power management. Operating systems are really good at monitoring CPU utilization, thermal state, and power draw, and using these inputs to make management decisions that ensure optimal battery life and minimal power consumption. For example, an operating system knows how to drop the CPU frequency and voltage and take advantage of sleep states when there is not a lot of work to do. Consider the management of two virtual machines on a dual-core processor.

An operating system can use its power-sensing intelligence to determine when both virtual machines are not very busy and can be migrated onto a single core where they are time-sliced, enabling the second core to be put to sleep. Security is one area where hypervisors have been seriously under fire. Entrusted to run on the

bare metal, hypervisors managing virtual machine workloads must ensure that those workloads are kept securely and reliably isolated, free from denial of service attacks, and free of software vulnerabilities that could allow a guest operating system or its applications to commandeer the system via the hypervisor. In 2006, the SubVirt project demonstrated hypervisor rootkits that subverted both VMware and VirtualPC. The BluePill project took hypervisor rootkits a step further by demonstrating a malware payload that was itself a hypervisor that could be installed on-the-fly, beneath a natively running Windows operating system. Tavis Ormandy performed an empirical study of hypervisor vulnerabilities. The researchers generated random I/O activity into the hypervisor, attempting to trigger crashes or other anomalous behavior.

The project discovered vulnerabilities in QEMU, VMware Workstation and Server, Bochs, and a pair of unnamed proprietary hypervisor products. Clearly, the risk of an escape from the virtualization layer, exposing all guests, is very real.

This is particularly true of hypervisors characterized by monolithic code bases. As one analyst has said, "Virtualization is essentially a new operating system . . . , and it enables an intimate interaction between underlying hardware and the environment. The potential for messing things up is significant." At the 2008 Black Hat conference, security researcher Joanna Rutkowska

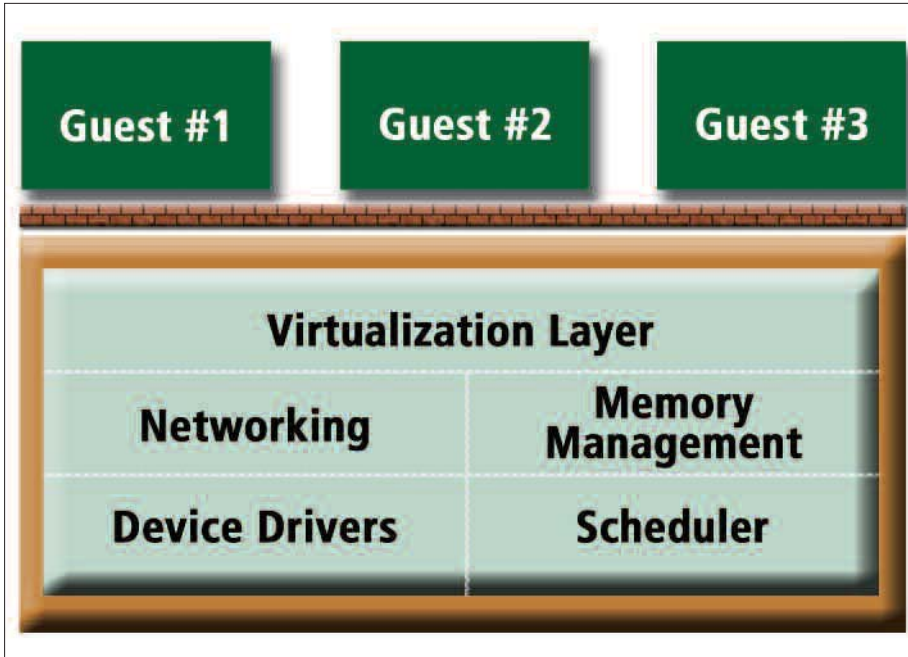


Figure 2. Traditional hypervisor architecture

and her team presented their findings of a brief research project to locate vulnerabilities in the Xen hypervisor. One hypothesis was that Xen would be less likely to have serious vulnerabil-

ities, as compared to VMware and Microsoft Hyper-V, due to the fact that Xen is an open source technology and therefore benefits from the many-eyes exposure of the code base. The

Rutkowka team discovered three different and fully exploitable vulnerabilities that the researchers used to commandeer the computer by way of the hypervisor. Ironically, one of these attacks took advantage of a buffer overflow defect in the Xen Flask layer. Flask is a security framework that is the same one used in SELinux. It was added to Xen to improve security.

A few products have taken a different approach to system virtualization, adding hypervisor-like capabilities to operating systems, thereby taking advantage of their preexisting resource management prowess. One example is Integrity from Green Hills Software which employs a microkernel with virtualization software relegated to user-mode processes, one per virtual machine. Integrity also solves the security challenge and is the first operating system certified to EAL 6+ under the International Common Criteria (ISO 15408) security standard, meeting what the US National Security Agency deems as high robustness: protection of high value information against determined and sophisticated attackers.

This same operating system is used in NSA-approved cryptographic communications devices, avionics systems that control passenger and military jets, life-critical medical systems,

secure financial transaction systems, and a wide variety of other safety and security-critical systems. Its proven security policies ensure that virtual machines are securely isolated from each other and that the system is immune to denial of service attacks and other vulnerabilities. In addition to its resource management strengths, the microkernel-based approach provides another major benefit over traditional hypervisors. The operating system provides a full-featured applications programming interface (API) and software development kit (SDK), enabling the creation and deployment of secure and/or real-time applications that cannot be trusted to run on a guest. For example, firewalls, databases, and cryptographic subsystems can be deployed alongside but securely separated from general purpose operating environments such as Windows or Linux. The combination of virtualized and native applications results in a powerful hybrid operating environment, as shown in figure 3, for the deployment of highly secure yet richly functional systems. In the following section, we shall discuss how this hybrid architecture is being applied in emerging embedded applications.

The use of virtualization outside traditional enterprise PC and server markets is nascent, and yet presents a significant opportunity. In this section, we shall discuss a sample of emerging applications with significant promise. Telecom Blade Consolidation: Virtualization enables multiple embedded operating systems, such as Linux and VxWorks, to execute on a single telecom computer. In addition, the microkernel-based virtualization architecture enables hard real-time applications to execute natively. Thus, control plane and data plane applications, typically requiring multiple blades, can be consolidated. Telecom consolidation provides the same sort of size, weight, power, and cost efficiencies that enterprise servers have enjoyed with VMware.

In-Vehicle Infotainment: Demand for more advanced infotainment systems is growing rapidly. In addition to theater-quality audio and video

and GPS navigation, wireless networking and other office technologies are making their way into the car. Despite this increasing complexity, passenger expectations for instant-on and high availability remain. At the same time, automobile systems designers must always struggle to keep cost, weight, power, and component size to a minimum. Although we expect desktop operating systems to crash occasionally, automobile passengers expect the radio and other traditional head-unit components never to fail. In fact, a failure in one of these components is liable to cause an expensive (for the automobile manufacturer) visit to the repair shop. Even worse, a severe design flaw in one of these systems may result in a recall that wipes out the profit on an entire model year of cars. Exacerbating the reliability problem is a new generation of security threats: bringing the internet into the car exposes it to all the viruses and worms that target networked Windows-based computers.

The currently deployed solution, found on select high-end automobiles, is to divide the infotainment system onto two independent hardware platforms, placing the high-reliability, real-time components onto a computer running a real-time operating system, and the Windows component on a separate PC. This solution is highly undesirable, however, because of the need to tightly constrain component cost, size, power, and weight within the automobile.

The microkernel virtualization architecture provides an ideal solution. Head unit applications running under control of the real-time kernel are guaranteed to perform flawlessly. Because the real-time kernel is optimized for the extremely fast boot times required by automotive systems, instant-on requirements are met. Multiple instances of Windows, powered by multiple instances of the virtual machine, can run simultaneously on the same computer. In the back seat, each passenger has a private video monitor. One passenger could even reboot Windows without affecting the email session of the second passenger.

Next Generation Mobile Devices: Using the hybrid virtualization architecture, mobile device manufacturers and service providers can leverage traditional operating systems and software, such as the Linux and Symbian, while guaranteeing the integrity, availability, and confidentiality of critical applications and information. We bring our mobile devices wherever we go. Ultimately, consumers would like to use mobile devices as the key to the automobile, a smart card for safe internet banking, a virtual credit card for retail payments, a ticket for public transportation, and a driver license and/or passport. There is a compelling world of personal digital convenience just over the horizon.

The lack of a high security operating environment, however, precludes these applications from reaching the level of trust that consumers demand. High assurance secure platform technology enables this level of trust. Furthermore, security applications can be incorporated alongside the familiar mobile multimedia operating system on one chip (SoC), saving precious power and production cost. With secure virtualization technology, the mobile device can host multiple instances of mobile operating systems. For example, the device can incorporate one instance of Linux that the consumer uses for the phone function, e-mail, and other critical applications. A second instance of Linux can be used specifically for browsing the internet. No matter how badly the internet instance is compromised with viruses and Trojans, the malware cannot affect the critical instance of the user. The only way for files to be moved from the internet domain to the critical user domain is by using a secure cut and paste mechanism that requires human user interaction and cannot be spoofed or commandeered. A simple key sequence or icon is used to switch between the two Linux interfaces.

Secure virtualization can also be used to provide a mobile device with multiple operating system personalities, enabling service providers, phone manufacturers, and consumers to provide and enjoy a choice of environments on a single device. Furthermore, by virtualizing the user environment, personas (personal data, settings, and so on) can be easily migrated across devices, in much the same way that virtual machines are migrated for service provisioning in the data center. In a recent article discussing the growth of mobile devices in corporate environments, USA Today stated that mobile devices represent the most porous piece of the IT infrastructure. The same problems that plague desktops and servers are afflicting mobile devices. Secure operating systems and virtualization technology provide a solution to the demand for enhanced security in the resource-constrained environment of portable consumer devices. ■

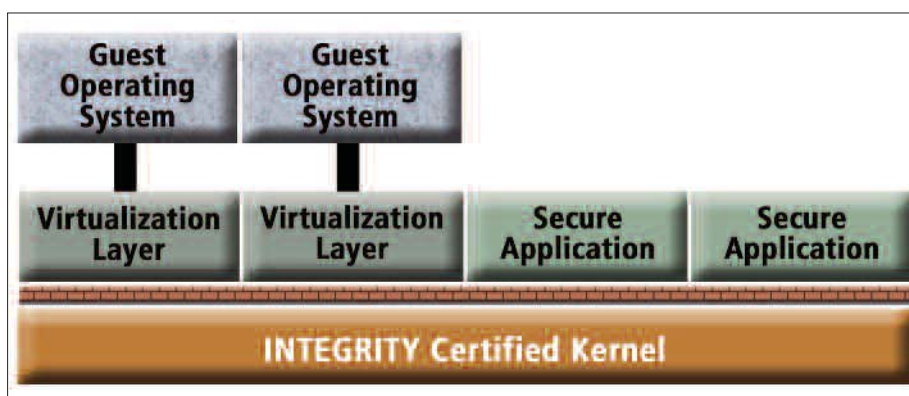


Figure 3. Microkernel-based virtualization architecture