A SURVEY ON REAL-TIME ISSUES IN EMBEDDED SYSTEMS VIRTUALIZATION

A.SAI HRUDAI
Students
Sreenidhi institute of Science and Technology,
Hyderabad, India
sanjuhrudai@gmail.com

Abstract:

Virtualization has won high-quality attractiveness inside the server and cloud computing arena. In current years, it has additionally been broadly applied to actual-time embedded systems with stringent timing constraints. We present a comprehensive survey on real-time issues in virtualization for embedded systems, protecting famous virtualization structures including KVM, Xen, L4 and others.

Keywords:

Virtualization; Embedded Systems; Real-Time Scheduling

Introduction:

Platform virtualization refers to the introduction modern-day digital Machines (VMs), additionally called domain names, visitor OSes, or partitions, jogging on the physical machine controlled via a virtual machine monitor (VMM), also called a hypervisor. Virtualization era permits concurrent execution modern a couple of VMs at the equal hardware (single or multicore) processor. Virtualization era has been broadly implemented inside the organisation and cloud computing space. In recent years, it has been state-of-the-art broadly deployed in the embedded systems area, consisting of avionics structures, industrial automation, mobile telephones, and many others. in comparison to the traditional utility area ultra-modern agency structures, virtualization in embedded systems should location robust emphasis on problems like actual-time overall performance, protection and dependability, and so forth. A VMM can run either at the hardware immediately (known as bare-metallic, or kind-1 virtualization), or run on pinnacle modern a number operating machine (referred to as hosted, or type-2 virtualization). any other way to classify platform-level virtualization technologies is complete virtualization vs paravirtualization. full virtualization lets in the guest OS to run on the VMM without any change, at the same time as paravirtualization calls for the guest OS to be changed by means of including hypercalls into the VMM. consultant type-2, full virtualization answers consist of KVM, VirtualBox, Microslatestt digital laptop, VMWare laptop; representative type-1, paravirtualization answers include Xen, L4, VMWare ESX. There are a few studies attempts at building kind-1, complete virtualization answers, e.g., Kinebuchi et al. implemented this sort of answer by means of porting the QEMU machine emulator to run as an utility on L4Ka::Pistachio microkernel; in turn, unmodified visitor OS can run on pinnacle modern-day QEMU; Schild et al.used Intel VT-d HW extensions to run unmodified visitor OS on L4. There also are type-2, para-virtualization solutions, e.g., VMWare MVP (cellular Virtualization Platform), in addition to some attempts at including para-virtualization functions to kind-2 virtualization systems to enhance performance, e.g., assignment-grain scheduling in KVM. in view that embedded structures brand newten have stringent timing and performance constraints, virtualization for embedded structures ought to address actual-time problems. We consciousness on realtime problems in virtualization for embedded systems, and miss sure topics which might be greater relevant to the server application area latest space constrains, consisting of: strength and strength-aware scheduling, dynamic adaptive scheduling, multicore scheduling on high-quit NUMA (Non-Uniform memory get entry to) machines, nested virtualization, and so on. in addition, we cognizance on current traits in this discipline, as opposed to imparting a ancient attitude.

This paper is based as follows: we talk tough real-time virtualization solutions for safety-vital systems in segment 2; Xen-based answers in phase 3, inclusive of hard real-time and smooth actual-time extensions to Xen; KVM-based totally answers in phase 4; micro-kernel primarily based answers represented by L4 in phase 5; different virtualization frameworks for embedded systems in section 6; OS virtualization in section 7; mission-grain scheduling in segment eight; the Lock-Holder Preemption hassle in section 9; and in the end, a brief end in segment 10. (The topics cutting-edge Sections 8 and 9 are pass-slicing troubles A 49a2d564f1275e1c4e633abc331547 db Survey on real-Time troubles in Embedded structures Virtualization that aren't precise to any virtualization approach, but we agree with they may be contemporary sufficient importance to commit separate sections to cowl them.

Hard Real-Time Virtualization for Safety-Critical Systems:

The ARINC 653 fashionable defines a software program structure for spatial and temporal partitioning designed for safety-essential IMA (incorporated Modular Avionics) applications. It defines services such as partition control, system management, time management, inter and intra-partition verbal exchange. in the ARINC 653 specification for device partitioning and scheduling, each partition (digital device) is allocated a time slice, and partitioned are scheduled in a TDMA (Time division multiple access) way. A related fashionable is the a couple of impartial levels of safety (MILS) structure, an enabling architecture for developing security-essential programs conforming to the common standards safety assessment. The MILS structure defines four conceptual layers of separation: separation kernel and hardware; middleware services; relied on programs; allotted communications. Authors from Lockheed Martin offered a feasibility assessment closer to applying the Microkernel Hypervisor architecture to enable virtualization for a consultant set of avionics packages requiring a couple of guest OS environments, inclusive of a aggregate of safety-important and non-safety-important guest OSes. several commercial RTOS merchandise comply with the ARINC 653 trendy, inclusive of LynuxWorks LynxOS-178, green Hills INTEGRITY-178B, Wind River VxWorks, BAE structures CsLEOS, and DDC-I DEOS, etc. Many vendors additionally provide business virtualization products for protection-important structures, many by adapting present RTOS products. as an instance, LynuxWorks carried out a virtualization layer to host their LynxOS product, referred to as the LynxSecure hypervisor that supports MILS and ARINC 653. WindRiverprovides a hypervisor product for both its VxWorks MILS and VxWorks 653 platforms, together with help for multicore. other comparable products encompass: Greenhills INTEGRITY MultiVisor , RealTime structures GmbH Hypervisor, Tenasys eVM for windows national instruments actual-Time Hyper Hypervisor, Open Synergy COQOS , Enea Hypervisor , SysGO PikeOS, IBV Automation GmbH QWin and so on. VanderLeest , from a enterprise named DornerWorks, took a extraordinary approach with the aid of adapting the open-supply Xen hypervisor to enforce the ARINC 653 widespread. XtratuM is a kind-1 hypervisor concentrated on protection essential avionics embedded structures. It runs on LEON3 SPARC V8 processor, a extensively used CPU in area applications.

It capabilities temporal and spatial separation, green scheduling mechanism, low footprint with minimum computational overhead, green context switch of the partitions (domains), deterministic hypervisor device calls. Zamorano et al. ported the Ada-based totally Open Ravenscar Kernel (ORK+) to run as a partition on XtratuM, forming a software program platform conforming to the ARINC 653 fashionable. Campagna et al.applied a dual-redundancy system on XtratuM to tolerate brief faults like single-occasion-disappointed, common on the excessive-radiation space surroundings. three walls are accomplished concurrently, two of them run identical copies of the application software program, and the 0.33 exams consistency in their outputs. The OVERSEE (Open Vehicular comfortable Platform) Projectaims to convey the avionics popular to automobile structures by using porting FreeOSEK, an

OSEK/VDX-compliant RTOS, as a paravirtualized visitor OS strolling on top of the XtratuM hypervisor. while maximum ARINC-653 compliant virtualization answers are primarily based on para-virtualization, Han et al. offered an implementation of ARINC 653 primarily based on type-2, complete virtualization architectures, together with VM-Ware and VirtualBox. subsequent, we briefly mention a few virtualization solutions for protection-crucial systems that are not specifically designed for the avionics area, for this reason do no longer comply with the ARINC-653 popular. Authors from Indian Institute of era developed SParK (protection Partition Kernel), a kind-1 para-virtualization answer designed for safety-important systems; an open-source RTOS uC/ OS-II and a customized model of saRTL (stand-alone RT Linux) are ported as guest OSes on SParK. Authors from CEA (Atomic strength commission), France evolved PharOS, a dependable RTOS designed for car manage systems presenting temporal and spatial isolation inside the presence of a mixed workload of each time-precipitated and event-induced obligations. They adapted Trampoline, an OSEK/VDX-compliant RTOS, as a paravirtualized visitor OS jogging on top of PharOS host OS to form a type-2 virtualization architecture. To make sure temporal predictability, Trampoline is run as a timetriggered undertaking inside PharOS. Authors from PUCRS, Brazil [25] evolved digital-Hellfire Hypervisor, a type-1 virtualization machine primarily based at the microkernel HellfireOS featuring spatial and temporal isolation for safety-vital packages. The target HW platform is HERMES network-on-Chip with MIPS-like processing factors.

Xen-Based Solutions :

Cherkasova et al. introduced and evaluated three CPU schedulers in Xen: BVT (Borrowed virtual Time), SEDF (easy Earliest deadline First), and credit score. since the BVT scheduler is now deprecated, we handiest discuss credit and SEDF here. The default scheduling set of rules in Xen is the credit SciRes. JSEA A 49a2d564f1275e1c4e633abc331547 db Survey on actual-Time troubles in Embedded structures Virtualization Scheduler: it implements a proportional-share scheduling approach in which a user can adjust the CPU percentage for every VM.

It also capabilities automatic workload balancing cutting-edge digital CPUs (vCPUs) throughout bodily cores (pCPUs) on a multicore processor. This set of rules ensures that no pCPU will idle when there exists a runnable vCPU inside the device. every VM is associated with a weight and a cap. when the cap is 0, VM can receive more CPU time unused through different VMs, i.e., WC (work-preserving) mode; while the cap is nonzero (expressed as a percentage), it limits the amount present day CPU time given to a VM to now not exceed the cap, i.e., NWC (Non-work-retaining) mode. by default the credits latest all runnable VMs are recalculated in periods contemporary 30 ms in share to every VM's weight parameter, and the scheduling time slice is 10 ms, i.e., the credit modern the running digital CPU is decreased every 10 ms. in the SEDF scheduler, every domain can specify a decrease certain on the CPU reservations that it requests by way of specifying a tuple contemporary (slice, duration), in order that the VM will acquire at least slice time gadgets in every length time gadgets. A Boolean flag shows whether the VM is eligible to get hold of more CPU time.

If genuine, then it is WC (WorkConserving) mode, and any available slack time is sent in a fair manner after all of the runnable VMs have received their unique slices. in contrast to the credit score Scheduler, SEDF is a partitioned scheduling algorithm that doesn't permit VM migration across a couple of cores, therefore there's no global workload balancing on multicore processors. Masrur et al. offered improvements to Xen's SEDF scheduler in order that a site can make use of its entire price range (slice) within its period even though it blocks for I/O earlier than the use of up its entire slice (inside the original SEDF scheduler, the unused finances is misplaced as soon as a challenge blocks). in addition, sure important domain names can be specific as real-time domains and accept better constant-priority than other domain names scheduled with SEDF. One limitation is that each real-time area is restrained to contain a single real-time assignment. Masrur et al. carried out the Compositional Scheduling Framework in Xen-ARM. Jeong et al. evolved PARFAIT, a hierarchical scheduling framework in Xen-ARM. At the lowest level (near the HW), SEDF is used to provide CPU bandwidth ensures to Domain0 and actual-time VMs; at the higher level, BVT (Borrowed digital Time) is used to agenda all non-realtime VMs to provide fair distribution brand new CPU time among them, via mapping all vCPUs in the non actual-time VMs to a single summary vCPU scheduled through the underlying SEDF scheduler. Lee et al. , Xi et al, Yoo et al., Jeong et al. handiest addressed CPU scheduling problems, however did not keep in mind I/O scheduling. Xen has a split-motive force architecture for dealing with I/O, wherein a special driving force domain (Domain0) contains the backend driving force, and person domains include the frontend motive force. physical interrupts are first handled by using the hypervisor, which then notifies the target visitor area with virtual interrupts, handled whilst the guest domain is scheduled via the hypervisor scheduler. Hong et al.stepped forward network I/O performance latest Xen-ARM via acting dynamic load balancing brand new interrupts many of the more than one cores by way of enhancing the ARM11 MPCore interrupt distributor. similarly, every guest area includes its very own local tool drivers, as opposed to putting all device drivers in Domain0. Yoo et al. [36] proposed numerous upgrades to the Xen credit Scheduler to enhance interrupt response time, consisting of: do now not deschedule the driving force area whilst it disables virtual interrupts; make the most ARM processor support for FIQ, which has better priority than regular IRQ, to assist actual-time I/O devices.

KVM-Based Solutions:

KVM is a type-2 virtualization answer that trendy Linux as the host OS, as a result any actual-time enhancements to the Linux host OS kernel at once translate into actual-time improvements to the KVM VMM. for example, realtime kernel patches like Ingo Milnar's PREEMPT_RT patch, can be applied to improve the actual-time overall performance and predictability trendy the Linux host OS. Kiszka [4] presented a few real-time upgrades to KVM. real-time guest VM threads are given real-time priorities, which include the main I/O thread and one or greater vCPU threads, at the rate brand new giving decrease priorities to threads in the host Linux kernel for various device-huge offerings. A paravirtualized scheduling interface became introduced to allow task-grain scheduling by way of introducing two hypercalls for the visitor VM to tell the host VMM about priority contemporary the presently walking venture in the VM, as well as when an interrupt handler is completed execution in the VM. creation of these hypercalls implies that the resulting machine is now not a strict fullvirtualization system because the conventional KVM. Zhang et al. offered actual-time enhancements to KVM with coexisting RTOS and GPOS visitors: giving the visitor RTOS vCPUs better precedence than GPOS vCPUs; use CPU shielding to dedicate one CPU center to the RTOS visitor and guard it from GPOS interrupts, by way of placing the CPU affinity parameter brand new each host OS procedures and interrupts. Experimental effects suggest that the RTOS interrupt response latencies are reduced. based at the paintings modern , Zuo et al. offered additional improvements via adding hypercalls that allow the visitor OS to boost priority today's its vCPU whilst a highpriority assignment is commenced, e.g., like an interrupt handler, and deboost it when the high-precedence challenge is finished. Cucinotta et al. presented the IRMOS scheduler, which implements a hard reservation version ultra-modern CBS (regular Bandwidth Server) on top modern-day the EDF scheduler in Linux via extending the Linux CGroup interface. while carried out within the context modern day KVM, inter-

VM scheduling is CBS/EDF, whilst intra-VM assignment scheduling is constant-priority. while Cucinotta et al. best addressed SciRes. JSEA A 49a2d564f1275e1c4e633abc331547 db Survey on real-Time problems in Embedded systems Virtualization 281 CPU scheduling, Cucinotta et al. addressed I/O and networking troubles with the aid of grouping both visitor VM threads and interrupt handler threads in the KVM host OS kernel into the same CBS reservation, and over-provisioning the CPU budget assigned to every VM via an amount that is depending on the overall networking visitors executed by using VMs hosted on the same device. Cucinotta et al.stepped forward the CBS scheduler inside the KVM host OS for a mixed workload cutting-edge each compute-intensive and networkintensive workloads. similarly to the regular CPU CBS reservation contemporary (Q, P), in which the VM is guaranteed to get hold of budget Q time units modern CPU time out state-of-the-art P time devices, a spare CPU reservation present day (Qs, ps) is attached to the VM and dynamically activated upon a brand new packet arrival.

MicroKernel-Based Solutions:

L4 is a consultant microkernel working gadget, prolonged to be a type-1 virtualization structure. There are more than one active variants of L4. discern 1 suggests the evolution lineage of L4 editions. Authors from Avaya Labs implemented a par virtualized Android kernel, and ran it along a par virtualized Linux kernel from good enough Labs, L4Linux, on an ARM processor. Iqbal et al. presented an in depth comparison observe among microkernel-based approach, represented through L4, and hypervisor-primarily based approach, represented by means of Xen, to embedded virtualization, and concluded that L4-based totally strategies have higher overall performance and security residences. Gernot Heiser, founding father of OKL4, argued for microkernel-primarily based virtualization for embedded systems. overall performance assessment of OKL4 hypervisor on Beagle Board based on 500 MHz Cortex A8 ARMv7 processor, strolling netperf benchmark suggests that the TCP and UDP throughput degradation due to virtualization is most effective 3% - 4%. Bruns et al. and Lackorzynski et al. achieved overall performance assessment of L4Linux. the usage of L4/Fiasco as the hypervisor, L4Linux as the guest OS, an ARM-based Infineon XGOLD618 processor for mobile phones, Bruns et al. [60] in comparison thread context-switching times and interrupt latencies of L4Linux to those of a stand-by myself RTOS (FreeRTOS), and established that L4-primarily based gadget has fantastically small runtime overhead, however calls for considerably extra cache sources than an RTOS, with cache competition as the main perpetrator for overall performance degradations.

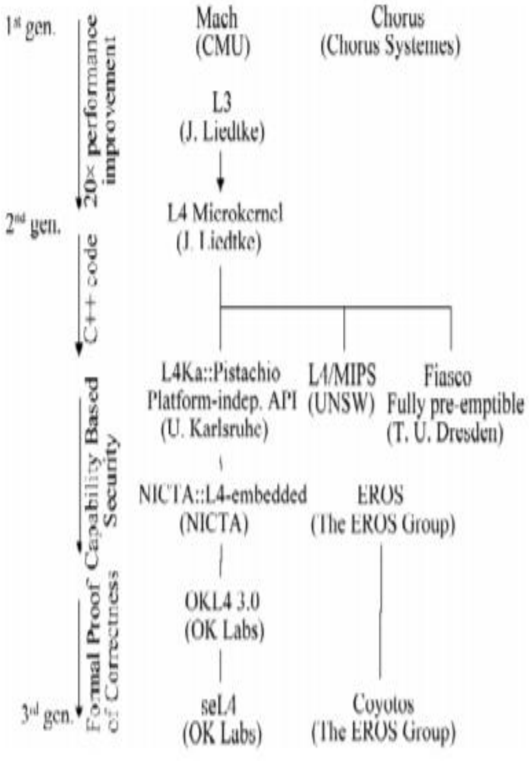
1st gen.
2nd gen.
3rd gen.
20× performance improvement
C++ code
Capability Based Security
Formal Proof of Correctness
Mach (CMU)
Chorus (Chorus Systemes)
L3 (J. Liedtke)
L4 Microkernel (J. Liedtke)
L4Ka::Pistachio Platform-indep. API (U. Karlsruhe)
L4/MIPS (UNSW)
Fiasco Fully pre-emptible (T. U. Dresden)
NICTA::L4-embedded (NICTA)
EROS (The EROS Group)
OKL4 3.0 (OK Labs)
seL4 (OK Labs)
Coyotos (The EROS Group)

Figure 1. L4 evolution lineage (from Acharya et al.)

Other Virtualization Frameworks for Embedded Systems:

Authors from Motorola argued for the use of type-1 hypervisors instead of kind-2 hypervisors for cell phone virtualization for their protection benefits due to a small TCB (relied on Computing Base). representative mobile virtualization solutions include MVP (mobile Virtualization Platform) from VMWare VLX from crimson Bend Xen-ARM from Samsung , and many others. L4, VLX and Xen-ARM are all type-1 hypervisors. VMWare MVP is a type-2 virtualization answer with Linux as both host and guest OSes implemented on ARMv7 processor. It adopts a lightweight para-virtualization technique:

1) the whole guest OS is run in CPU consumer-mode; maximum privileged instructions are treated via trap-and-emulate, e.g., privileged coprocessor access commands mcr and mrc; different touchy commands are replaced with hypercalls;

2) all gadgets are para-virtualized; especially, the paravirtualized TCP/IP is different from traditional para-virtualized networking via operating at the socket machine call level rather than at the tool interface degree:

when a guest VM calls a syscall to open a socket, the request is made immediately to the offload engine inside the host OS, which opens a socket and performs TCP/IP protocol processing in the host OS kernel. SPUMONE (software Processing Unit, Multiplexing ONE into two or extra) is a lightweight type-1 virtualization software program applied on a SH-4A processor with the intention of going for walks a GPOS (popular-reason OS) like Linux and an RTOS like TOPPERS at the same HW processor. it may either use fixedpriority scheduling, and always assign the RTOS higher precedence than GPOS, or undertake assignment-grain scheduling , in which each venture can be assigned a specific priority to permit extra quality-grained control, e.g., an important assignment within the

GPOS can be assigned a higher priority than a much less vital venture inside the RTOS. Inter-OS communication is achieved with IPI (Inter-Processor Interrupts) and shared memory. Aalto advanced DynOS SPUMONE, an extension to SPUMONE to enable runtime migration of visitor OSes to one of a kind cores on a 4-core RP1 processor from Renesas. Lin et al. supplied a redecorate of SPUMONE as a multikernel architecture. (Multikernel means that a separate reproduction of the kernel or VMM runs on every middle on a multicore processor.) each processor core has its private local memory. To gain better fault isolation, the VMM runs within the local memory, whilst the guest OSes run within the global shared major reminiscence; both run in kernel mode. To decorate security, Li et al. brought a small relied on OS referred to as xv6 to run a tracking carrier to discover any integrity violations of the big and untrusted Linux of OS.

OS Virtualization:

OS virtualization, additionally called field-based totally virtualization, lets in to partition an OS surroundings into more than one domain names with unbiased call spaces to reap a sure level of protection and safety between extraordinary domains. Examples include FreeBSD Jails , OpenVZ , Linux VServer , Linux packing containers LXC and Solaris Zones. the important thing distinction among OS virtualization and gadget-stage virtualization is that the former simplest has a unmarried replica of OS kernel at runtime shared among more than one domain names, while the latter has more than one OS kernels at runtime. As a result, OS virtualization is extra light-weight, but can most effective run a unmarried OS, e.g., Linux. commercial merchandise that incorporate OS virtualization era consist of: Parallels Virtuozzo boxes is a industrial version of the open-supply OpenVZ software program; the MontaVista automotive era Platform adopts Linux boxes in combination with SELinux (protection more desirable Linux) to run more than one Linux or Android OSes on pinnacle of MontaVista Linux host OS. aid manage and isolation in OS virtualization are often completed with the Linux kernel mechanism CGroup. for example, Linux VServer enforces CPU isolation by using masking a token bucket clear out (TBF) on top of the usual O(1) Linux CPU scheduler. each VM has a token bucket that accumulates incoming tokens at a special fee; every timer tick, the VM that owns the walking method is charged one token. The authors modified the TBF to offer honest sharing and/or work-keeping CPU reservations. CPU capability is effectively partitioned among two training of VMs: VMs with reservations get what they have reserved, and VMs with stocks cut up the unreserved CPU ability proportionally. For community I/O, the Hierarchical Token Bucket

(HTB) queuing area of the Linux site visitors manage facility is used to offer community bandwidth reservations and honest carrier among VMs. Disk I/O is managed using the standard Linux CFQ (completely-fair Queuing) I/O scheduler, which tries to divide the bandwidth of each block device pretty the various VMs appearing I/O to that device.

Conclusion:

on this paper, we've presented a comprehensive survey on actual-time troubles in embedded structures virtualization. we are hoping this newsletter can function a useful reference to researchers in this region

REFERENCES:

[1] Y. Kinebuchi, H. Koshimae and T. Nakajima, "Constructing Machine Emulator on Portable Microkernel," Proceedings of the 2007 ACM Symposium on Applied Computing, 11-15 March 2007, Seoul, 2007, pp. 1197- 1198. doi:10.1145/1244002.1244261

[2] H. Schild, A. Lackorzynski and A. Warg, "Faithful Virtualization on a Realtime Operating System," 11th RealTime Linux Workshop, Dresden, 28-30 September 2009.

[3] K. Barr, P. P. Bungale, S. Deasy, V. Gyuris, P. Hung, C. Newell, H. Tuch and B. Zoppis, "The VMware Mobile Virtualization Platform: Is That a Hypervisor in Your Pocket?" Operating Systems Review, Vol. 44, No. 4, 2101, pp. 124-135. doi:10.1145/1899928.1899945

[4] J. Kiszka, "Towards Linux as a Real-Time Hypervisor," 11th Real-Time Linux Workshop, Dresden, 28-30 September 2009.

[5] Airlines Electronic Engineering Committee, "ARINC 653—Avionics Application Software Standard Interface," 2003

[6] T. Gaska, B. Werner and D. Flagg, "Applying Virtualization to Avoinics Systems—The Integration Challenge," Proceedings of the 29th Digital Avionics Systems Conference of the IEEE/AIAA, Salt Lake City, 3-7 October 2010, pp. 1-19. doi:10.1109/DASC.2010.5655297

[7] Wind River Hypervisor Product Overview. http://www.windriver.com/products/hypervisor

[8] INTEGRITY Multivisor Datasheet. http://www.ghs.com

[9] Real-Time Systems GmbH. http://www.real-time-systems.com

[10] Tenasys eVM for Windows. http://www.tenasys.com/products/evm.php

[11] Quick Start Guide, NI Real-Time Hypervisor. Copyright © 2012 SciRes. JSEA A State-of-the-Art Survey on Real-Time Issues in Embedded Systems Virtualization 287 http://www.ni.com/pdf/manuals/375174b.pdf.

[12] OpenSynergy COQOS. http://www.opensynergy.com/en/Products/COQOS

[13] Enea Hypervisor. http://www.enea.com/software/products/hypervisor

[14] SysGO GmbH. http://www.sysgo.com

[15] S. H. VanderLeest, "ARINC 653 Hypervisor," Proceedings of 29th Digital Avionics Systems Conference of IEEE/AIAA, Salt Lake City, 3-7 October 2010, pp. 1-20. doi:10.1109/DASC.2010.5655298