Real-Time Operating Systems Case Study ~ LynxOS vs. VxWorks - Presentation Project for Course COP 4610

Technical Report · December 2017

DOI: 10.13140/RG.2.2.28636.44166

Lukas Garcia
Florida Atlantic University

Real-Time Operating Systems

Case Study

~

LynxOS vs. VxWorks

Lukas W. Garcia

December 5th, 2017

Presentation Project – COP 4610

TABLE OF CONTENTS

Introduction to Operating Systems.. 3

What is LynxOS & VxWorks?.. 5

LynxOS & VxWorks Architecture.. 9

Operating System Functions 15

Memory Security - Design Evaluation.. 20

References 22

Introduction to Operating Systems

What is an Operating System (OS)?

An operating system, a computer program that can support a computer's basic functions, and can provide services to other programs that can run on a computer. The computer can be any system like a home desktop, a mobile device, or an embedded hardware system that has functionalities where a user can manipulate it and be able to use the services that operating system provides. The user will have needs or wants that a computer will be able to handle, control, or direct through the services, or in other words, applications or programs, that the user can run on the computer. An operating system (OS) helps to write, maintain, and use these applications faster and simpler than having to backtrack through lines of code that a developer has written for the application. An example of this, will be the web browser in a desktop. The web browser is an application that is running within the operating system's environment.

Most operating systems usually allow for multiple applications to run at the same time. When more than one application is running at the same time as another, this is called multi-tasking. A unit within the operating system, called the scheduler, decides what programs to run, when that specific program should run, and in a way, provides the user to see that multiple programs are running at the same time. These programs seem to run at the same time, but is an "illusion of simultaneous execution by rapidly switching between each program."[1]

How an operating system is distinct from another OS, since there are multiple types, is by how the scheduler decides on which program gets to run first and so on. The scheduler in the operating system is what distinguishes a variant OS from one another. For example, the 'UNIX OS will make sure that each program gets a fair share of the CPU processing time, whereas the Microsoft Windows OS tries to make sure that the computer stays responsive for its user.'[2] Even though both examples are their own operating systems in their own functionality, both share the reason that they make sure the scheduler runs only for the OS that they are programmed for.

[1] Richard Barry, "What is an RTOS?", freeRTOS, Dec. 2, 2017, https://www.freertos.org/about-RTOS.html.
[2] Barry, https://www.freertos.org/about-RTOS.html.

What is a Real-Time Operating System (RTOS)?

Unlike a Microsoft Windows or the Unix operating systems, the scheduler for a Real-Time Operating System (RTOS) is created to give a predictable program execution pattern. This type of operating system if mostly found on embedded systems where usually the system has real-time requirements to meet. In other words, if an embedded system has a specific deadline that the program has, it must be finalized before the deadline is reached. The guarantee that the program can meet the deadline is based exclusively if the performance of the operating system's scheduler can be predicted. The prediction of the scheduler's performance is known as being "deterministic"[3]. To achieve determinism, the user or developer of a program must assign priority to every thread that needs to be executed. The scheduler than checks the priority to make sure which thread is going to be executed next.

There are two different specifications for Real-Time Operating Systems. The first specification is a Hard-Real-Time Operating System, in which if the program could not run specifically to meet the deadline that the scheduler expected, the operating system will "terminate the [program] with a failure."[4] An example of this would be like a machine that is waiting to scan a jar in a jar and bottle assembly line. If the jar's barcode does not meet the machine by a specific time, the machine will cause the assembly line to halt the assembly line. The other specification is a Soft-Real-Time Operating System, where if the program could still run, but it was not able to complete the task because there is data missing after running the instruction set. If the jar doesn't meet the barcode scanning machine, the machine will allow the assembly line to continue, but the "production output would be lower as objects [jars] failed to appear at their designated time. This causes the machine to be temporarily unproductive."[5]

The importance of an RTOS is that it is needed to respond to a set case of events in a timely manner. If these responses are missed or late, it is declared a failure. A failure would lead to catastrophic failures, and because RTOS' are mainly used in aeronautical applications, it would lead to the consequences in loss of human life.

[3] Barry, https://www.freertos.org/about-RTOS.html.

[4] Margaret Rouse, "Real-Time Operating System (RTOS)", SearchDataCenter, Dec. 2, 2017, http://searchdatacenter.techtarget.com/definition/real-time-operating-system.

[5] Rouse, http://searchdatacenter.techtarget.com/definition/real-time-operating-system.

What is LynxOS & VxWorks?

What is LynxOS?

Developed by Lynx Software Technologies, LynxOS is a Real-Time Operating System that is considered to be absolutely deterministic. It is based off the UNIX operating system, which is also based off Linux, and it confirms to the POSIX set of standards and makes sure to use the most concise of the embedded kernel footprint that is found most embedded systems today. POSIX is short for 'Portable Operating System Interface for uni-X' and are standards that were developed by the Institute of Electrical and Electronic Engineers (IEEE) to ease "the task of developing programs where the developer only has to write a program once to run all POSIX-compliant systems." [6] LynxOS uses POSIX-compliant APIs (Application Programming Interface) which provide symmetric multi-processing support so that the operating system can take full control of any multi-core processors.

Not only does LynxOS have a host of different APIs that allow for better control of the multi-core processors, but it does have an array of tools, debuggers, and cross-development support for multiple systems. The support alone for these multiple systems allows for I/O technology support, as well as to enable any state-of-the-art security features that may exist today. An application in LynxOS well also be able to rely on the operating system's real-time determinism feature. It is considered a foundational feature were a "predictable response is ensured even in the presence of heavy I/O as a result of the kernel's unique and highly optimized threading model."[7]

Where is LynxOS Found?

LynxOS can be found in a variety of modern application systems, including trades such as military, aeronautical, medical, and industry. The operating system even helps businesses that are based in office automation as they "benefit from […] security and networking improvements"[7] that are found in next-gen LynxOS architectures. The one industry that LynxOS

[6] Indiana University, "What is POSIX?", Dec. 2, 2017, https://kb.iu.edu/d/agjv

[7] Lynx Software Technologies, "LynxOS Datasheet Final", Dec. 2, 2017, www.lynx.com/pdf/LynxOSDatasheetFinal.pdf

can be found in the most is that of aeronautical and aerospace application design. In avionics, the operating system must be able to meet the DO-178 standards which were set by the Federal Aviation Administration (FAA). The DO-178 standards is a straightforward way of being able to certify developed aviation software as being safe for avionic use. Any additional information regarding to the DO-178 standards can be found under Title 14: Aeronautics and Space of the Code of Federal Regulations (CFR), Part 21, Subpart O.[8] Since LynxOS has the DO-178 certification[9], as well as military certifications, several systems that have been developed in LynxOS include:[10]

- Airbus A380 Superjumbo Flight Test and Simulation
- Boeing 777 Cabin Services System
- Common ARTS Air Traffic Control Systems
- NASA's SLR2000 Satellite Ranging System
- Raytheon MK 57 Launching System DD(X)

As noted, most of these application systems were developed in the security of human life. As explained before, a Real-Time Operating System must be able to act as needed before a deadline. If there is a miscalculated step, or an improper performance spike, there will be a risk of human life casualties.

What is VxWorks?

Developed by Wind River Systems, an Intel Company, VxWorks is another Real-Time Operating System that is considered as a completely modular, secure, and scalable operating system which can be molded by the developer working on the application within the OS. Wind River's operating system states that it has been "building embedded devices and systems for more than 30 years."[11] Unlike LynxOS, VxWorks maintains to be fully customizable using a built on upgradable, and future proof OS architecture, where it can rapidly respond to any of the

[8] Federal Aviation Administration, "Regulations and Guidelines", Dec. 2, 2017, https://www.faa.gov/

[9] Lynx Software Technologies, "LynxOS-178 Certified RTOS", Dec. 2, 2017, http://www.lynx.com/products/real-time-operating-systems/lynxos-178-rtos-for-do-178b-software-certification/

[10] Lynx Software Technologies, "Lynx Software Technologies in Military Programs", Dec 2, 2017, http://www.lynx.com/industry-solutions/lynx-software-technologies-in-military-programs/.

[11] Wind River Systems, "VxWORKS Product", Dec. 2, 2017, https://www.windriver.com/products/vxworks/#VxWorks

changing industry requirements. If customer needs and technology advancements are occurring, VxWorks is a modifiable operating system where a developer may work without having to do a complete overhaul of application development. It is stated that the "VxWorks core kernel is separate from protocols, applications, and other packages, enabling upgrades and new feature additions to be accomplished faster and with minimal retesting of the entire system."[12]

Not only is VxWorks considered to be a reliable operating system when it comes to modulation, it sets itself apart from any other RTOS by stating that it has extensive support for both 32-bit and 64-bit central processing units (CPUs). This support ranges to also include any single-core and multi-core architectures that the application design might need to be supported on. Just like LynxOS, VxWorks is also completely deterministic and has the ability to read-in to a specific responsiveness that a computer might need. The one key feature that does make VxWorks shine out from LynxOS is that the operating system is able to function without the use of a memory management unit (MMU). This a key component for hardware support as it can access physical memory on the computer. It is important because it makes for a dependability for memory protection, and is mostly required for a good chunk of Real-Time Operating Systems. But because VxWorks doesn't require the developer to use an MMU, a developer has more flexibility for application design, but does not guarantee any memory protection when running.

Where is VxWorks Found?

Just like LynxOS, VxWorks has been found in many applications in the same industries as its competitor. VxWorks not only holds a DO-178C flight safety certification for avionics, but it also holds an IEC 61508 certification for operating system safety for the industrial trade, a EN 50126 certification for the transportation trade, and an ISO 26262 certification for automotive technologies.[13] Just like the mentioned DO-178 flight safety certification, information about the other certifications can be researched on the world-wide web. Not only does Wind River Systems hold the highest standards in safety, it also makes sure that the safety of the operating system is accounted for in the reliability and functionality of the system before it is sent out into

[12] Wind River Systems, "VxWorks Product Overview", Dec. 2, 2017, https://www.windriver.com/products/product-overviews/2691-VxWorks-Product-Overview.pdf.

[13] Wind River Systems, "Functional Safety in a Software-Defined World", Dec. 4, 2017, https://www.windriver.com/functionalsafety/intel/

the world. Most of the application designs were VxWorks is included are systems were once again human life is factored in. The multiple systems that have been created using VxWorks and are currently still in operation include systems for:

- NASA's Mars Rover Curiosity[14]
- Northrup Grumman Unmanned Combat Air System[15]
- Safety-Certified Boeing 787 Dreamliner[16]
- Olympus Surgical Generator[17]
- Varian Image-Guided Radiotherapy and Radiosurgery System[18]

Most of the listed systems above are currently in use to this day, especially the Boeing 787 Dreamliner. The 787 Dreamliner does have variants models of its specific class aircraft and can hold up to an approximate 335 passengers at a time. Now imagine, that the lives of those passengers are in the hands of not just the pilots who fly the airliner, but because of the integration of VxWorks into the entire the aircraft systems. VxWorks isn't just in the main computer of the airliner, but allows to "integrate multiple applications at different safety levels on the Common Core System."[19] It is the job of VxWorks to keep this airliner afloat, and because of its real-time strategies, it is one of the prime examples of how a Real-Time Operating System is so important.

[14] Wind River Systems, "NASA's Mars Rover Curiosity Powered by Wind River", Dec. 4, 2017, https://www.windriver.com/announces/curiosity/Wind-River_NASA_0812.pdf.

[15] Wind River Systems, "Wind River Customers: Northrop Grumman", Dec. 4, 2017, https://tinyurl.com/y95hrw85.

[16] Wind River Systems, "Wind River Customers: The Boeing Company", Dec. 4, 2017, https://tinyurl.com/y86q2rq8.

[17] Wind River Systems, "Wind River Customers: Olympus Surgical", Dec. 4, 2017, https://tinyurl.com/ydxypxgg

[18] Wind River Systems, "Wind River Customers: Varian Medical Systems", Dec. 4, 2017, https://tinyurl.com/ybdfwehy

[19] Wind River Systems, "Airbus A400M and Boing 787 Dreamliner First Flight", Published Dec. 15, 2009, http://blogs.windriver.com/wilson/2009/12/airbus-a400m-and-boeing-787-dreamliner-first-flight.html

LynxOS & VxWorks Architecture

LynxOS Architecture

Now throughout the case study, some similarities between the operating systems will arise. Since they are both Linux-kernel based Real-Time Operating Systems, both will have some hardware and software features that will act the same way. This can be like the architecture on the operating systems are based on, the way they function, or better yet who the company is trying to gain interest from. The differences will be that the structural design of the operating systems and how they manage memory, as well as schedulers is what defines them both. We already know that VxWorks is modular and does not require the use of the MMU, whereas LynxOS is not modular and requires the developer to use the MMU always.

First, LynxOS runs on a microkernel architecture, also known as its own "separation kernel" where it can combine secure and real-time components using varying partitions within the system design. A microkernel is a minimal OS kernel where it is less prone to errors, system services are easier to implement at user-level servers between the connections of multiple applications, and has a better protection between individual components in the system.[20]

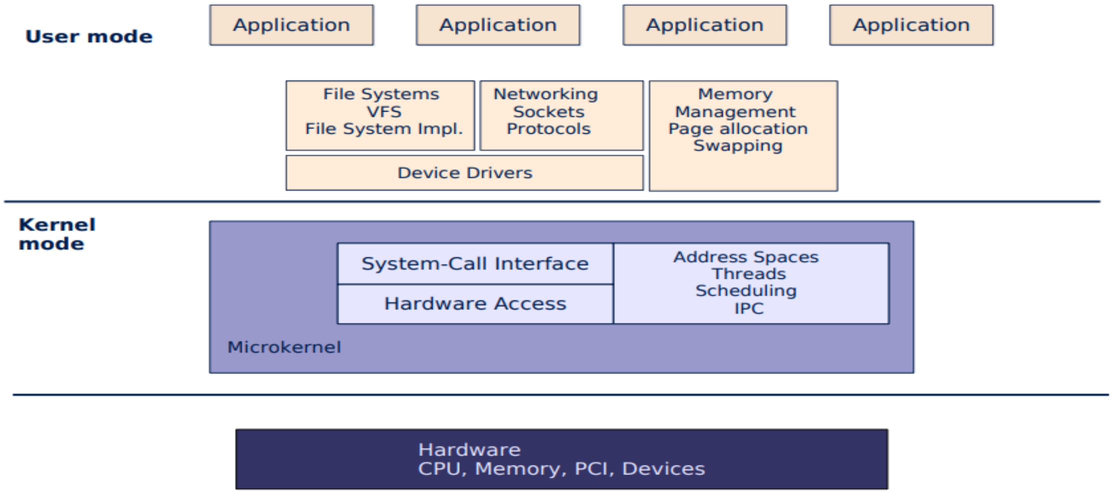


Figure 1: Basic Microkernel Model

[20] Technical University of Dresden, "Microkernel-based Operating Systems – Introduction", Published Oct. 14, 2008, http://os.inf.tu-dresden.de/Studium/KMB/WS2008/01-Introduction.pdf

A microkernel based operating system also has the safe possibility that if one component decides to crash, or terminate on its own because of a failure, it doesn't necessarily crash the entire system. The component that is running on a separate partition of the system, will only crash within that partition without affecting the other partitions controlling the other components or applications. As noted, applications and system services exist within partitions, where the hardware components exist in their own level. [21]

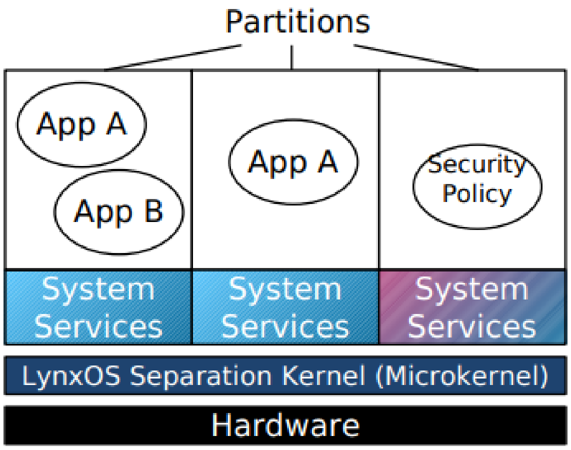


Figure 2: LynxOS Partition System

Component Structure of LynxOS

One of the operating systems that Lynx Software Technologies prides on of course its LynxOS-178 Certified RTOS which is the operating system of choice for some developers when designing applications and systems for aerospace and aeronautical use. Remember that, because the operating system is DO-178 certified, it means that the operating system passed all tests regarding that the OS was fit for use and was able to pass all safety measurements that the FAA takes care of. Within the LynxOS package, there exist components that serve to protect the controller and the system that is running on LynxOS. Most of the important safety features of the operating system exist in the ARINC 653 Services package. This a multitude of many services that make sure to check on services such as partition management, process management, time management, and of course interpartition communication which is responsible for any communication to exist between the services that exist in the multiple partitions of the system.

[21] Technical University of Dresden, http://os.inf.tu-dresden.de/Studium/KMB/WS2008/01-Introduction.pdf

Since LynxOS is a microkernel based operating system, and as noted before, this is known as the separation kernel that exists on the bounds of software and hardware, a key component that really makes up for this structure is the ARINC 653 Health Monitoring component. This component is important because it is a component that lets the OS know that an error has occurred and is predicting a fault within the system itself. The component is invoked by either the service application, the OS, or even the hardware that is becoming faulty as the alert is being sent out. Since LynxOS wants to achieve complete system security, it uses "Virtual Machine (VM) brick-wall partitions of time, memory and resources."[22] Each of these partitions in a way acts like a stand-alone version of the operating system on its own. Any system events that occur in the OS, in either one of the partitions cannot share any resources or interfere with other system events in any of the other partitions. The only partition that can be interfered with is Virtual Machine Zero (VM0), which handles system administrative services within the root of the POSIX system services unit.[23]

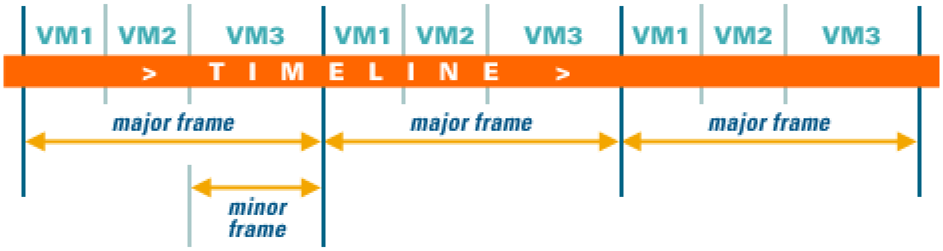


Figure 3: ARINC 653 Health Monitoring Component

Another special component of LynxOS is the CPU Support Package (CSP) which contains all the processor routines, which includes the MMU, the floating point, and the processor exception handlers. These routines are all linked to the LynxOS-178 microkernel. This package contains multilevel system setup routines that apply to hardware, system software, and the application software within the system that is being designed. For example, the above mentioned ARINC 653 Health Monitor exists in the application software level, but stays within the first partition. This monitor is then linked to the system software's partitioning kernel that makes sure that all connections are secured and functioning. From here, the connections are then

[22] Lynx Software Technologies, "LynxOS-178 Certified RTOS", Dec. 4, 2017, http://www.lynx.com/products/real-time-operating-systems/lynxos-178-rtos-for-do-178b-software-certification/

[23] Lynx Software Technologies, http://www.lynx.com/products/real-time-operating-systems/lynxos-178-rtos-for-do-178b-software-certification/

linked to the CSP, board support package, or to any of the other middleman packages that exist between the system software and hardware levels. After passing through these packages, the link is made once more to either the microprocessor, any hardware components, the PCI controller, or any optional hardware that isn't connected to one of the main pieces of equipment. This structure maintains that all partitions are separated from one another, so that LynxOS can run on them separately, but still makes sure that all connections are made within the computing system.[24]

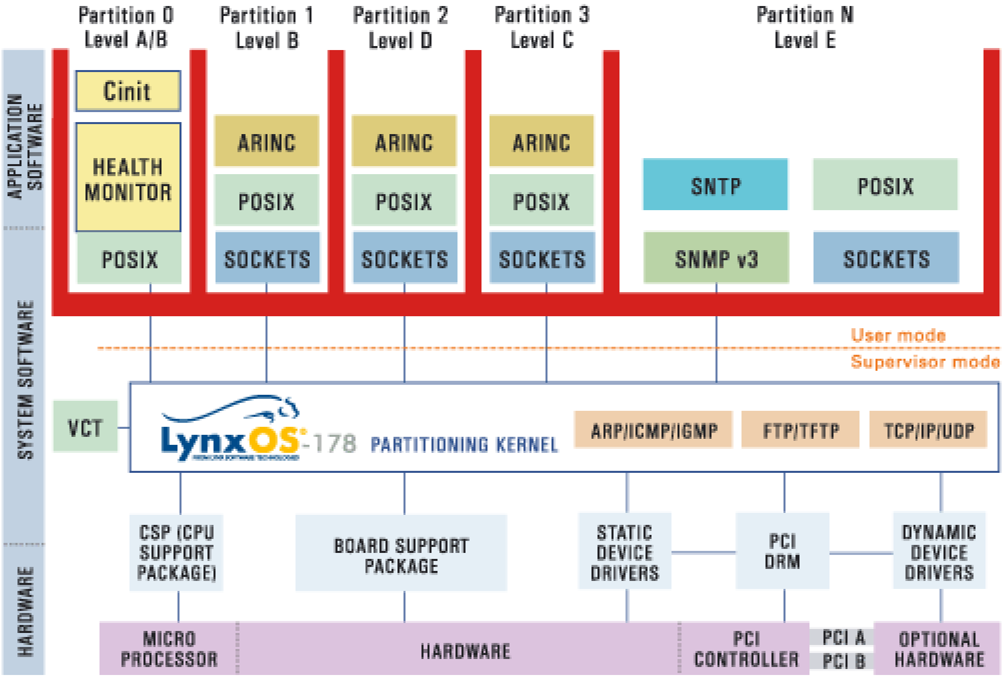


Figure 4: LynxOS Structure Model

VxWorks Architecture

Just like LynxOS, the VxWorks RTOS is a Linux-kernel based operating system, that also runs as a microkernel system. The operating system looks familiar to that of LynxOS, but instead of having separations of software on a user level, there are modules of middleware within the first block. The first block is considered the user mode section of the architecture of VxWorks. Instead of having three multiple levels of structure design like that of LynxOS, VxWorks has two, a user level side where development of application occurs, and a kernel mode side which ties the microkernel, a board support package, and a hardware board with a secure

[24] Lynx Software Technologies, http://www.lynx.com/products/real-time-operating-systems/lynxos-178-rtos-for-do-178b-software-certification/

boot helper all together. This is considered a multitasking kernel which can guarantee fast interrupt handling because of its preemptive scheduling. The reason the structure of the operating system is designed like this is so that it can maintain consistent, deterministic system performance. This performance is there in case there is a need for a few number of partitions or even a large amount of partitions to design and implement into a computer system. The separation kernel that exists between the software and hardware bounds also exist, but because of the introduction to middleware design within the system software build, is what helps to make sure the operating system is ready for a working environment. The VxWorks architecture model is shown below.[25]

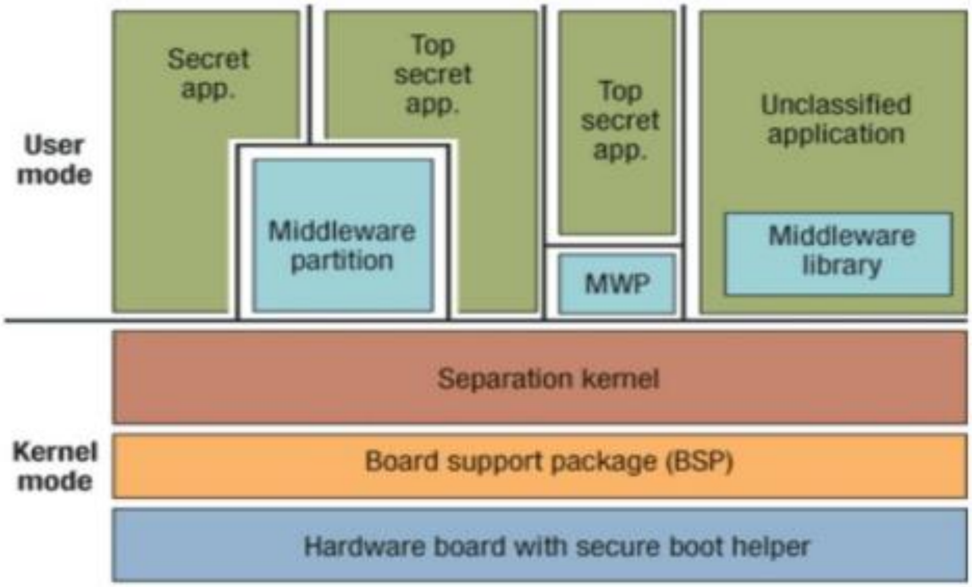


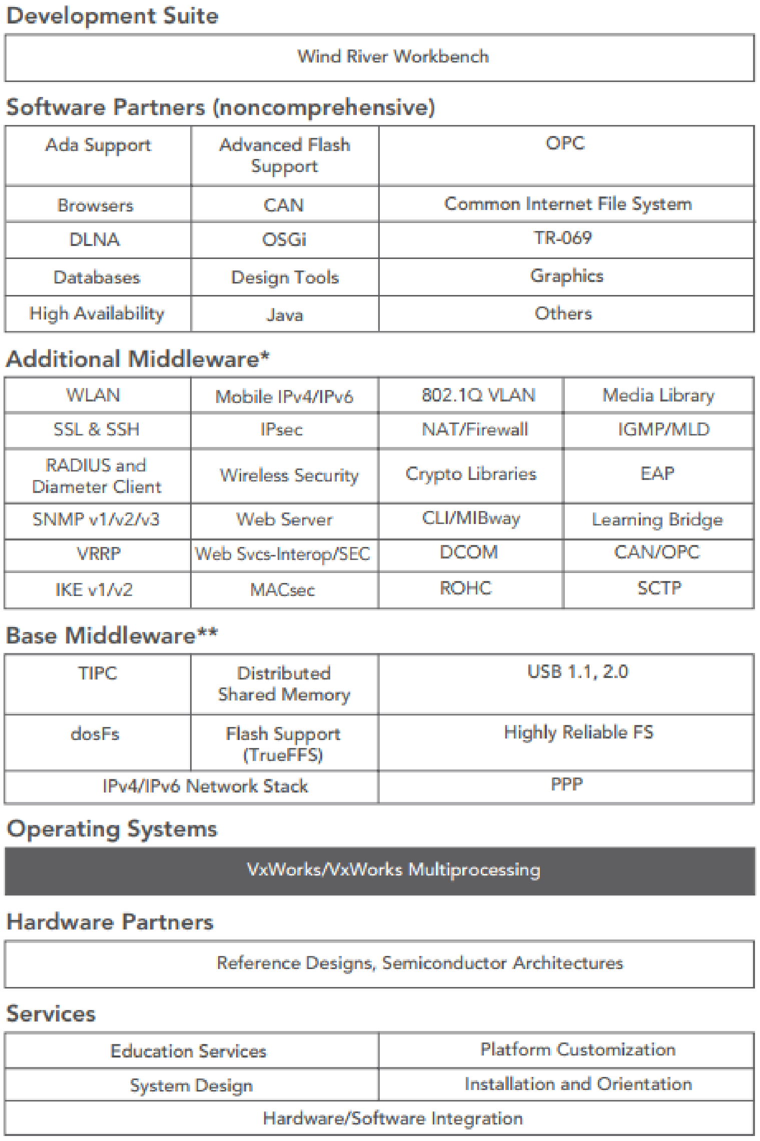
Figure 5: VxWorks Architecture Model

Component Structure of VxWorks

Unlike the structural system of LynxOS where the key elements of the operating system structure are linked to one another, VxWorks has the uncanny ability to be modular. So, this means that a developer can decide what kind of components he/she can add into the operating system while they are developing the application. From the very top of the application structure, there is the Wind River Workbench which comes bundled with tools and debuggers that

[25] Kensing, Hampus. “A comparison between VxWorks and LynxOS regarding memory management second edition.” (2006).

application developers can use to test their systems with the operating system. The Workbench is an Eclipse-based (Java IDE) development suite for developing projects, as well as being able to manage host-to-target communications between embedded hardware and other applications. The Workbench comes primed with configuring, debugging, and monitoring tools that can watch VxWorks as well as the VxWorks applications that are running on either a simulated version of the system, or on real hardware. A simulated version of the OS is included in the Workbench.



Development Suite

Wind River Workbench

Software Partners (noncomprehensive)

Ada Support	Advanced Flash Support	OPC
Browsers	CAN	Common Internet File System
DLNA	OSGi	TR-069
Databases	Design Tools	Graphics
High Availability	Java	Others

Additional Middleware*

WLAN	Mobile IPv4/IPv6	802.1Q VLAN	Media Library
SSL & SSH	IPsec	NAT/Firewall	IGMP/MLD
RADIUS and Diameter Client	Wireless Security	Crypto Libraries	EAP
SNMP v1/v2/v3	Web Server	CLI/MIBway	Learning Bridge
VRRP	Web Svcs-Interop/SEC	DCOM	CAN/OPC
IKE v1/v2	MACsec	ROHC	SCTP

Base Middleware**

TIPC	Distributed Shared Memory	USB 1.1, 2.0
dosFs	Flash Support (TrueFFS)	Highly Reliable FS
IPv4/IPv6 Network Stack		PPP

Operating Systems

VxWorks/VxWorks Multiprocessing

Hardware Partners

Reference Designs, Semiconductor Architectures

Services

Education Services	Platform Customization
System Design	Installation and Orientation
Hardware/Software Integration	

VxWorks platform components

Additional components[26] can be added into the customization of VxWorks. Reminder that because of its modular state, any additional middleware can be included and anything that isn't needed can be removed. This is a powerful addition for the design of any memory-constrained devices that can only have a small-footprint of memory in the system. The smaller the memory footprint, the faster the operating system can boot itself up and allow for applications to start working. The Base Middleware is the only middleware that stays within the operating systems main feature log. All other components like Hardware Partners and Services are accompanied with the OS so that proper application design can be integrated.

[26] Wind River Systems, "Wind River VxWorks Platforms 6.9", Dec. 4, 2017, https://www.windriver.com/products/product-notes/PN_VE_6_9_Platform_0311.pdf

Operating System Functions

LynxOS Preemption Delays and Scheduling

In a Real-Time Operating System, the model for most of any of the real-time applications that are designed to make sure that multiple tasks, have their own response times and needs. In LynxOS, this is supported by providing a priority preemptive scheduling process. In preemptive scheduling, a process that is currently running can be interrupted if a higher priority process comes into the schedulers view. The current running process will be paused and will allowed to be finished until the new process has finished. Adding priority to this means that if a process has a higher priority than a lower priority item that is currently running, the lower priority item will be paused and allows the higher priority item to cut in front of the line. LynxOS' scheduling was designed to be preemptive and reentrant, as well as being based on scheduling algorithms such as First-In, First-Out and Round Robin. This allows the operating system to set true task priorities and task preemption into the kernel. The operating system even goes beyond this by making sure to execute any extended and asynchronous interrupts that are being processed at any task priority levels. Even if there are any preemption delays or blocking times that can be caused by a fault in the kernel, these can be used in conjunction with other task execution times so that single tasks can reach their deadlines on time and without process faults.

The microkernel itself was designed in a specific way so that it can be fully preemptive, without having the hassle of adding long blocking regions that can cut out other process in the running scheduler. Data structures within the kernel, which are shared without the application developer being aware, are protected by being temporarily disabled for a brief time. This in effect creates a priority ceiling protocol that protects the data structures using semaphores. As a double check, to check that preemption is disabled for that brief period, the data structures were built for an increased performance in deterministic access. The data structures that are being shared, such like that as a data file or an I/O channel, have fast, but long access times that can commit to the running scheduler. The known creation of blocking and of the preemption delays within the LynxOS kernel, even where there is a presence of interruption, makes it possible to be able to use “analytical methods to ensure a set of real-time tasks that will always meet their deadlines.”[27]

LynxOS Interrupt Handling System

The LynxOS has an intriguing interrupt system that handles interruption unlike any other RTOS. During the date of the published work for where this is being researched from, the AAUGN states that “most operating systems simply execute interrupt processing to completion, allowing it to be preempted only by higher priority interrupts.”[28] Since it is possible for a computer to be connected to a network, have access to a mass storage device, or having to handle a user interface, the computer can receive multiple hardware interruptions from various sources at the same time. As the scheduler is running, this would put strain on the CPU, as it would steal time from the tasks that would really need the CPU. But LynxOS does come to the rescue as it solves this problem by executing a bulk of the interrupting services at the task priority level by using dedicated kernel threads for these processes. The interrupt thread priority is based on the highest priority task that has access to the device that is generating the interruption. Once the task is located, any interruptions are re-engaged by the kernel thread. This puts a deadline bound on the amount of time a high priority task can be delayed because of the interruptions. With this, a system can be created to be predictable even when in the face of unpredictable interruptions that can be caused in a very high, and stressful environment.

LynxOS Memory Management

Unlike VxWorks unique state of modular, LynxOS does require that a developer uses the reliability of the Memory Management Unit (MMU) within the computer/system. This unit can be found in the lowest section of the operating system’s kernel. This memory manager provides the “…advantages of protected memory and performance advantages of virtual addresses.”[29] Unlike other real-time operating systems where they only rely on unprotected processes running in a single virtual address space, LynxOS makes sure to enable each task to run in its own virtual

[27] Australia Post; AUUGN, “Australian UNIX systems User Group Newsletter AUUGN”, Vol. 13(No.6), Published December 1992, ISBN: 1035-7521, pages 68 – 70, https://tinyurl.com/yb968slk

[28] Australia Post, AUUGN, page 69

[29] Lynx Software Technologies, “LynxOS RTOS”, Dec. 4, 2017, http://www.lynx.com/products/real-time-operating-systems/lynxos-rtos/

address space protected, from other processes that would be currently running. Memory is also managed by being partitioned by dividing the random-access memory (RAM) into distinct blocks of non-overlapping physical address spaces. Each one of these partitions is given only one block of memory to run on. From inside the partition, the virtual address spaces of other several processes and tasks are mapped to the memory from the assigned memory block that it was given.[30]

Priority Scheduling in VxWorks

In VxWorks, the Wind microkernel that runs the scheduler within the operating system is noted to have a small memory footprint. This microkernel, which is small, runs on priority based scheduling just like LynxOS, except it has 256 priority levels by using both preemptive and non-preemptive round-robin (RR) scheduling algorithms. In this algorithm, priority 0 has the highest priority, while priority 255 is the lowest of the count. If a task with a high priority is set to be ready to run, then the current task that is running on the scheduler is paused and is preempted. As it pauses, any task data is saved and the next process that is ready to run has the context of its data placed into the set of the new task.

Any transfer of data that might be needed is passed along from one task to another. But, VxWorks introduces round-robin especially when multiple tasks of the same priority want to use the CPU. Round-robin uses time slicing so that tasks can equally share the CPU among any other tasks of that same level of priority. Once a task uses up the entirety of their time slice, it is moved to the back of the queue and then another task is pushed up into the ready queue where the old task was it. This ensures that all tasks have an equal share of the CPU, and that data can be attained without the extent of a failed task item. This scheduler, unlike the one used by LynxOS, can be disabled and enabled by the developer if needed. In case of synchronization issues, semaphores with priority inheritance is used. Not only does VxWorks use one set of Wind Semaphores that are created by Wind River Systems, but it also uses POSIX Semaphores for a better ease of portability amongst other systems.[31]

[30] Lynx Software Technologies, “LynxOS-178 Certified RTOS”, Dec. 4, 2017, http://www.lynx.com/products/real-time-operating-systems/lynxos-178-rtos-for-do-178b-software-certification/

[31] Carlgren, Henrik; Ferej, Ranjdar, “Comparison of CPU Scheduling in VxWorks and LynxOS”, Dec. 4, 2017, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.111.82&rep=rep1&type=pdf

VxWorks Memory Management

Since VxWorks is once again noted to be modular, the operating system is distinctly different than its counter-part LynxOS, because it allows a developer to turn off the Memory Management Unity (MMU). This unit can be disabled to allow for better flexibility for a developer that might be programming an application that might be running on an older version of VxWorks where the MMU wasn't needed. As the operating system is scalable, being able to turn off the MMU allows for a developer to upgrade the current version of their operating system to something newer without it affecting the application design. But, even though this might be an effective use of flexibility, turning off the MMU also brings trouble as memory reliability is also nonexistent. A device manufacture can use a wider array of processors, being able to fit the needs of the system, but now the programmer must make sure that all code is effective. Since the memory-model is also non-overlapping, this allows for memory translation tables to not be required. This slightly improves the access performance of the memory, as well as being able to save memory space.

Even though the access performance of the memory is improved, as noted by Kensing and Vralstad, "It seems VxWorks gets less stable without MMU memory protection. Several developers experience that the OS crashes if only one single task crashes."[32] Disabling the memory management unit does keep costs low for developers, but for those who do choose to enable it, the operating system does support memory partitioning and heap allocation services. In earlier versions of VxWorks, memory allocation was based on a first-fit policy. This policy meant that had a small overhead of memory access and the search algorithm that was being used was fairly simple. This method could be designed so that within the memory, if a data structure of a linear linked list existed, the lists could link together any other blocks of lists that were free roaming. This method gives fast memory allocation speeds at the very start of the bootup process of the system, but if there was a dynamic allocation that needed to be performed, the fragmentation of the data structures would occur. This caused reduced memory allocation performance right when the system started to initialize memory from one sector to another. After

[32] Kensing, Hampus; Vralstad, Andreas, "A comparison between VxWorks and LynxOS regarding memory management", Published Autumn 2006, page 9, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.116.7916

Wind River Systems realized that application complexity, as well as the allocation of dynamic memory was in hot demand, the company decided on using a fit policy that resulted in less memory allocation, but an increase in execution time.

VxWorks Memory Protection

Even though the MMU can be disabled, Wind River Systems pushes for the use of MMU so that they can have free reign of the memory protection units that VxWorks uses. VxWorks uses a traditional kernel-mode execution system, but also provides an additional user-mode application execution that protects the kernel's execution from any user-mode programs that might be running already. These already running processes are called real-time processes (RTPs) and VxWorks keeps maintaining to keep these processes secured even while other intervening process would like to hitch along for the ride. The memory protection that is found in the MMU provides an isolation cell from the kernel, allowing for an increase in device reliability. Not only does this secure a process from an RTP, but it also helps to simplify the model of application development. Not only is memory protected, but VxWorks also has the "…ability to create private or public objects in the kernel and in RTPs, offers flexibility to use objects that are protected from manipulation or that can be easily shared among kernel and process tasks."[33]

[33] Wind River Systems, "Wind River VxWorks Platforms 6.9", page 4, Dec. 5, 2017, https://www.windriver.com/products/product-notes/PN_VE_6_9_Platform_0311.pdf

Memory Security - Design Evaluation

Evaluation of LynxOS

Out of the two Real-Time Operating Systems, I would have to say that LynxOS is a more suited and reliable operating system compared to VxWorks. Lynx Software Technologies makes sure to provide the developer with the best set of resources and guides so that one may be able to develop applications using their software. It might not be as flexible as VxWorks regarding modularity and scalability, and that might set LynxOS back a couple of steps. But because the company makes sure that the operating system only runs when the memory management unit is enabled, makes for constant safe applications in system to be created. In my eye, if a system is modular, then this mean that there might be insecurities that could cause issues for outside threats. After doing a majority of research to see if there were any inconsistencies could have reported, none were found. In my mind, LynxOS will stay as the reliable and trusted Real-Time Operating System that many developers choose to go with.

Evaluation of VxWorks

On the other hand, VxWorks is a modular operating system that gives reliable flexibility for developer when first starting out in the technological world. This means, that a company could have wants for their application, and being able to remove libraries and manage memory by removing content that is not needed is a big plus. But, being modular also so means that there are risks that come with it. After doing research, I came across an article about ethical hacking, which is being able to break into a system, find its weaknesses, and tell the company of how they exist. The company can then fix the issue and fix upcoming issues that might grow from the extent of the damage. The main reason that this hack was able to break and bring the operating system down was because it was a modular system. The main issue that was found was within the system-level debugger the for the operating system that runs on a User Datagram Protocol (UDP) Port #17185. This port, which stays open for developer use, allows anyone access to the port to be able to "read memory, write memory, call functions, and manage tasks"[34] within the

[34] Live Hacking: Ethical Hacking | Penetration Testing, "VxWorks Vulnerabilities", Published Aug. 2, 2010, http://www.livehacking.com/2010/08/02/vxworks-vulnerabilities/

operating system. The writer also notes that since the port is a UDP, and there is no authentication needed, a handshake between applications, nor a session ID of who is accessing this port, requests made to the debugger can be spoofed by an attacker. Spoofing, means that an attacker can gain access to a non-locked feature, fool it into thinking that the developer is actually accessing this port, and can than modify anything within the system without any issues.

Luckily, this issue has already been resolved, but it also questions whether or not the operating system is reliable. LynxOS isn't modular, it doesn't allow for customization options because it acts in a way like a lock where the lock does need to be changed frequently. Whereas in VxWorks, if options need to be modified, removed and reapplied, it's like keeping the door open regarding memory security. LynxOS keeps the door shut, hammered in with nails, and glues the outside edges so that nothing can be changed within the operating system. It is possible that LynxOS could be hacked, or could be spoofed, but because they make sure that the system is not modifiable means that chances for this might be slime.

The idea of modular operating systems isn't bad, it just depends whether a grad student is using the operating system for a project, or if a multi-million-dollar corporation is using for their systems. As mentioned before, VxWorks is installed onto the onboard systems of a Boing Dreamliner 787. In 2008, there was an article by Kim Zetter for Wired Magazine[35] that stated that it is possible for a hacker to actually take over the network on the airplane and have access to it. The article has no mention of VxWorks, but it is possible that VxWorks could be for blame. Even seven years after the first report had come out, it was still noted in another article[36] that the same class of airliners could still be hacked. Seven years, and yet the problem still exists. If it could be possible, systems like these should have more restricted access to developer side modules. Certifications need to be improved, and a system like VxWorks should a complete overhaul and see where the leak is at. Why is it that LynxOS hasn't had an issue like this so far? If anything, Wind River Systems, should learn a thing or two about the security that are going to be needed now, and in the greater future of technological advancements.

[35] Kim Zetter, "FAA: Boeing's New 787 May Be Vulnerable to Hacker Attack", Published Jan. 4, 2008, https://www.wired.com/2008/01/dreamliner-security/

[36] Kim Zetter, "Hackers Could Commandeer New Planes Through Passenger Wi-Fi", Published April 15, 2015, https://www.wired.com/2015/04/hackers-commandeer-new-planes-passenger-wi-fi/

References

[1] Richard Barry, “What is an RTOS?”, freeRTOS, Dec. 2, 2017, https://www.freertos.org/about-RTOS.html

[2] Margaret Rouse, “Real-Time Operating System (RTOS)”, SearchDataCenter, Dec. 2, 2017, http://searchdatacenter.techtarget.com/definition/real-time-operating-system.

[3] Indiana University, “What is POSIX?”, Dec. 2, 2017, https://kb.iu.edu/d/agjv

[4] Lynx Software Technologies, “LynxOS Datasheet Final”, Dec. 2, 2017, www.lynx.com/pdf/LynxOSDatasheetFinal.pdf

[5] Federal Aviation Administration, “Regulations and Guidelines”, Dec. 2, 2017, https://www.faa.gov/

[6] Lynx Software Technologies, “LynxOS-178 Certified RTOS”, Dec. 2, 2017, http://www.lynx.com/products/real-time-operating-systems/lynxos-178-rtos-for-do-178b-software-certification/

[7] Lynx Software Technologies, “Lynx Software Technologies in Military Programs”, Dec, 2, 2017, http://www.lynx.com/industry-solutions/lynx-software-technologies-in-military-programs/.

[8] Wind River Systems, “VxWORKS Product”, Dec. 2, 2017, https://www.windriver.com/products/vxworks/#VxWorks

[9] Wind River Systems, “VxWorks Product Overview”, Dec. 2, 2017, https://www.windriver.com/products/product-overviews/2691-VxWorks-Product-Overview.pdf.

[10] Wind River Systems, “Functional Safety in a Software-Defined World”, Dec. 4, 2017, https://www.windriver.com/functionalsafety/intel/

[11] Wind River Systems, “NASA’s Mars Rover Curiosity Powered by Wind River”, Dec. 4, 2017, https://www.windriver.com/announces/curiosity/Wind-River_NASA_0812.pdf.

[12] Wind River Systems, “Wind River Customers: Northrop Grumman”, Dec. 4, 2017, https://tinyurl.com/y95hrw85.

[13] Wind River Systems, “Wind River Customers: The Boeing Company”, Dec. 4, 2017, https://tinyurl.com/y86q2rq8.

[14] Wind River Systems, “Wind River Customers: Olympus Surgical”, Dec. 4, 2017, https://tinyurl.com/ydxypxgg

[15] Wind River Systems, “Wind River Customers: Varian Medical Systems”, Dec. 4, 2017, https://tinyurl.com/ybdfwehy

[16] Wind River Systems, “Airbus A400M and Boing 787 Dreamliner First Flight”, Published Dec. 15, 2009, http://blogs.windriver.com/wilson/2009/12/airbus-a400m-and-boeing-787-dreamliner-first-flight.html

[17] Technical University of Dresden, “Microkernel-based Operating Systems – Introduction”, Published Oct. 14, 2008, http://os.inf.tu-dresden.de/Studium/KMB/WS2008/01-Introduction.pdf

[18] Lynx Software Technologies, “LynxOS-178 Certified RTOS”, Dec. 4, 2017, http://www.lynx.com/products/real-time-operating-systems/lynxos-178-rtos-for-do-178b-software-certification/

[19] Kensing, Hampus; Vralstad, Andreas, "A comparison between VxWorks and LynxOS regarding memory management", Published Autumn 2006, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.116.7916

[20] Wind River Systems, "Wind River VxWorks Platforms 6.9", Dec. 4, 2017, https://www.windriver.com/products/product-notes/PN_VE_6_9_Platform_0311.pdf

[21] Australia Post; AUUGN, "Australian UNIX systems User Group Newsletter AUUGN", Vol. 13(No.6), Published December 1992, ISBN: 1035-7521, pages 68 – 70, https://tinyurl.com/yb968slk

[22] Lynx Software Technologies, "LynxOS RTOS", Dec. 4, 2017, http://www.lynx.com/products/real-time-operating-systems/lynxos-rtos/

[23] Lynx Software Technologies, "LynxOS-178 Certified RTOS", Dec. 4, 2017, http://www.lynx.com/products/real-time-operating-systems/lynxos-178-rtos-for-do-178b-software-certification/

[24] Carlgren, Henrik; Ferej, Ranjdar, "Comparison of CPU Scheduling in VxWorks and LynxOS", Dec. 4, 2017, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.111.82&rep=rep1&type=pdf

[25] Wind River Systems, "Wind River VxWorks Platforms 6.9", page 4, Dec. 5, 2017, https://www.windriver.com/products/product-notes/PN_VE_6_9_Platform_0311.pdf

[26] Live Hacking: Ethical Hacking | Penetration Testing, "VxWorks Vulnerabilities", Published Aug. 2, 2010, http://www.livehacking.com/2010/08/02/vxworks-vulnerabilities/

[27] Kim Zetter, "FAA: Boeing's New 787 May Be Vulnerable to Hacker Attack", Published Jan. 4, 2008, https://www.wired.com/2008/01/dreamliner-security/

[28] Kim Zetter, “Hackers Could Commandeer New Planes Through Passenger Wi-Fi”, Published April 15, 2015, https://www.wired.com/2015/04/hackers-commandeer-new-planes-passenger-wi-fi/