LYNUXWORKS™

ARINC and POSIX

FOR SAFETY-CRITICAL APPLICATIONS

POSIX and ARINC for Safety-Critical Applications

DOC-0620-00

Product names mentioned in *POSIX and ARINC for Safety-Critical Applications* are trademarks of their respective manufacturers and are used here for identification purposes only.

Printed in the United States of America.

Contents

1 POSIX and ARINC

Why Choose LynxOS-178 2.0? Because it combines the efficiency, versatility, and strength of POSIX with the safety, control, and reliability of ARINC.

With full POSIX compliance and support for ARINC 653-1, LynxOS-178 2.0 enables developers to take advantage of the time-to-market and investment-protection benefits of open standards-based development.

For developers of safety-critical systems, the LynxOS-178 offers a commercial off-the-shelf (COTS) system with the following key benefits:

- Low risk—Standards-based DO-178B level A certifiable, real-time operating system package at a known cost.
- Reduced costs—Eliminates man-years of effort and significantly lowers overall cost of certification.
- High value—Most robust feature set of any DO-178B certifiable operating system enables rapid time-to-market for safety- and business-critical applications.
- POSIX compliance—The only real-time, POSIX-compliant RTOS available today.
- Support for ARINC 653-1—Ensures application portability, software reuse and interoperability between embedded systems.

LynxOS-178 2.0 delivers the security and real-time responsiveness needed for safety-critical systems, with a level of POSIX and ARINC 653-1 compliance previously unavailable in the avionics industry.

With the ability to efficiently design products by leveraging readily available POSIX- and ARINC 653-1-compatible software, developers can enter the market more quickly and economically than ever before.

2 LynxOS-178 Overview

LynxOS-178 is the first DO-178B- and EUROCAE/ED-12B-certifiable, POSIX-compliant Real Time Operating System (RTOS) solution. The LynxOS-178 product family includes:

- A suite of standards-based development tools
- Full customer support, including DO-178B-capable consulting services
- An artifacts package for the OS with DO-178B-required documentation

LynxOS-178 Key Features

Hard Partitioning of Resources in Time and Space

LynxOS-178 implements a fixed-cyclic scheduling algorithm that gives each partition fixed execution time so that the system can be deterministically safe. Additionally, the system allows multiple applications of differing criticality levels within partitions to execute, completely isolated, on the same hardware resource. With LynxOS-178, each application runs protected in its own space for uncompromising reliability within a hard-partitioned virtual machine, enabling easier application certification.

Multiprocess and Multithreaded Environment

LynxOS-178 supports a multiprocess, multithreaded environment within a partition that allows applications (POSIX and ARINC) to run seamlessly within a single partition. Applications use processes and threads, make system calls, and use device drivers. The product can run a shell on a serial port for a developer to interact directly with the target machine. It also has device drivers to permit mounting an external disk drive to facilitate testing and data capture.

Previously Approved DO-178B Level A Certified Artifacts

LynxOS-178 provides previously certified software and artifacts that allow developers to speed safety-critical systems to market. LynxOS-178 certified software provides full DO-178B traceability through requirements, design, code, test, and test results.

Full Requirements-Based Testing on 100 Percent of Code

One of the most costly efforts of DO-178B level A certification is the requirements-based testing, also known as the Structural Coverage requirement. For DO-178B level A, the code is required to be verified with Modified Condition/Decision Coverage (MCDC), which means that every point of entry and exit in a program must have been invoked at least once in testing, every decision in the program must have taken all possible outcomes at least once, and each condition in a decision must have been shown to independently affect that decision's outcome. LynxOS-178 satisfies the 100 percent MCDC structural coverage requirement out-of-the-box, allowing systems developers to concentrate on their applications rather than trying to get those last lines of system code exercised for system certification.

Mature, Stable, and Fully Certifiable Kernel

LynxOS-178, based on the LynxOS RTOS, is an embedded RTOS that has been rigorously exercised through millions of deployments and is the foundation of multiple safety-critical systems that have been certified to DO-178B, including the KC-135 and the Bombardier Challenger 300.

Mountable File System Support

LynxOS-178 implements a POSIX-compliant file system interface that supports the creation of fully functional file systems in DRAM, Flash, and so on. These file systems can be mounted read-write or read-only for additional flexibility in safety-critical environments.

Dynamic Device Driver

Applications and drivers are not required to be linked to the operating system and can, therefore, be isolated, limiting recertification efforts for the full operating system when only an application or driver needs modification.

LynxOS-178 Configurations

LynxOS-178 has two configurations: a certifiable production configuration for the delivery of safety-critical software and a powerful development configuration enhanced with development tools to accelerate product time-to-market.

LynxOS-178 Production Configuration

The production configuration of the LynxOS-178 has a feature set that has complete DO-178B artifacts and traceability for level A certification of the LynxOS-178 kernel and libraries.

LynxOS-178 Development Configuration

The development configuration (a superset of the production configuration) has additional features that assist in application development and debugging on LynxOS-178:

- More kernel features—`TTY`, `ptrace`, `skdb`, and CodeTEST hooks, for example.
- Shells and Utilities—`Bash`, `TFTP` application, `zmodem`, `ps`, `gzip`, high water mark utilities, and additional file system utilities such as `ls`, `cat`, `mkdir`, `copy`, and `rm`.
- Debuggers—Standard `gdbserver`
- Additional device drivers, such as SCSI drivers

Note that these additional features do not have supporting DO-178B artifacts and are intended for use during development only.

Partitioning

LynxOS-178 is a POSIX-compliant operating system designed to allow multiple real-time applications of different criticality levels to execute concurrently on the same processor. Adhering to ARINC 653-1, LynxOS-178 provides time, memory, and resource partitioning to applications, which prevents any application from affecting, or being affected by, any other application.

Virtual Machine

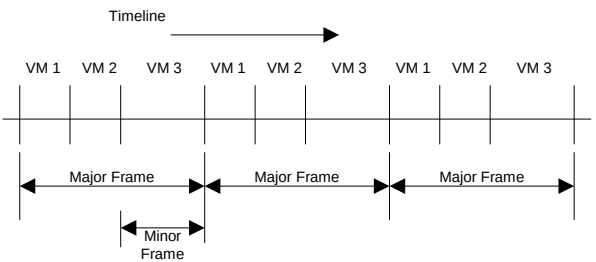
In LynxOS-178, each partition is called a virtual machine (VM). The name implies that processes running in one partition are aware of only events in their own partition, just as a process running on an ordinary operating system in one computer is aware of only events in that computer. The exception is VM0, which

has root privileges compared to the rest of VMs. These privileges are like the root privileges in UNIX systems. In addition, VM0 can “see” all the processes and threads of the other nonroot VMs. Within a nonroot VM (for example, VM1, VM2…), the processes cannot affect, or be affected by, processes in other nonroot VMs. In fact, processes in a nonroot VM have no way of knowing whether other VMs exist. The partitioning involves exclusive access of three kinds:

- Time
- Memory
- Resources

Time Partitioning

Time partitioning is done through a fixed-cyclic time-slice scheduler, which allocates periods of time to each partition as illustrated in the figure below.



During each time slice, only processes in the assigned partition are permitted to execute. LynxOS-178 implements an ARINC 653-1-based time partition scheduling algorithm that gives each partition fixed execution time so that the system can be deterministically safe.

Memory Partitioning

Memory partitioning is achieved by dividing RAM into discrete blocks of nonoverlapping physical address space. Each partition is assigned one and only one block of memory. Within the partition, the virtual address spaces of various processes are mapped to memory from the assigned memory block.

Resource Partitioning

Resource partitioning means that each device can be assigned to only one partition. This means that a fault in a device or its driver will be contained within a single partition. Each partition mounts a RAM-based file system for data storage. The file systems are private to the partitions and are never shared with other partitions.

LynxOS-178 and ARINC

ARINC is a leader in aviation communications and is recognized as the leading provider of transportation communications and systems engineering solutions for five major industries: aviation, airports, defense, government, and transportation. ARINC standards specify the air transport avionics equipment and systems used by more than 10,000 commercial aircraft worldwide. By adhering to the ARINC 653-1 standard, LynxOS-178 provides a solution that offers the source code portability of POSIX and meets the most stringent safety and reliability standards for military and commercial avionics systems. For more information about ARINC 653-1, see “Industry Standards.”

3 Industry Standards

DO-178B

The military and aerospace industry mandates rigorous technical and process requirements for safety-critical computing. In the U.S., these are expressed in the Radio Technical Commission for Aeronautics (RTCA) DO-178B standard for the production of software for airborne systems. The quantity of items needed for DO-178B certification, and the amount of information that they must contain, is determined by the level of certification being supported. The targeted DO-178B certification levels are as follows:

- A—catastrophic
- B—hazardous-severe
- C—major
- D—minor
- E—no effect

Running Different Levels of Certification within Partitions

Along with the operating system and the development tools, LynuxWorks provides the necessary artifacts to permit LynxOS-178 to be used in systems that are certifiable up to level A of the RTCA DO-178B standard. The artifacts for the LynxOS-178 product include requirements documents, design descriptions, test procedures, and test results for the components that make up LynxOS-178 reusable components.

LynxOS-178 supports RTCA DO-255-compliant system partitioning, which allows the system to run software of different DO-178B criticality levels on the OS in different partitions/VMs. This means the OS can run a DO-178B level A application in one VM and a level C application in another, both on the same processor in the same system.

ARINC 653-1

The term ARINC 653-1 is an abbreviation for the *Draft 3 of Supplement 1 to ARINC Specification 653: Avionics Application Standard Software Interface*, published July 15, 2003. The specification became a standard in October, 2003.

LynxOS-178 conforms to the ARINC 653-1 Application Executive Software (APEX) Interface defined by the ARINC 653-1 standard. LynxOS-178 provides the following system service groups in accordance with the ARINC 653-1 standard:

- Partition Management—services related to partition management. `GET_PARTITION_STATUS` and `SET_PARTITION_MODE` are Partition Management service requests.
- Process Management—services related to process management. `GET_PROCESS_ID` and `GET_PROCESS_STATUS` are Process Management service requests.
- Time Management—services related to time management. `TIMED_WAIT` and `PERIODIC_WAIT` are Time Management service requests.
- Interpartition Communication—services responsible for communication between processes residing in different partitions. There are two types of Interpartition Communication services:
  - Sampling Port Services—A sampling port is a communication object allowing a partition to access a channel of communication configured to operate in sampling mode. `CREATE_SAMPLING_PORT` and `WRITE_SAMPLING_MESSAGE` are Sampling Port Services service requests.
  - Queuing Port Services—A queuing port is a communication object allowing a partition to access a channel of communication configured to operate in queuing mode. `CREATE_QUEUING_PORT` and `SEND_QUEUING_MESSAGE` are Queuing Port Services service requests.
- Intrapartition Communication—services responsible for communication between processes residing in the same partition. There are four types of Intrapartition Communication service requests:
  - Buffer Services—A buffer is a communication object used by processes of a same partition to send or receive messages. `CREATE_BUFFER` and `SEND_BUFFER` are Buffer Services service requests.
  - Blackboard Services—A blackboard is a communication object used by processes of the same partition to send or receive messages. `CREATE_BLACKBOARD` and `DISPLAY_BLACKBOARD` are Blackboard Services service requests.

  - Semaphore Services—A counting semaphore is a synchronization object commonly used to provide access to partition resources. `CREATE_SEMAPHORE` and `WAIT_SEMAPHORE` are Semaphore Service service requests.
  - Event Services—An event is a synchronization object used to notify the occurrence of a condition to processes that may wait for it. `CREATE_EVENT` and `SET_EVENT` are Event Services service requests.
- Health Monitoring—The Health Monitor (HM) is invoked by an application calling the `RAISE_APPLICATION_ERROR` service or by the OS or hardware detecting a fault.

POSIX

POSIX—the Portable Operating System Interface—is a family of standards designed to ensure source-code portability of application programs across hardware and operating systems. POSIX was developed by the Institute of Electrical and Electronics Engineers (IEEE) and is recognized by the International Organization for Standardization (ISO) and American National Standards Institute (ANSI).

The POSIX standards provide for communication between an application and the underlying operating system. Because POSIX conformance ensures code portability between systems, it is increasingly mandated for commercial applications and government contracts.

LynxOS-178 offers POSIX.1 conformance and also supplies all the services specified by POSIX 1.b (real-time extensions) and POSIX 1.c (threads extensions). The POSIX real-time and thread extensions are later additions to the original POSIX.1 standard, and they have extensive applicability for real-time and embedded systems.

The real-time extensions include priority scheduling, real-time signals, clocks and timers, semaphores, message passing, shared memory, asynch and synch I/O, and memory locking. The threads extensions include specifications for thread creation, control, and cleanup; thread scheduling; thread synchronization; and signal handling.

4 LynxOS-178 Architecture

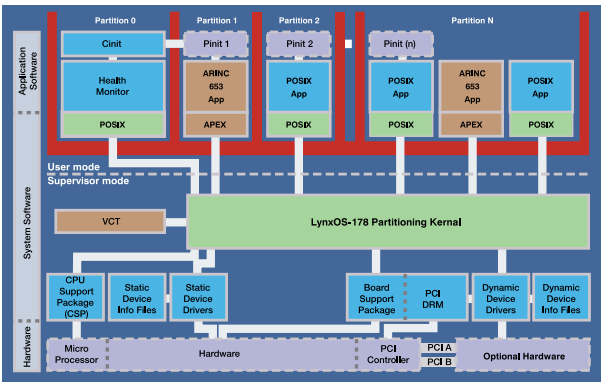
The software that runs on the target system is generally classified as either System Software (that is, the LynxOS-178 operating system) or Application Software.

System Software includes parts of the operating system, such as device drivers, which execute in processor supervisor mode. It also includes System Applications, such as `cinit`, which executes with operating system “root” privileges.

Application Software is the functional software that executes within a partition on the target system. Application Software always executes with operating system “user” privileges and is verified to the DO-178 criticality level appropriate for the intended function.

LynxOS-178 Operating System

The following diagram shows the LynxOS-178 software architecture.



NOTE: LynxOS-178 comprises all the components that appear in the region labeled System Software in this diagram.

The components labeled POSIX (System Services) and LOS-178 kernel are the reusable software components.

The CSP, device drivers, BSP, and configuration tables may be different on different boards or microprocessors.

The Application Software is usually supplied by the system integrator.

LynxOS-178 is a UNIX-style POSIX-compliant operating system designed to allow multiple real-time applications of different criticality levels to execute concurrently on the same processor. The LynxOS-178 provides time, memory, and resource partitioning to applications by preventing an application from affecting, or being affected by, any other application.

The LynxOS-178 operating system is designed to be independent of its underlying hardware platform. A unique Board support package (BSP) and CPU support package (CSP) provide the hardware-specific services to LynxOS-178. The application's only interaction with LynxOS-178 is through its documented Application Programming Interface (API).

LynxOS-178 also handles errors and exception conditions that applications do not or cannot trap.

The following sections explain the parts of the LynxOS-178 architecture illustrated in the previous figure.

Boot Code

The Boot Code boots the host processor and performs the appropriate level of power on self test (POST) to assure correct operating conditions of a limited set of hardware devices. The Boot Code is in the firmware module on the Thales VMPC6x board.

CPU Support Package (CSP)

The CSP contains all the processor family-specific routines, including the MMU, Floating Point, and processor exception handlers. The CSP routines are linked with the LynxOS-178 kernel.

Board Support Package (BSP) and Device Resource Manager (DRM)

The BSP contains routines for initializing and controlling hardware on the target system. The primary responsibilities of the BSP are:

- Interface with Boot and Shutdown software
- Establish virtual address map for onboard I/O
- Interface with the interrupt controller
- Provide default handlers for error-signaling interrupts
- Interface with the PCI controller
- Interface with the system time (tick timer)

The PCI Device Resource Manager (DRM), shown with the BSP above, is platform-independent.

The primary responsibilities of the PCI DRM are:

- Locate the PCI devices
- Manage ownership of PCI devices
- Map devices into virtual address space
- Provide access to the PCI configuration space

The BSP and the DRM are linked with the LynxOS-178 kernel.

Static Device Drivers

The Static Device Drivers are software components that isolate specific details of hardware devices from Application Software components. Items such as hardware dependent interrupt handlers (for example, power warn and load shed) and kernel threads are added to the kernel with device drivers. Static device drivers are linked with the kernel.

Static Device Info Files

The Static Device Info Files are used to configure the Static Device Drivers for devices available in the target system. There are one or more info files per device driver. The static device info files are linked with the LynxOS-178 kernel.

Dynamic Device Drivers

The Dynamic Device Drivers are hardware access routines for optional devices on the target system. These device drivers are stored in the file system and installed after the LynxOS-178 kernel is booted, but before partitioning is invoked.

Dynamic Device Info Files

The Dynamic Device Information Files are used to configure the Dynamic Device Drivers for optional devices on the target system. There can be one or more information files per device driver. These device info files are stored in the file system and installed after the LynxOS-178 kernel is booted, but before partitioning is invoked.

POSIX (System Services)

The system services are linked with the application code (C or C++) and run in processor user mode.

Application Programming Interfaces (API) include:

- POSIX API—Provides POSIX 1003.1, 1003.1b, and 1003.1c operating system services.
- File System—Provides a file system with a POSIX API.
- IEEE Floating Point Services—Provide services to configure floating point responses.

System Admin Services, available to VM0 (that is, VM “zero”) only, include:

- File System Admin services—`mount`, `ffsck`, `mkffs`

- Scheduler service—`create pinit()`, `initialize scheduler`, and others.
- High Water Mark services—`get_resource_entry()`
- LynxOS-178 Library—`dr_install()`, `setgroups()`

Kernel

The LynxOS-178 kernel is statically linked with the CSP, BSP, and Static Device Drivers to create the LynxOS-178 operating system. During initialization, Dynamic Device Drivers are dynamically linked with LynxOS-178 and effectively become part of the operating system.

Common Initialization (cinit)

`cinit` is the first POSIX process to run after the LynxOS-178 kernel is initialized. `cinit` executes with operating system root privileges. It reads the Virtual Machine Configuration Table (VCT) and creates VM partitions within LynxOS-178.

The primary responsibilities of `cinit` are as follows.

- Validate and read the VCT.
- Load Dynamic Device Drivers.
- Initialize system wide environment variables.
- Mount the file systems.
- Initialize the scheduler.
- Respond to partition fatal errors as defined in the VCT.

Partition Initialization (Pinit)

At the point in the LynxOS-178 initialization where the OS is able to run partitions, `cinit` transforms into a unique `Pinit` process in each partition. `Pinit`, as the first process in the partition, completes initialization of the partition's environment and transforms into the Application Software for the partition. `Pinit` executes with operating system root privileges.

Components Created by the User

In addition to the Application software itself, the Systems Integrator creates files and binary images that are part of a fully functional software configuration.

Virtual Machine Configuration Table (VCT)

The Virtual Machine Configuration Table (VCT) contains configuration information to create VMs within LynxOS-178. Each VCT also contains a Virtual Machine/Partition configuration profile for each software application. This information is used to allocate system resources to the application software, defining a valid configuration of the target system as determined by the user.

The VCT contains information based on the set of applications loaded on the target system.

Kernel Downloadable Image (KDI)

A Kernel Downloadable Image (KDI) is a single downloadable image containing LynxOS-178 and a file system. The file system is a UNIX-style root file system with a POSIX API. It contains the minimum system software necessary for `cinit` to complete its initialization tasks. It also contains file system mount points for all other file systems used by any VM. A RAM disk is created on `/mnt` and mounted for each VM partition. All file systems specified in the VCT are mounted in directories created on the RAM disk. Read-write file systems are mounted only for the VM partition that owns the file system. Read-only file systems are mounted for all partitions. This way, the read-write file systems are visible to only the VM partition that owns the file system, and read-only file systems are visible to all VM partitions.

5 Development Tools

Along with LynxOS-178, LynuxWorks provides powerful development tools to produce software that is high in quality and unquestionably reliable, and to make the development timelines for that software more efficient than ever before.

LynxOS-178 supports the following layered development tools:

- TotalView—a source-level, window-oriented, embedded debugger designed specifically for today's distributed multitasking environments
- VisualLynux—a cross-development tool that brings the functionality and convenience of Microsoft Visual Studio to embedded-system developers creating real-time applications
- SpyKer—a dynamically instrumented system trace tool to debug, diagnose, and optimize system performance
- Luminosity—an Eclipse-based IDE providing an enhanced tool set that enables embedded system developers to accelerate product time-to-market in the aerospace, telecommunications, and military sectors

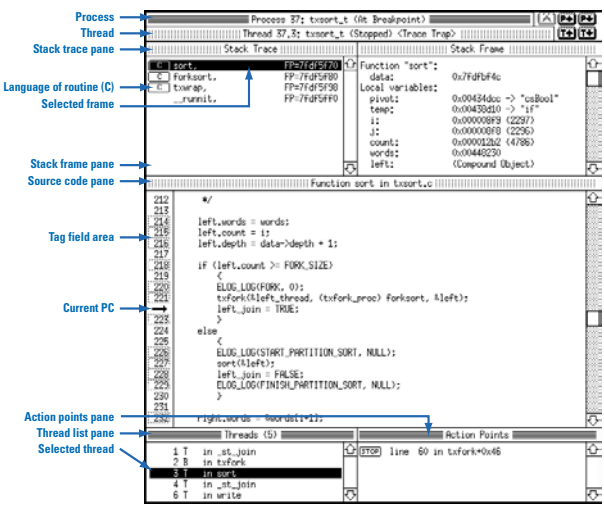
TotalView, VisualLynux, SpyKer, and Luminosity are described in more detail in the pages that follow.

TotalView

The TotalView debugger is part of a suite of software development tools for debugging, analyzing, and tuning the performance of programs, including multiprocess multithreaded programs and for performing multipartition and multinode debugging.

TotalView supports cross-debugging of user-level applications and the LynxOS-178 kernel. Cross-debugging is similar to remote debugging and distributed debugging except that with cross-debugging the host machine and the target machines are not of the same type. For debugging of user-level applications, TotalView communicates with the TotalView Debugger Server process run on the target machine either via TCP/IP or using an RS232 serial line. For debugging of the LynxOS-178 kernel, the TotalView client communicates with the LynxOS-178 Simple Kernel DeBugger (SKDB) using a dedicated serial line connection.

The TotalView debugging session window is shown below with the components it contains.



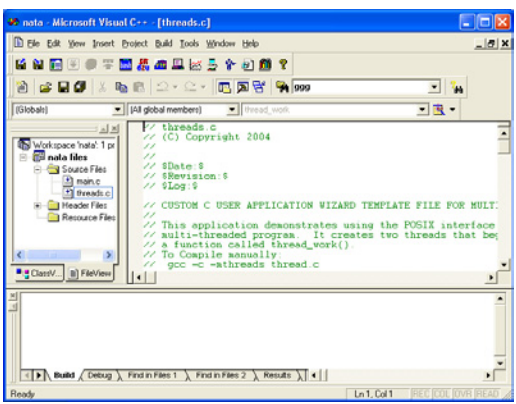
VisualLynux

VisualLynux is a cross-development tool with a graphical user interface (GUI). It expands the functionality of Microsoft Visual Studio 6.0 and .NET to provide users the ability to build and debug real-time embedded applications on Microsoft Windows hosts for target platforms running the LynxOS-178 operating system.

A VisualLynux project can contain multiple target-and-host build configurations. Each build configuration indicates an associated BSP and cross-development tools, such as the CYGWIN package, compilers, linkers, shells, and utilities. Developers can set up and troubleshoot their installed configurations via a target platform administrator. The FTP tool transfers files between the host and target, while a cross process viewer can be used to monitor processes and threads on the target. The BTP utility runs Bootp, Tftp, and Pftp servers on the host to boot targets via TCP/IP or a parallel port.

The LynuxWorks Cross Process Viewer (CPV) is a part of VisualLynux. The CPV runs on Windows cross-development hosts and communicates with the LynxOS-178 target machine to display all processes and threads running on the target.

VisualLynux provides developers with off-the-shelf POSIX template code. C applications programmers can be instantly productive using selectable templates of POSIX facilities. These POSIX component functions are included as part of the LynuxWorks long-standard commitment to standards support. The following screen shot shows a VisualLynux GUI window with a new project created using POSIX multithread sample code.



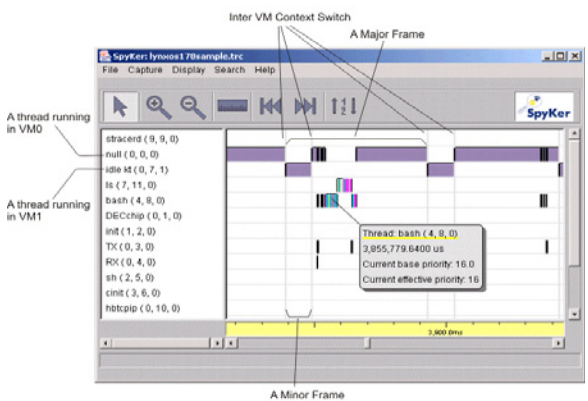
SpyKer

SpyKer belongs to a class of software development tools known as tracing tools. Tracing tools typically provide a system developer the ability to record, capture, and display operating system and application events on a running system. By capturing event trace data and displaying it with an easy-to-read GUI, trace tools provide system developers with powerful system tuning and debugging capabilities. The unique feature of SpyKer is its ability to provide dynamic event tracing without requiring an instrumented LynxOS-178 kernel. The SpyKer GUI detects that the target system is LynxOS-178 and provides the user with a VM-partitioned LynxOS-178-specific view of the target system.

The SpyKer LynxOS-178 target trace contains events for all VMs running on the target. The LynxOS-178 target trace contains the following LynxOS-178-specific information:

- The VM that generated each particular event
- The current consumption of per-VM resources at a given point of time
- Initial per-VM recourse limits assigned for each VM running on the target
- Start/run-time interpartition schedules
- Inter- and intra-VM scheduling events (end of a minor frame, end of a major frame, and so on)

The following figure demonstrates a LynxOS-178 target trace shown in the SpyKer GUI window.



Luminosity

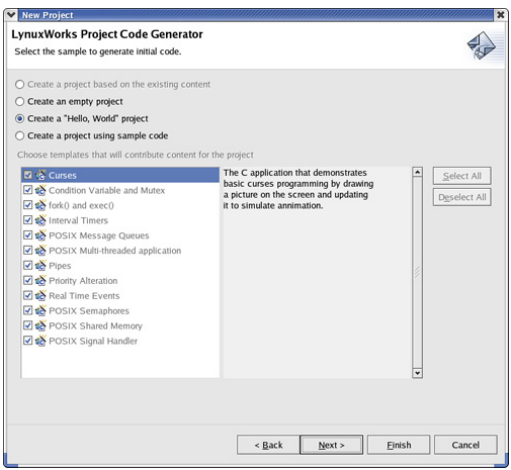
Luminosity provides developers a simplified, flexible platform to accelerate the time-to-market of their embedded system products.

Luminosity is a Linux- and Solaris-based Integrated Development Environment (IDE) powered by the Eclipse IDE. It gives developers complete control over creating, editing, compiling, managing, and debugging C/C++ and Java embedded and real-time applications. With Luminosity, developers can select a project wizard to jumpstart development and easily view all process and thread activity taking place on the target through the system viewer.

Luminosity offers developers a variety of programs and options, including the following:

- Platform Administration for configuring and managing different cross-development environments
- Target Administration for defining the exact hardware target to use
- Build Engine for specifying options for debugging and building final projects
- Project Wizards for easy creation of project coding frameworks (C/C++)

Luminosity's LynuxWorks Project Code Generator dialog is shown below.



6 Success Stories

Bombardier Challenger 300

The “super-midsize” Bombardier Challenger 300 business jet was officially launched at the 1999 Paris Air Show, as the Bombardier Continental Business Jet. Rockwell Collins was responsible for all aspects of the aircraft’s avionics program. Their advanced avionics system onboard the eight-passenger aircraft has received U.S. Federal Aviation Administration and Transport Canada certification.

Rockwell Collins is using LynxOS-178 as the embedded RTOS in the Challenger 300’s adaptive flight display systems.

Aérospatiale Airbus Flight-Control Systems

In 1995, Aérospatiale of France, a founding partner of EADS, embarked on a long-term project to develop the Air Traffic Services Unit (ATSU) computer for Airbus 320, 330, and 340 aircraft, part of the Future Air Navigation Systems (FANS) program. The Aérospatiale Air Traffic Services Unit (ATSU) integrates all services dedicated to data communication between the aircraft and the air traffic centers, all the needed resources to implement the future ATN Network, as well as the present Aircraft Communication Addressing and Reporting System functions (ACARS). This project marked the first time that a commercial RTOS would be used in a computer-controlled navigation system certified to DO-178B level C.

Aérospatiale chose the LynxOS RTOS to power these next-generation Airbus flight-control systems after evaluating dozens of RTOS suppliers during a year-long review process. Aérospatiale finally chose LynxOS for price, performance, reliability, and portability.

Raytheon MK 57 Vertical Launching System DD(X)

The MK 57 Advanced Vertical Launching System (AVLS) is the next-generation naval missile launching system for future surface combatants of the U.S. Navy. The MK 57 is being designed for DD(X)—the next-generation destroyer now being developed by the DD(X) National Team. The MK 57 can rapidly launch of missiles into a 360-degree hemispherical volume in support of multiple DD(X) mission areas including Land Attack Warfare, Integrated Air and Surface Dominance, and Integrated Undersea Dominance. Raytheon Integrated Defense Systems serves as the ship electronics and weapons integrator for DD(X), and the MK 57 development is led by Raytheon in partnership with United Defense, L.P.

The LynxOS RTOS is used extensively by the DD(X) program in multiple areas, such as in ships' embedded distribution application processors (DAPs). The DAPs control key interfaces of the ship, such as the missile launching equipment, the ship's propulsion equipment, and external communications equipment.

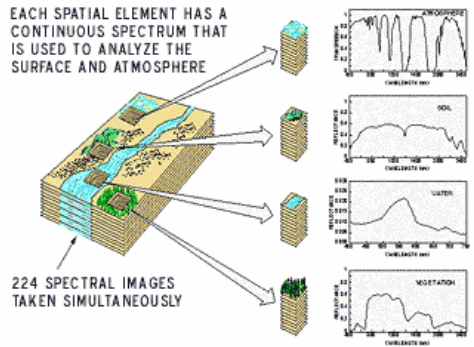
Common ARTS Air Traffic Control Systems

Automated Radar Terminal Systems (ARTS) manage air-traffic control at U.S. airport locations such as New York, Dallas/Fort Worth, Chicago, Southern California, and Atlanta. The Common ARTS program sought to develop a common software baseline at the Nation's Terminal Radar Approach Controls (TRACONs) and incorporate more COTS products. Common ARTS has evolved into a real-time distributed open-systems architecture that is multithreaded, and that uses a local area network. Thousands of tracks are supported, and so are hundreds of displays. Because Common ARTS is a safety-critical application, a standby system is always ready to take control if any subsystem ever ceases to broadcast its heartbeat on the network.

Programs are written in C language and run under the LynxOS RTOS. LynxOS provides POSIX conformance, which increases ease of porting to other systems. Though the system currently operates on a PowerPC platform, the software formerly ran on Motorola 68K CPUs. LynxOS is relied upon for its absolute reliability in this safety-critical system.

NASA's Airborne Visible/Infrared Imaging Spectrometer

The Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) earth remote-sensing instrument scans the ground below while it flies aboard Twin Otter and ER-2 aircraft. Its 224 wavelength-sensitive detectors obtain spectral data that will be processed for display as images and to reveal information about the viewed area's composition. AVIRIS was developed for NASA at the California Institute of Technology's Jet Propulsion Laboratory. AVIRIS first went into service in 1987, three years after the start of design and construction.



AVIRIS operates under a mission-critical, nonstop, digital control subsystem, at the heart of which the LynxOS RTOS runs on an Intel 8085A-2 microprocessor. An additional role of this control subsystem is to interface with the plane's navigation computer so that flight parameter data can be recorded along with the data obtained by the AVIRIS instrument.

SLR2000 Satellite Laser Ranging System

SLR2000 is NASA's fully automated, next-generation Satellite Laser Ranging (SLR) station, currently under development. SLR stations around the globe measure the round-trip time of ultrashort light pulses sent to satellites equipped with special reflectors. The measurements recorded are used for a host of important science products.

The LynxOS RTOS will be running inside the pseudo operator of the manless SLR2000. The pseudo operator will monitor the health and safety of the system and will control the acquisition and tracking of the satellites. LynxOS will run on a Pentium 200 single-board computer mounted in a VME bus chassis, which will communicate between the interface control computer and the data analysis computer using a Bit-3 PCI-to-VME bus adapter.

LynxOS can also be found inside all of NASA's existing SLR stations—from the MOBLAS-8 in Tahiti to the TLRS-3 in Peru.

AN/USQ-78A and AN/USQ-78B Display Control Units

Aboard many of the U.S. Navy's P-3C Orion maritime surveillance aircraft, AN/USQ-78A and AN/USQ-78B Display Control Units (DCUs) operate as integral components of the aircraft's Single Advanced Signal Processor (SASP). The system is responsible for command and control of the signal processors, drives the displays and controls for two sensor stations, provides acoustic contacts to the Tactical Subsystem, and communicates with other acoustics subsystems.

AN/USQ-78A and AN/USQ-78B DCUs have been installed in the context of the U.S. Navy's Block Modification Upgrade Program (BMUP) and its Anti-surface Warfare Improvement Program (AIP).

At the heart of each AN/USQ-78A or AN/USQ-78B DCU, the LynxOS RTOS can be found running on a Cetia COTS processor. LynxOS was chosen because it is a hard real-time, POSIX-conformant operating system.

ICI's Improved Data Modem

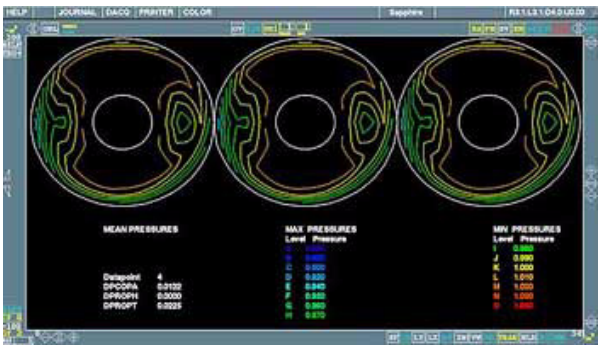
ICI's Improved Data Modem (IDM) is a communications and targeting system that can interface between the different communications formats in use by the U.S. Army and the U.S. Air Force. An RTOS resides inside the IDM. Future plans called for incorporating in the IDM an embedded subset of the Army's Force XXI Battle Command, Brigade and Below (FBCB2) system software.

Historically, FBCB2 software ran on the Solaris operating system, while VxWorks was being used in the IDM. After three years of unsuccessful attempts to port the FBCB2 software to VxWorks, the original project team cut its losses and turned to the LynxOS RTOS.

In six months, the port to LynxOS succeeded.

M6000 Data Acquisition System

The Cranfield Data Systems M6000 data acquisition system is a multifunction signal processing and monitoring unit designed for a variety of engineering environments, including aircraft and automotive systems that permit computationally intensive signal analysis to occur simultaneously with continuous high-throughput data acquisition. The product is expandable from 32 to 256 input channels, and multiple systems can be networked using high-speed fiber optics for an unlimited channel count.



CDS data acquisition software applications were originally designed to run on a proprietary RTOS and proprietary processor architecture.

After eventually considering COTS products, CDS decided to migrate the M6000 to the LynxOS RTOS, running on a PowerPC G4 processor. Migration went smoothly and resulted in a competitive and profitable product.

Boeing 777 Cabin Services System

In 1998, Boeing awarded a long-term contract to LynuxWorks for the LynxOS RTOS software to be used in the cabin services system aboard Boeing 777s, their newest and most technically advanced commercial jetliner.

The cabin services system automates many tasks that flight attendants previously performed manually and allows easy control of lighting, boarding music, address system, and cabin doors. The cabin services system has improved the display of system faults by consolidating them on one panel rather than on individual panels above each seat. A better response time of maintenance crews servicing the aircraft has resulted.

LynxOS X and Motif subsystems handle the entire user interface. Boeing chose LynxOS for its support of numerous off-the-shelf devices and for its strong adherence to industry standards. Standards allow programmers to port code rapidly from other operating systems to LynxOS.

DO-178B and GATM USAF C/KC-135 Upgrades

When the United States Air Force (USAF) decided to update its C/KC-135s to GATM guidelines and the DO-178B standard, electronic displays and Control/Display Unit (CDU)-based flight management equipment were modified to be compatible with Rockwell Collins Integrated Processing Center (IPC) units. Most processing now occurs in the IPC, which provides multiple modular processing resources for hosting mission, flight-management, and display-management processing functions. DO-178 certification was achieved.

The IPC is interfaced to an avionics-quality Ethernet Local Area Network (LAN). Inside the IPC cabinet is a series of Line Replaceable Modules (LRMs). Of these modules, the LynxOS-178 certifiable, safety-critical RTOS powers the IPC Common Computing Module (CCM) and the Input/Output Concentrator Module (IOC).

U.S. Army Crusader

The US Army Crusader field artillery system is a two-vehicle system that consists of a fully automated 155mm self-propelled howitzer and a resupply vehicle. Crusader can fire up to 12 rounds per minute and can travel at cross-country speeds of 48 kilometers per hour, highway speeds of 78 kilometers per hour, and dash speeds of 750 meters in 90 seconds. Crusader was conceived to replace the Paladin howitzer cannon.

LynxOS was chosen to provide on-board processing of required functions such as Crusader's ballistic calculations, the graphical user interface and real-time equipment control systems.

When LynxOS was selected, it already supported the microprocessors and specialty boards that were running on the Crusader, thus eliminating the need to port an OS to new hardware.

All-Purpose Remote Transport System

The All-Purpose Remote Transport System (ARTS) is a robotics system for remote-controlled vehicles in high-risk environments. An operator-control unit (OCU) is used to control the engine, propulsion, lights, and tools of an unmanned utility tractor. The tractor communicates audio and video back to the OCU and its operator. ARTS systems have been deployed for hazardous material handling, fire fighting, natural-disaster clean-up, and mine clearing in Afghanistan.

The LynxOS RTOS runs the ARTS OCU, which is packaged in a weather-tight, shock-isolated enclosure. The OCU is comprised of a radio for transmitting command signals to the tractor, a receiver for audio/video signals transmitted from the tractor, a joystick control box, and a television monitor with built-in videocassette recorder.

C3I Upgrades for Flyvefisken Class Ships

Flyvefisken Class ships (also known as Standard Flex 300 or SF300) are modularly designed with containerized weapon systems and equipment. As a result, these Royal Danish Navy vessels can quickly change roles for diverse functions such as surveillance, surface combat, antisubmarine warfare (ASW), mine countermeasures, or pollution control. Standard equipment for all functions includes sonars, radars, and the integrated C3I system (command control and communication information).

The LynxOS RTOS was selected for a Flyvefisken C3I system upgrade by Danish contractor Terma A/S. LynxOS was the only RTOS capable of satisfying all the project requirements, which included binary compatibility with Linux-based battlefield simulation software, support for embedded Java and Java/C/C++ applications, and field-proven high stability.

1.800.255.5969

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

LynuxWorks Europe
2, allée de la Fresnerie
78330 Fontenay-le-Fleury
France
+33 1 30 85 06 00
+33 1 30 85 06 06 fax