Green Hills Software

INTEGRITY®-178

The only operating system compliant to both SKPP/EAL 6+ and RTCA/DO-178B Level A

Green Hills® SOFTWARE

The proven provider of safety & security solutions

Since 1997, a dedicated group at Green Hills Software has focused on safety and security software solutions for platform and application development. As a proven provider of software and certification solutions, this group has handled the risks associated with operating systems, run-times, common service libraries, and the development tool-chain. With a broad product line that spans different processors and certification types, Green Hills Software has repeatedly demonstrated its capability for managing these risks.

Supporting the entire certification effort

Our certification approach is to provide proven software system solutions with completed security certificates and safety compliance approvals—not just claims of being "certifiable." Green Hills Software's unique solutions combine customer-specific certification support with common software capabilities that are reused across a multitude of customers. Green Hills Software supports a customer's entire certification effort, including hardware/software product compatibility, custom hardware device driver development, complete product testing on customer's hardware, appropriate life cycle data generation and delivery, and audit support. This frees customers to focus on their core competencies, lowering the schedule, cost, and certification risks associated with using internal or multi-supplier resources.

A complete safety critical line

Green Hills Software offers a full line of safety & security critical products that are available today with complete DO-178 Level A certification evidence. This includes:

- **INTEGRITY®-178**—a complete time, space, and resource partitioned real time operating system (RTOS)
- **ANSI C Library**—an ANSI C Library subset
- **Embedded C++**—a C++ Library subset
- **GMART**—a single-tasking Ada run-time based on the SPARK Ada profile
- **GSTART**—a multi-tasking Ada run-time based on the Ravenscar Ada profile
- **GMART Bare Target**—an Ada bare target run-time, used with GMART

The Rockwell Collins Avionics Management and Display System aboard the S-92 Sikorsky helicopter is among the INTEGRITY-178 applications approved as compliant to DO-178B Level A, the most stringent safety assurance level for avionics software.

- **GCERT Bare Target**—a C bare target run-time, used with the ANSI C Library
- **ARINC-653**—ARINC-653 compliant Part 1 APEX interface and Part 2 file system
- **PJFS-178**—a file and directory management system and user interface supporting physical and virtual storage devices
- **IPFLITE/TFTP**—UDP/IP network stack and socket library interface, including support for TFTP capabilities
- **Audit Logging**—logging and retrieval of kernel and application triggered events during RTOS execution
- **Secure Hash Algorithm (SHA-1)**—verifies integrity of ELF images at RTOS startup and during run-time
- **Abstract Machine Testing**—verifies correct operation of hardware at RTOS startup and during run-time

INTEGRITY-178B RTOS Features

- **Safe and Secure By Design**
  - Real-Time Operating System designed for use in reliable, mission critical, safety critical and secure (MILS & MLS) applications
  - Based on modern microkernel RTOS design
  - Fast, deterministic behavior with absolute minimum interrupt latencies
- **Complete Application Partitioning**
  - Space & time & resources
- **Robust Hardware Architecture**
  - Hardware abstraction layer
  - PowerPC Architectures, MIPs, ARM, x86 support
  - Multi-core support
- **Robust Flexible Software Architecture**
  - Pure RMA
  - Pure ARINC-653
  - Enhanced Partition Scheduling
  - Asymmetric and Symmetric multi-processing
  - Guest OS Virtualization
- **Middleware Support**
  - ARINC-653
  - PJFS-178B
  - IPFLITE / TFTP
  - GHNet-178
  - POSIX
  - ARINC-615A Data Loader
- **Full and Certified Subset Language Support**
  - Ada, C, C++

Within an INTEGRITY-178 partitioned environment, additional Green Hills products, tailored for applications that can use commercial off-the-shelf software without detailed and rigorous certification evidence (e.g., DO-178B Levels D/E) are supported:

- **ARINC-615A Data Loader**—ARINC-615A data loading support
- **POSIX**—a POSIX run-time Library subset
- **C/C++**—a full C/C++ run-time Library
- **Ada**—a full Ada run-time Library
- **File System**—a FAT file system
- **Networking**—GHNet-178 TCP/IP IPV4/IPV6 stack, advanced router stack, Net-SNMP stack, IPSec library, and other networking related capabilities

Proven experience

Green Hills Software Safety & Security development center of excellence is located in Palm Harbor, Florida. The staff has significant experience in the commercial and military avionics industry, providing customers with dedicated experts that understand industry perspectives.

All certification activities—including independent verification— are performed in-house. This single-supplier approach simplifies customer oversight and reduces the time and number of contacts required to resolve integration issues. Our dedicated in-house verification team ensures customers that the DO-178B independence objectives are satisfied and also provides direct access to the experts needed for robust and complete test development. The Green Hills staff also includes an FAA authorized Designated Engineering Representative (DER), to provide assistance during the certification process and ensure customer-directed compliance activities and audits are both understood and completed.

RTOS life-cycle data is generated and verification activities are performed for each specific certification program and its associated target hardware platform(s). Each certification effort takes full advantage of reuse, traceability, and pedigree from appropriate prior certifications. Green Hills also provides Board Support Package (BSP) and device driver development and certification services, freeing customers from RTOS and low-level hardware interface issues.

For projects that must be free from foreign influence, all Palm Harbor certification activities—including development and verification—are completed by US citizens.

Proven pedigree

The INTEGRITY-178 RTOS has earned its pedigree through a unique combination of powerful capabilities:

- A single partitioning-supporting operating system that satisfies both DO-178B Level A safety assurance requirements and NSA High Robustness security functional and assurance requirements
- Proven in real-world customer applications since 1997 with over 60 certification packages developed for more than 30 different microprocessors delivered to date
- The first commercial partition-enforcing RTOS approved as complying to DO-178B Level A objectives (2002)
- The first RTOS to obtain SKPP/EAL 6+ certification (2008)
- Highly scrutinized RTOS source code—perhaps the most scrutinized to date

National Information Assurance Partnership

Common Criteria Certificate

is awarded to

Green Hills Software, Inc.

The IT product identified in this certificate has been evaluated at an accredited testing laboratory using the Common Methodology for IT Security Evaluation (Versions 2.3, 3.0, 3.1) for conformance to the Common Criteria for IT Security Evaluation (Version 2.3). This certificate applies only to the specific version(s) and release of the product in its evaluated configuration. The product's functional and assurance security specifications are contained in its security target. The evaluation has been conducted in accordance with the provisions of the NIAP Common Criteria Evaluation and Validation Scheme and the conclusions of the testing laboratory in the evaluation technical report are consistent with the evidence adduced. This certificate is not an endorsement of the IT product by any agency of the U.S. Government and no warranty of the IT product is either expressed or implied.

Product Name: INTEGRITY-178B Separation Kernel
Evaluation Platform: INTEGRITY-178B Real Time Operating System (RTOS), version IN-ICR750-0402-GH01_Rel (Version 4.2) running on Compact PCI card, version CPN 944-2021-021 w/PowerPC, version 750CXe
Assurance Level: EAL 6+, High Robustness

CCTL: Science Applications International Corporation
Validation Report Number: CCEVS-VR-VID10362-2011
Date Issued: 31 January 2011
Protection Profile: US Government Protection Profile for Separation Kernels in Environments Requiring High Robustness, Version 1.03, 29 June 2007

Original Signed By
Director, Common Criteria Evaluation and Validation Scheme
National Information Assurance Partnership

Original Signed By
Information Assurance Director
National Security Agency

The INTEGRITY-178 RTOS SKPP/EAL 6+ Certificate

EAL6+ High Robustness

In July 2007, a new U.S. government sponsored protection profile (PP) titled "U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness" (SKPP) was completed. This PP defined the required functional capabilities and assurance activities for a high-assurance separation kernel product as follows:

"A separation kernel evaluated against this PP provides a highly robust foundation for system services and applications in mission-critical embedded systems, and a high degree of assurance for the enforcement of related security policies. Such policies include those for the management of classified and other high-valued information, whose confidentiality, integrity or releasability must be protected."

According to Department of Defense (DOD) guidance, High Robustness refers to "security services and mechanisms that provide the most stringent protection and rigorous security countermeasures."

In September of 2008, after rigorous evaluation by the National Information Assurance Partnership (NIAP) and the National Security Agency (NSA), NIAP issued to Green Hills Software a certificate of conformance to the SKPP for the Green Hills INTEGRITY-178 Operating System product. As stated in the SKPP, "A high robustness TOE is necessary protection for environments where the presence of both sophisticated threat agents and high value resources makes the likelihood of an attempted compromise high."

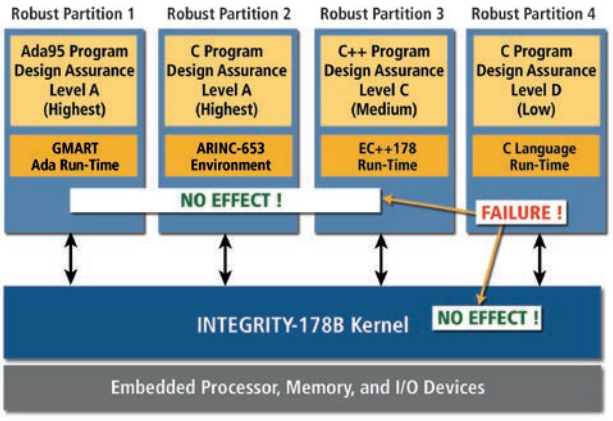
Certification to EAL 6+/High Robustness means that INTEGRITY-178 can provide this level of threat protection as a foundation for other security products. The highest security standard to which any other operating system (Windows®, Linux™, Solaris® and others) has ever been certified only protects against "inadvertent or casual attempts to breach the system security."

In September of 2011, primary security assurance responsibilities associated with Separation Kernels changed from NIAP to individual government systems where it is managed as part of the system security assurance. Green Hills continues to provide the same level of assurance activities and support to each system that needs an operating system capable of supporting the "most stringent protection and rigorous security countermeasures." This level of commitment has been repeatedly demonstrated through the multiple security assurance maintenance deliveries completed to date, making the INTEGRITY-178 RTOS the right choice to include as part of a security based system.

Partitioning operating systems

The need to reduce maintenance costs and size/weight/power—as well as the growing capabilities of modern processors—created a demand for commercial partitioning operating systems (OS) that support multiple applications at different safety and/or security levels running on a single processor. A partitioning OS also must support resource allocation, fault detection, and fault isolation to prevent unintended interactions between independent applications.

The cost to test and certify safety critical software is directly proportional to the level of safety criticality. The higher the design assurance level, the more complex and expensive the certification process. An economical and architecturally ideal single-processor solution is to certify to the highest level only the applications that run at the highest criticality level. Applications or functions operating at lower criticality levels are certified to lower levels. For the operating system this means certifying to a level at least equal to the highest criticality level application or function running on the processor.



The INTEGRITY-178 RTOS guarantees that failures resulting from a defect in a program operating within one partition CAN NOT disrupt the operation of programs assigned to other partitions.

This method is valid as long as the OS guarantees that any failure resulting from a defect in one application CAN NOT, under any circumstance, have unplanned disruption in the operation of other applications, especially the higher safety level applications. The operating system must guarantee protection in both the space and time domains. In other words, a commercial real-time operating system must support robust partitioning in order to provide this level of real-time scheduling and memory protection. For security related applications, the memory protection supported by the OS must also prevent breaches in data confidentiality.

Protection in time and space domains

INTEGRITY-178's unique approach to resource management provides guaranteed resource availability for multiple safety-critical and/or security-critical applications on a single processor operating at different safety assurance levels (A/B/C/D/E) and/or security levels.

Engineered from the ground up to provide security and determinism, the INTEGRITY-178 RTOS guarantees protection across both the time and space domains, including protecting the confidentiality and integrity of an application's data from unintended access by other applications.

To guarantee bounded computation times, the kernel does not depend on features such as dynamic memory allocation. Underlying hardware mechanisms are utilized to provide full system memory protection for all software components, including user applications, device drivers, and inter-partition communications. Protection of clocks and timers is guaranteed through access permissions and by using a supervisor mode hardware timer. Memory-protection and error-handling features provide a secure system with built-in fault isolation and tolerance. Traditional kernel access problems such as invalid kernel addresses and invalid system call parameters are eliminated.

The system designer defines a partition schedule that includes execution time windows for each partition:

- CPU resource is guaranteed when partition is active
- Normal priority preemptive task scheduler within a partition
- Multiple application partitions supported in each time window (facilitates implementing client-server functionality)
- Mode Change capability supports defining multiple schedules and switching between them
- Advanced synchronization of applications to start of time windows
- All scheduling tied to high resolution clock whose events are separate from system clock tick

Protection in the space domain

- **Guaranteed Resource Availability**—Partition's memory is protected from access by another partition
- **Memory Protection—**Utilizes underlying hardware MMU to enforce execute-read-write permissions
- **"Hard Currency" OS**—No shared resource pools, each partition is individually allocated resources for system calls
- **Statically verifiable MMU settings**—No dynamic manipulation of MMU to support message passing
- **Statically Verifiable System Resource Allocation**— Project defined boot table controls ownership
- **Connections**—Secure (non-bypassable) inter-partition communications

Protection in the time domain

- **Deterministic**
  - Given a state & input results in the same state transition
- **Scheduler/Timing Analysis**
  - No heuristics in scheduler
- **No Priority Inversion**
  - No binary semaphores in kernel implementation
  - Provide support for Highest Locker Semaphores, hence no unbounded blocking times
- **ARINC-653-1 Partition Scheduler**
  - Optimized two-level scheduler
  - Guaranteed execution time windows
  - Execution overrun detection
- **Bounded Computation Time For All System Calls**
  - No dynamic memory allocation in kernel space
- **No hidden execution time/latency**
  - Message transfers use task's execution time
  - Interrupts never disabled to update kernel structures
- **Software Timers with Access Permissions**

Lowering certification costs by minimizing regression testing

INTEGRITY-178's ARINC-653-1-Application/EXecutive (APEX) interface provides a recognized standard interface between the operating system of an avionics computer resource (ACR) and the application software. INTEGRITY- 178's ability to fully support ARINC-653-1 while complying with DO-178B Level A provides a COTS baseline avionics operating environment that meets standards already adopted and accepted by the commercial avionics industry for Integrated Modular Avionics (IMA).

INTEGRITY-178 reduces the time and cost for adding new or modified applications into existing systems. Through robust partitioning in both time and space, minimal regression testing—often the most expensive activity of the certification effort—is required for the unchanged applications.

This reduced effort translates into large cost savings and decreased time-to-market. For systems without robust partitioning, regression testing and/or analysis must be performed to assess the impact of the new or modified applications on the continued correct operation and margins of the unchanged applications.

As a result, both functional and performance tests will likely be required for the entire system when an operating system does not support robust partitioning. INTEGRITY-178 customers may also use the Green Hills DO-178B qualified G-BootTable tool to simplify the change impact analysis of their application changes and resource assignments on the system configuration.

Layered product extensions

Certified Language Support

Application development using C, C++, and Ada is supported, including library subsets designed for DO-178B Level A (and lower levels), Multi-Level Secure (MLS), and Multiple Independent Level Secure (MILS) related applications. Programming language considerations defined in the *Handbook for Object-Oriented Technology in Aviation (OOTiA)* were used in the selection of the supported object-orientated library features.

PJFS-178

The PJFS-178 product is a high assurance, reliable file system designed for DO-178B Level A certification that supports both file and directory services. This small footprint client-server implementation provides power-failure safe access to a variety of underlying storage devices. The PJFS-178 Client provides a POSIX-based API which can reside in one or more application partitions. The PJFS-178 Server provides the implementation of the file system, handles simultaneous API requests from clients, and manages all physical and virtual file storage devices. The PJFS-178 server provides partitioning support when used by clients running in different partitions.

The PJFS-178 Server can manage multiple storage devices, divided into separate volumes, each with their own client access permissions. A journal of operations is used to guarantee file system integrity, and provide power-failure safe write operations. Start-up time is very fast as operations to 'check-disk' are not necessary. PJFS-178 can also be integrated with DO-178B Level A wear-leveling flash device drivers.

IPFLITE

The IPFLITE product provides a UDP/IP network system for DO-178B Level A certification. This lightweight client-server implementation provides reliable networking support for Ethernet connected devices. The IPFLITE Client provides a BSD-style socket API that can be used to access networking services. The IPFLITE Server provides the implementation of the network stack, handles simultaneous API requests from

clients, ARP requests/broadcasts, and manages multiple Ethernet devices. The IPFLITE server provides partitioning support when utilized by clients operating in different partitions.

Trivial File Transfer Protocol (TFTP)

The TFTP product provides TFTP services compatible with the INTEGRITY-178, IPFLITE, and PJFS-178 products. The TFTP product supports both read and write file transfer requests generated through either the TFTP API or by a foreign host.

ARINC 653 APEX

The ARINC 653 (Part 1) Required Services product satisfies the characteristics and interfaces defined for an operating system in ARINC 653 Part 1, *Avionics Application Software Standard Interface Part 1- Required Services*. The ARINC 653 library provides an ARINC 653 compliant API to the INTEGRITY-178 RTOS, and includes module-level and partition-level health monitoring capabilities. The supported language bindings permit DO-178B Level A (and lower levels), MLS, and MILS related ARINC 653 applications to be developed in Ada, C, or C++.

The ARINC 653 (Part 2) File System Interface product satisfies the characteristics and interfaces defined for a file system in ARINC 653 Part 2, *Avionics Application Software Standard Interface Part 2 - Extended Services*. The ARINC 653 File System Interface consists of a single high-level file system component that is used with INTEGRITY-178 and PJFS-178 to provide file system capabilities to ARINC 653 based applications.

ARINC 615A Data Loader

The ARINC 615A Data Loader product satisfies the target data loading characteristics and file formats defined in the ARINC 615A and ARINC 665 specifications. The library provides capabilities to upload/ download target hardware memory and retrieve configuration information from the target hardware. The ARINC 615A Data Loader is compatible with the INTEGRITY-178, IPFLITE, and IPFLITE TFTP products.

Bare Target Products

For projects that require very small footprints with no time and space partitioning requirements, Green Hills Software provides several products that run as bare target runtimes. A bare target run-time executes directly on the underlying processor (i.e., no operating system involved in multi-partition, multi-tasking, or hardware abstractions).

The GMART Bare Target is a non-tasking, minimal, Ada language based run-time. GMART Bare Target is intended to be suitable for use by Ada application developers who are utilizing SPARK toolsets and development concepts.

The GCERT Bare Target is a non-tasking, minimal, C language based run-time. GCERT Bare Target provides a minimal run-time that supports the ANSI C Library subset.

Security product extensions

These products provide support for SKPP functional capabilities. The certification evidence also supports use of these products by DO-178B Level A (and lower levels) applications.

Audit Logging

The Audit Logging product supports the logging of a well-defined set of events during RTOS execution. A customer defined monitoring application uses the Audit Logging functionality to access the logged audit events to detect potentially malicious code behavior. INTEGRITY-178 includes definitions of the events being logged as part of RTOS execution. The logged events may only be read by applications explicitly authorized to access the audit log. A means to statically configure the system to exclude specific events on a partition and/or event basis is also supported.

Audit Logging also provides capabilities for authorized applications to reset the audit log, enable/disable audit logging, record specific events, and halt system operation.

Secure Hash Algorithm (SHA-1)

The Secure Hash Algorithm provides support for testing the integrity of ELF images at RTOS startup and after startup. SHA-1 can detect program memory failures that could result in loss of protective checks within INTEGRITY-178 (the partitioning architecture prevents malicious code in a partition from attempting to modify applications after load-time). The Green Hills implementation was designed to comply with the NIST/FIPS Secure Hash Standard (FIPS 180-2) that specifies the SHA-1 Algorithm.

A user application project's overall executable object code is constructed with digest information for each ELF image. The SHA-1 product provides the capability to detect integrity failures within these ELF images. When a failure is detected, an audit event is reported. The SHA-1 product is designed to operate with INTEGRITY-178 and the Audit Logging product.

Abstract Machine Test (AMT)

The Abstract Machine Test product provides support for testing and confirming the correct operation of the hardware at RTOS startup and on a periodic basis. AMT can detect hardware failures that could have allowed malicious code unauthorized access to hardware or software.

When a failure is detected, an audit event is logged and a failure is reported. The AMT product is designed to operate with INTEGRITY-178 and the Audit Logging product.

Resolving the complexity of multicore systems

To satisfy increasing demands for computing throughput, processor designers are adding multiple cores within a single chip package. Operating system support for multicore is a necessity in order to incorporate these devices into new designs and system upgrades.

Multicore design challenges

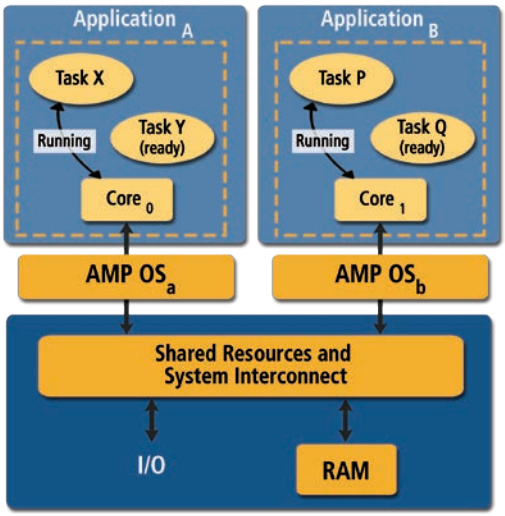
For developers of real-time embedded systems, the advent of multicore processors has resulted in several new design challenges.:

1. How to select an architecture that permits effective use of multicore processors.
2. How to develop, integrate, or port previously independent applications running on single-core processors to a multicore operating environment that includes shared system resources.
3. Finding an architecture that supports different use scenarios for different sets of processor cores.
4. Finding an architecture that provides the flexibility and tradeoffs necessary for eventual system certification on a multicore processor.

Most architectures, including the fundamental AMP and SMP scheduling approches, require developers to make tradeoffs when attempting to resolve these challeges. The INTEGRITY-178 RTOS uniquely provides a highly-flexible multicore solution that addresses them all.

Asymmetric Multiprocessing Architecture

The asymmetric multiprocessing (AMP) architecture is typically represented as multiple operating systems each in control of a single processor core. AMP scheduling is well suited for heterogeneous and homogeneous microprocessors. The significant scheduling behavior is that multiple independent applications can be running simultaneously on different processor cores. A system architect statically allocates applications to a given core and each core owns its own area of memory and may be allocated specific I/O devices. With AMP, each application is running in an independent operational environment on each core, providing a potential for a high degree of application parallelism.



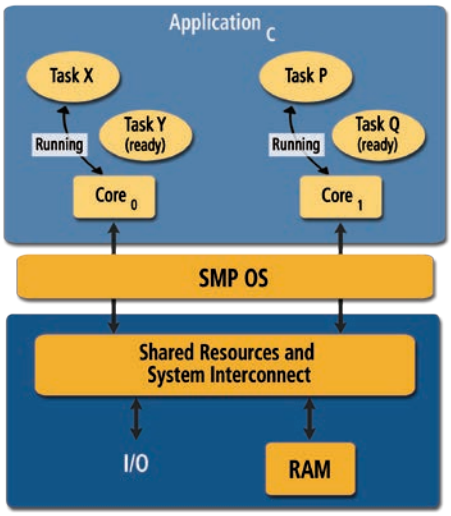
AMP: two separate OS each in control of one core.

If the OS used in an AMP architecture supports multi-tasking, there is only one core allocated to the application on which to schedule tasks. As such, only one task belonging to an application can run on the allocated core at any time. If each OS supports partitioning, each OS could be simultaneously executing a different partition.

In concept, porting existing single-core applications into an AMP architecture could consist of taking sets of applications that were previously running on individual single-core processors and porting each set to a different core in the multicore processor. Application growth remains limited to the resources available to the core that the set has been ported to. A significant challenge may involve resolving contention issues in the shared resources and system interconnect that may be accessed simultaneously by multiple applications. Changing an application may require moving other applications that were running on the same core to another core in order to provide sufficient time resources to the application being updated.

Symmetric Multiprocessing Architecture

The symmetric multiprocessing (SMP) architecture is typically represented as a single operating system controlling all of the multi-processor resources. SMP scheduling is best suited for homogeneous microprocessors. The SMP architec-



SMP: one OS runing two tasks of one application simultaneously on two cores.

ture permits the OS to select multiple tasks to run simultaneously on the available processor cores. When these tasks are associated with the same application, multiple cores are likely to have access to the same area of memory. With SMP, an application may run in an operational environment involving multiple cores, providing a potential for a high degree of task parallelism. The operating system determines which tasks should run on the available cores. A system architect may utilize core affinity techniques to bind tasks to run only on specific cores.

If an SMP OS supports partitioning, generally only one partition is active at any one time on the set of cores.

In concept, porting existing single-core applications into an SMP architecture could consist of taking one set on applications that were previously running on a single core processor and porting the set of applications to the multicore processor. Multi-tasking applications may take advantage of the multiple processor cores to perform some operations in parallel. Application growth permits multiple cores to be utilized simultaneously. When tasks are running simultaneously, contention issues in the shared resources and system interconnect may be limited to impacting only tasks belonging to the same application. Changing an application may result in utilizing additional cores to provide sufficient time resources. This may permit other ported applications to not be impacted by the changes.

Other General Multi-Processing Architectures

Additional multicore architectures are discussed in literature and papers for various industries. In general, all of these architectures make use of the AMP and/or SMP scheduling behaviors.

Heterogeneous Multi-Processing (HMP) Architecture. The HMP architecture is typically represented as multiple operating systems each independently controlling one or more processor cores (clusters). The system architect statically allocates the processor core assignments, creating independent AMP and/or SMP configurations.

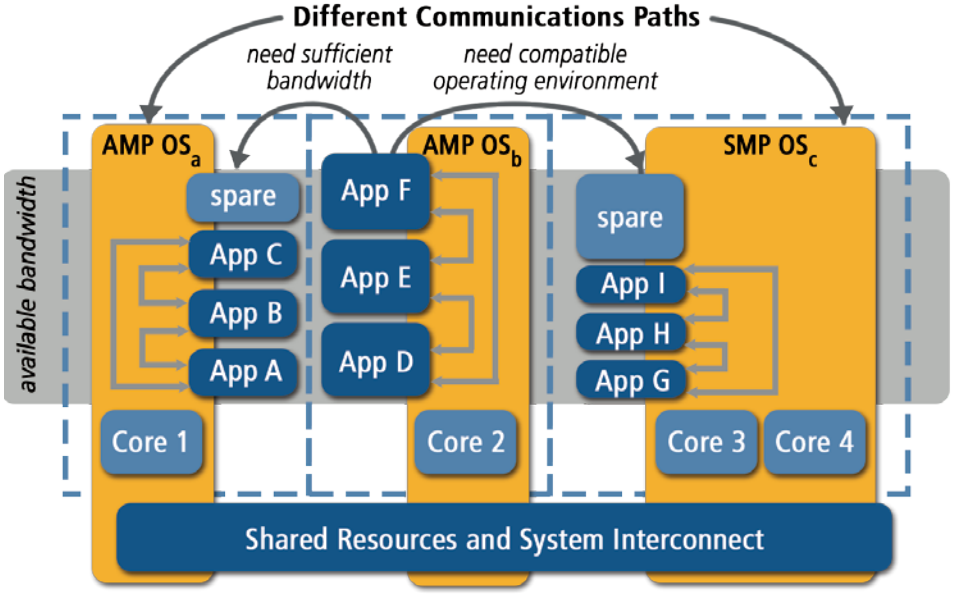
Bounded Multi-Processing (BMP) Architecture. The BMP architecture is typically represented as a single operating system controlling all of the multi-processor resources. The system architect binds some of the applications to run only on a specific core (or cores), but permits other applications to run on any available core.

Unified Multi-Processing (UMP) Architecture. The UMP architecture is typically represented as a single operating system controlling all of the multi-processor resources. The system architect divides the cores into subsystems, creating environments for AMP and/or SMP based applications to execute. Use of a common OS across all cores typically permits direct OS support for data communications between subsystems.

Load balancing

As part of an initial system deployment, system architects will utilize the selected multicore architecture to satisfactorily schedule all of the required applications. After initial deployment, the software system will over time likely need to change, whether to add required capabilities to existing applications or to add entirely new applications as part of staying competitive. These changes will require additional processor bandwidth. A long term system integration and maintenance challenge is to ensure such growth is achievable in the system's multicore architecture. This may require reallocation of applications or cores in order to ensure the processor load capabilities are not exceeded.

Load balancing can be especially challenging in HMP architectures, where cores have individually been allocated into clusters controlled by separate (and possibly different) operating systems.



An HMP example showing three clusters of cores and load balancing concerns associated with moving an application (App F) that has exceeded the available bandwidth to one of the other clusters.

Concerns with HMP include:

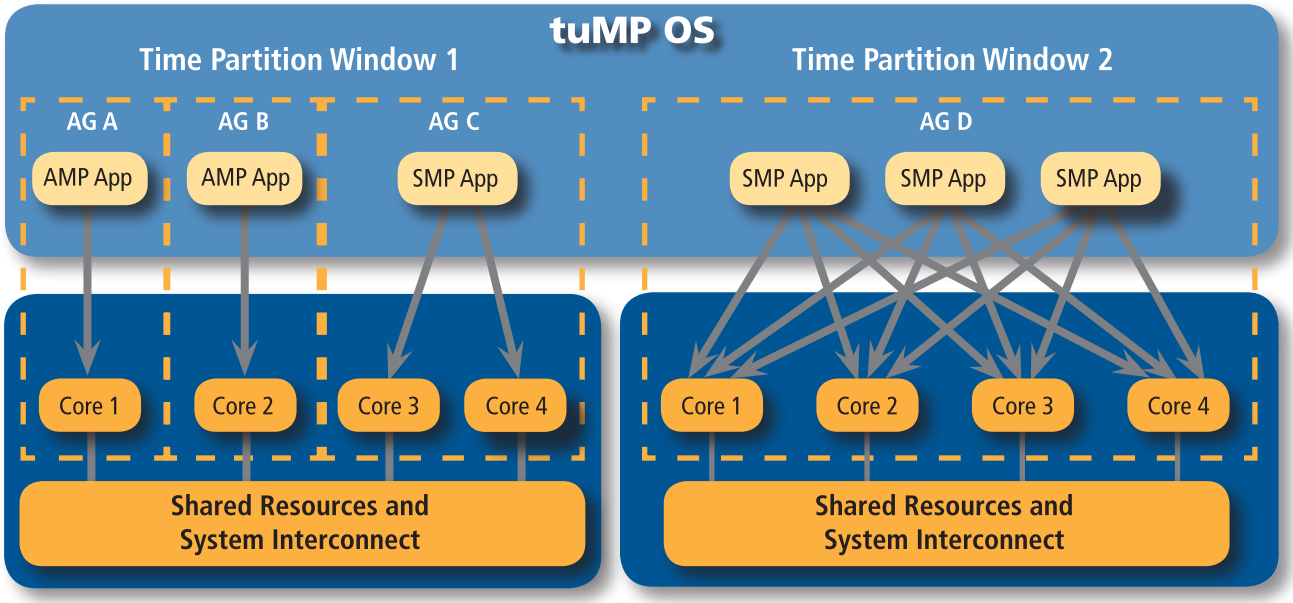
- The applications running in a cluster may have relied upon the controlling OS for communications between applications. If an application is moved to a different cluster, the communications path will be different (or non-existent).
- The OS running on the other clusters may be entirely different, either preventing movement of the application or significant updates to the applications to be compatible with a different operational environment.
- The cluster the application is moved to must have sufficient bandwidth for the entire application. In HMP, each cluster is an independent operating environment, limiting an application to utilize spare bandwidth associated with an individual cluster only.

In UMP architectures, a single operating system is responsible for all of the subsystems to which the cores have been assigned. Moving an application from one subsystem to another in UMP architectures has fewer concerns since each subsystem is controlled by one operating system that can provide a common operating environment and communication path. Since the assignment of cores to subsystems is fixed, UMP shares the HMP limitation that a subsystem must have sufficient bandwidth to support an entire application. Resolution of this limitation requires a multicore architecture that supports allocation of core assignments to change over time, permitting spare capacity of multiple clusters or subsystems to be utilized for application growth.

INTEGRITY-178 RTOS multicore support: Time-variant Unified Multi-Processing (tuMP)

Green Hills Software's resolution to the multicore design challenges is called Time-variant Unified Multi-Processing (tuMP, pronounced "2MP"), a multicore scheduling solution for the INTEGRITY-178 RTOS. The tuMP capabilities enable multiple independent safety and/or security-critical applications to execute on a multicore operating environment in a predictable, bounded, and application independent manner. The tuMP partition enforcing scheduling method results in a unified OS that provides practical time variant scheduling of both AMP and SMP applications simultaneously.

With tuMP, the system architect creates associations of cores and applications called Affinity Groups (AG) that correspond to some intended system function (or functions). Affinity Groups define how cores will be utilized by one or more applications, with the system architect defining how the Affinity Groups will be scheduled over time. Affinity Groups may be scheduled independent of other Affinity Groups, permitting time-lines that closely correspond to application requirements, yet also permitting new sets of Affinity Groups to be developed that can make use of any of the time windows where cores are not being utilized. Any new application (or extension of an existing application) can make use of the unallocated execution time across the entire multicore processor.



A tuMP example showing two time-partition windows each with different assignments of Affinity Groups (AG) for the available cores and containing unique AMP and SMP applications..

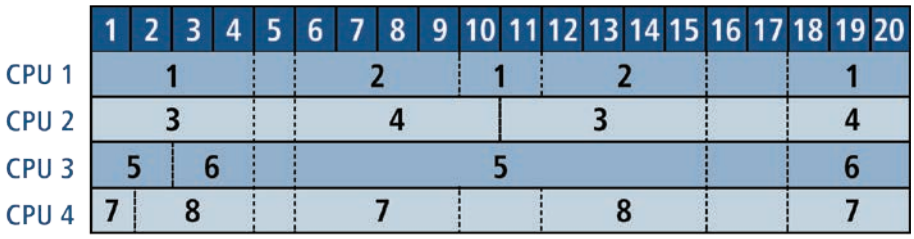
Multicore processors provide an opportunity for a significant scheduling bandwidth. This complicates the scheduling of system capabilities (e.g., background tasks, restarting applications, file systems, network servers) in that implicit scheduling may result in under-utilization of some capabilities, over-utilization of other capabilities, and unintended blocking of applications. The INTEGRITY-178 tuMP scheduler resolves this by providing the user direct control over the scheduling of all of the system capabilities. This level of flexibility supported by tuMP greatly simplifies the application load balancing concerns.

Features of the Green Hills Software INTEGRITY-178 tuMP multicore scheduling capabilities include:

- Single INTEGRITY-178 load image running on all cores concurrently
- Assignment of Address Spaces (AS) to cores (Address Space Core Affinity) using Affinity Groups
- Simultaneous support of AMP, SMP, UMP, HMP, and BMP architectures with the optional flexibility to change core assignments over time
- Optional assignment of Tasks to cores (Task Core Affinity)
- Assignment of Affinity Groups to partition time windows
- Definition of multiple partition sub-schedules per partition schedule
- Dynamic assignment of Tasks to cores at run-time
- Priority based preemptive scheduling within an Affinity Group while adhering to Task Core Affinity
- Restarting of multiple Address Spaces concurrently
- Explicit scheduling of background Tasks
- Definition and scheduling of multiple major and minor frame release semaphores
- Additional multicore specific Application Programming Interfaces (API)
- Guest OS Virtualization
- Extensive multicore debugging support
- DO-178B qualified configuration analysis and confirmation tool (G-BootTable)

The INTEGRITY-178 single-core capabilities and certification artifacts were utilized as the basis for INTEGRITY-178 tuMP. With this approach, tuMP certification considerations are an extension of the existing INTEGRITY-178 pedigree. Users of INTEGRITY-178 tuMP have this pedigree as part of the certification basis instead of the risks associated with an entirely new operating system product.

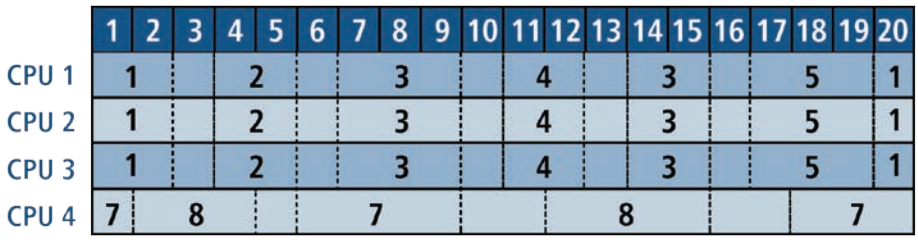
The INTEGRITY-178 tuMP scheduler is compatible with all of the Green Hills Software layered product extensions (e.g., C/C++/Ada programming language run-times, ARINC 653, network stacks, file systems).



Affinity Groups

1: AS1, AS2
2: AS3
3: AS4
4: AS5
5: AS6, AS7
6: AS8
7: AS9
8: AS10

AMP example: Every subsystem allocated to a core with its own schedule.



Affinity Groups

1: AS1
2: AS2, AS3
3: AS4, AS5, AS6
4: AS7
5: AS8
7: AS9
8: AS10

Mixed SMP/AMP example: Different Affinity Groups used to combine Address Spaces (AS), three cores in each SMP Affinity Group and one core running an AMP schedule.

Safety certification data

Green Hills Software's in-house safety and security experts develop, verify, support, and maintain the DO-178B Level A compliant software processes and life cycle data for all INTEGRITY-178 products. Through this dedicated team of experts, Green Hills Software supports customers throughout their safety critical certification efforts and deliver the required compliance substantiation data. Software life-cycle data managed as part an INTEGRITY-178 DO-178B Level A certification effort includes:

- Customer-specific Plan for Software Aspects of Certification (PSAC)
- Software Plans (Development, Verification, CM, SQA)
- Software Standards (Requirements, Design, Code)
- Software Requirements Documents
- Software Design Documents
- Source Code
- Executable Object Code
- Traceability Matrices
- Software Verification Test Cases and Procedures
- Software Verification Results
- Partition integrity, timing, memory, and stack analysis
- Problem Reports
- Software Configuration Management Records
- Software Quality Assurance Records
- Tool Accomplishment Summary
- Customer-specific Software Life Cycle Environment Configuration Index
- Customer-specific Software Configuration Index
- Customer-specific Software Accomplishment Summary (SAS)
- Integration guidance documentation

The above certification package includes Green Hills Software services for all the DO-178B Level A compliance activities associated with verifying the INTEGRITY-178 operating system on the processor architecture specified by a customer's requirements. All audits, reviews, analysis and testing of the INTEGRITY-178 operating system is performed by Green Hills Software using the customer's target processor system.

Security certification data

Green Hills also develops and maintains SKPP compliant processes and life-cycle data for INTEGRITY-178 security customers. By also completing all of the safety related processes and generating the corresponding safety life-cycle data, all security certifications support both safety (DO-178B Level A) and security (SKPP) usage in a single product.

Green Hills utilizes secure delivery procedures to deliver the substantiation evidence to security customers and to provide means for secure delivery authentication.

In addition to all of the safety-related life-cycle data, below is a list of SKPP related life-cycle data generated as part of the initial certification and/or a customer-specific security effort:

- Security-specific Software Development Plan
- Development Security Plan
- Security-specific Configuration Control Procedures
- Assurance Maintenance Plan
- Assurance Maintenance Requirements
- Installation, Generation, & Startup Guidance
- User and Administrator Guidance Document
- Security Target and Security Policy
- Formal model and proof
- Covert Channel Analysis
- Architecture Design Document
- Target Platform-specific Definition Document
- Target Platform-specific Vulnerability Analysis
- Customer-specific Security Impact Analysis
- Security-specific reviews

The Rockwell Collins' GPC-3000 Mission Computer on board the Shadow RQ-7B Unmanned Air System hosts the INTEGRITY-178 Time-Variant Unified Multi-Processing (tuMP) capabilities (image courtesy US Navy).

Corporate Headquarters

30 West Sola Street ▲ Santa Barbara, CA 93101
ph: 805.965.6044 ▲ fax: 805.965.6343 ▲ email: info@ghs.com ▲ www.ghs.com

European Headquarters

Fleming Business Centre ▲ Leigh Road ▲ Eastleigh ▲ Hampshire S050 9PD ▲ United Kingdom
ph: +44 (0)2380.649660 ▲ fax: +44 (0)2380.649661 ▲ email: info-emea@ghs.com

Safety & Security Critical Products

34125 US Hwy 19 North ▲ Suite 100 ▲ Palm Harbor, FL 34684
ph: 727.781.4909 ▲ fax: 727.781.3915 ▲ email: info-sscp@ghs.com