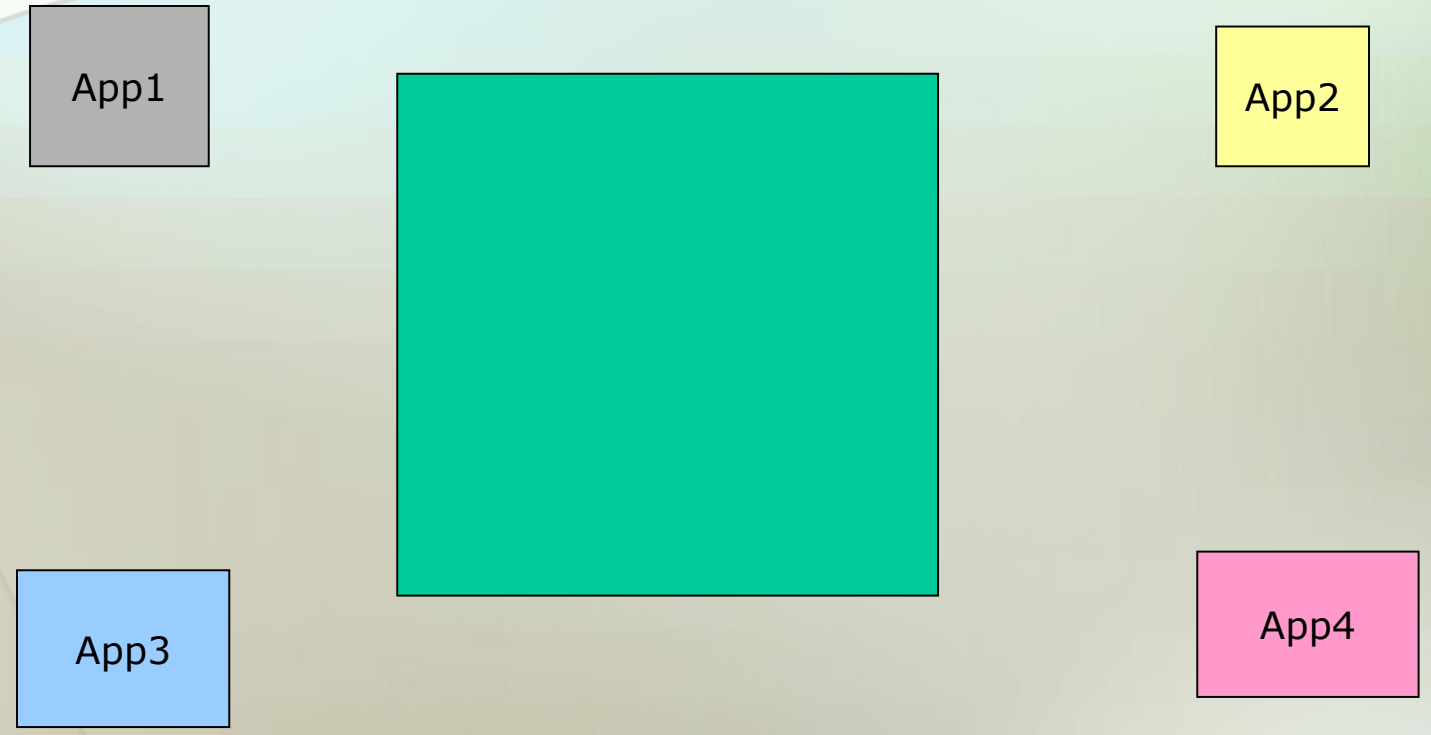
Exploit the Advantages and Resolve the Challenges of Multicore Technology with Linux

Multicore Issues to Solve

- Communications, synchronization, resource management between/among cores
- Debugging: connectivity, synchronization
- Distributed power management
- Concurrent programming
- OS virtualization
- Modeling and simulation
- Load balancing
- Algorithm partitioning
- Performance analysis

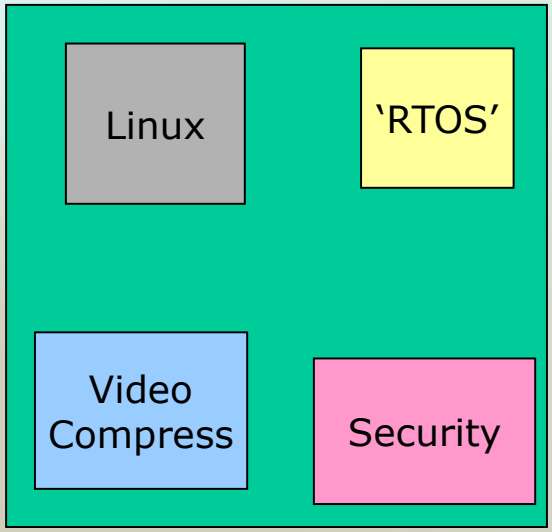
Multicore for Multiple Reasons

- Increase compute density
  - Centralization of distributed processing
  - All cores can run entirely different things; for example, four machines running in one package



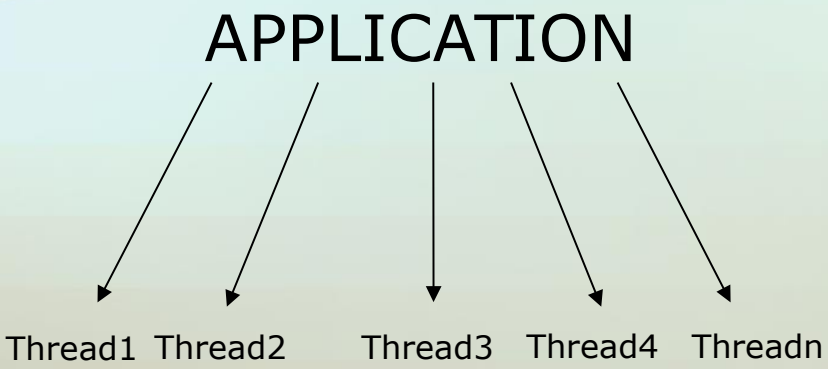
Multicore for Multiple Reasons

- Asynchronous multiprocessing (AMP)
  - Minimizes overhead and synchronization issues
  - Core 1 runs legacy OS, Core 2 runs RTOS, others do a variety of processing tasks (i.e. where applications can be optimized)



Multicore for Multiple Reasons

- Parallel pipelining
  - Taking advantage of proximity
  - The performance opportunity….

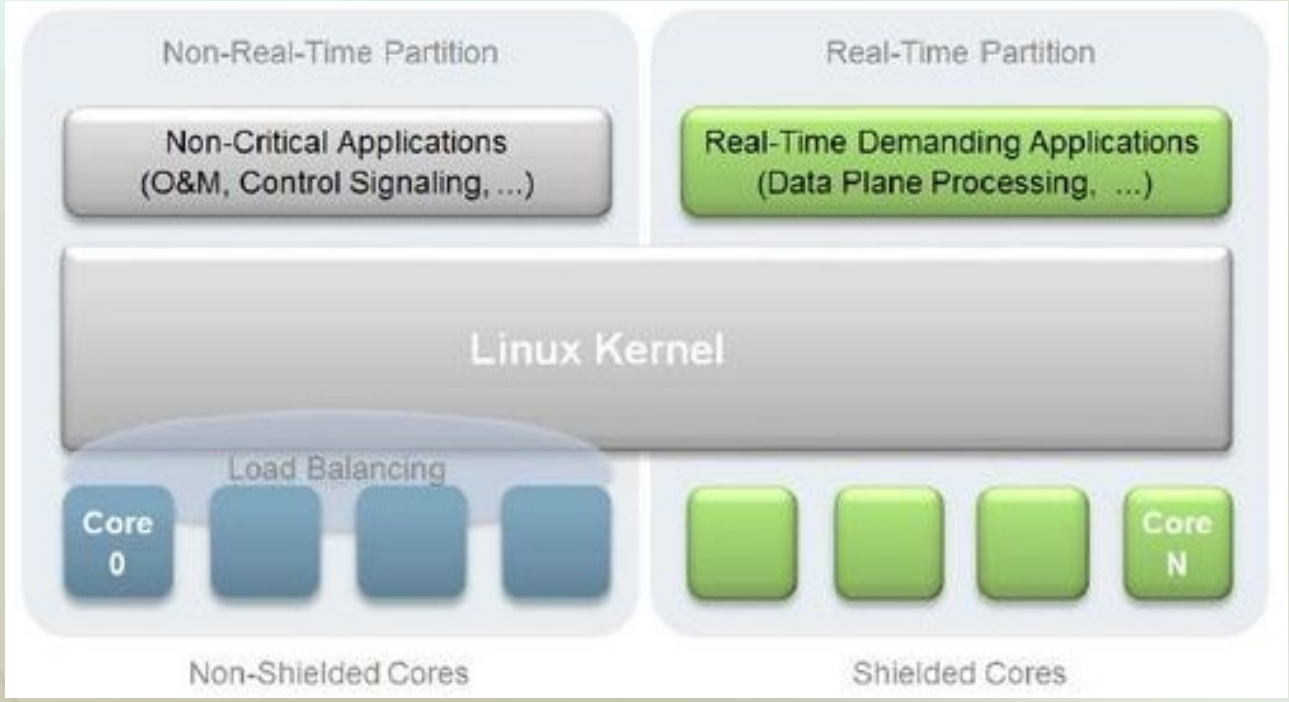


SMP Affinity

- Processor Affinity
  - use the command "taskset" which allows tasks to be assigned to a specific CPU according to their PID.
- IRQ Affinity
  - complements Processor affinity by providing the ability to directly route an interrupt to a specific processor.

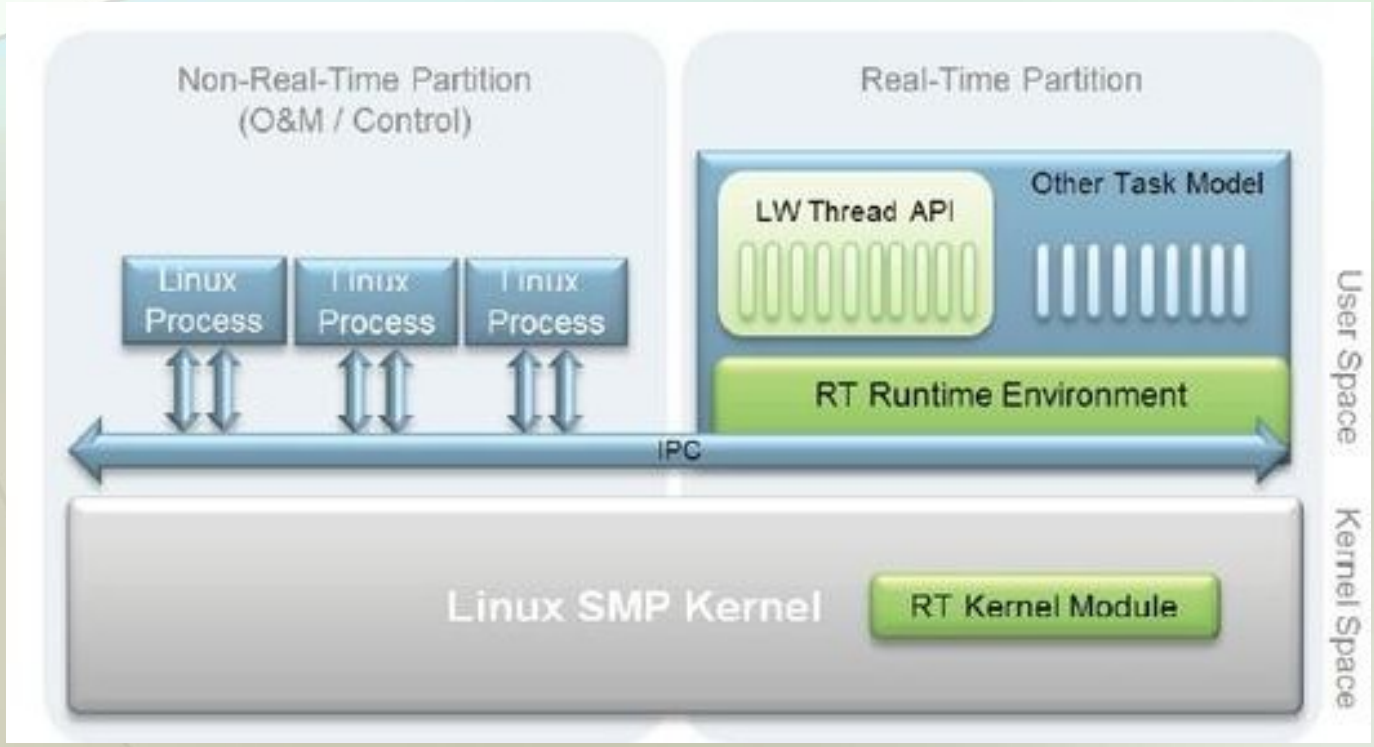
Enea's LWRT

offering user-space implementations of scheduling, message passing, and resource management, and unlike functionality provided by the kernel, the user-space implementation can be used without the overhead and indeterminism caused by context switches to supervisor mode.

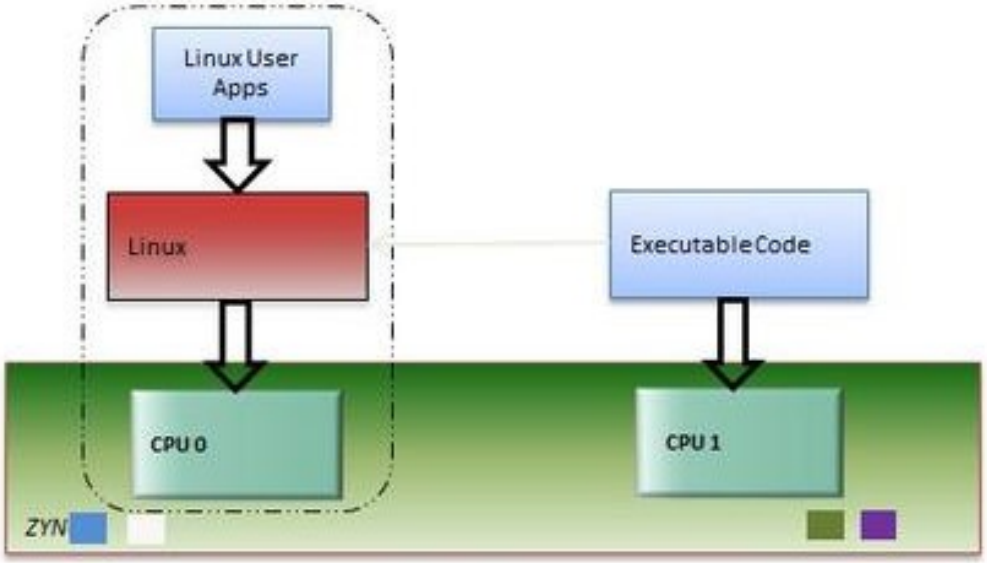


Enea's LWRT

Though LWRT uses a user-space implementation of Enea's OSE RTOS, LWRT exposes a Linux programming API to the developer for support of real-time tasks. As a result, applications can be migrated between Linux (standard) and Real-Time, easily.

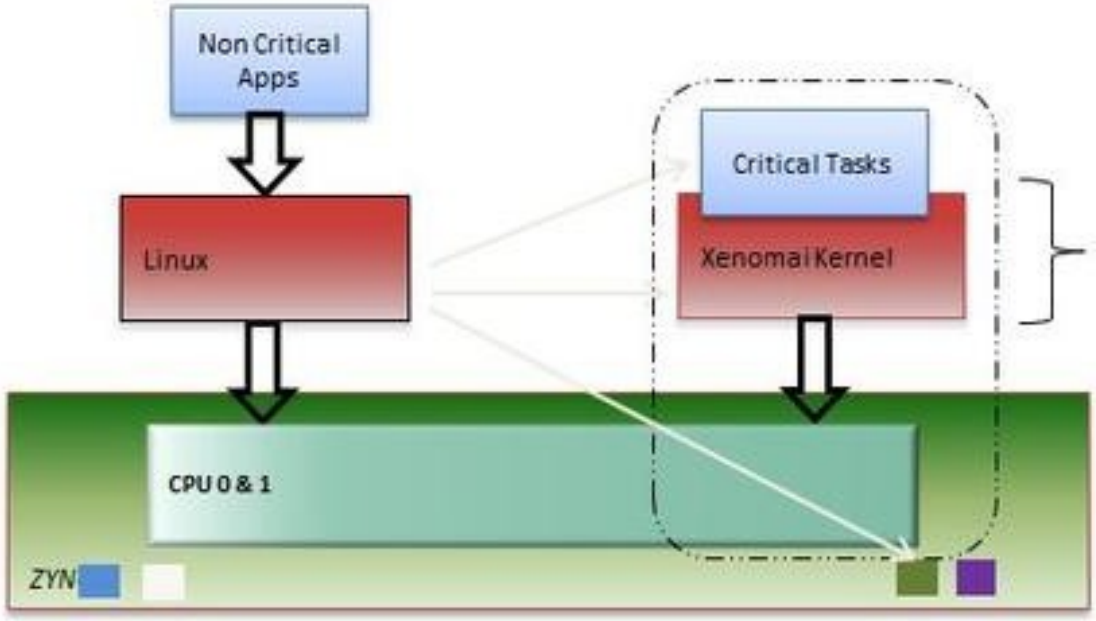


Linux / Bare-metal AMP



- Linux runs on CPU 0
  - Linux then starts CPU 1 executable that was loaded into memory by FSBL
- Bare-metal application runs in dedicated memory space on CPU 1
  - MMU 0 is used by Linux as normal; MMU 1 is used to define the memory context of bare-metal application, but no special coding is required
  - Bare-metal application does not run within the context of Linux memory
  - Linux doesn't know about the memory used by CPU 1
- MMU can be used to contain application
  - MMU defines what addresses (memory and AXI devices) application may normally access
  - CPU 1 is not restricted from accessing Linux memory space or shared devices (ICD or SCU), but Linux can detect if/when that occurs and take appropriate action.
- Implementation Approach
  - Communications between Linux and bare-metal use OCM
  - OCM caching is disabled for improved determinism
- Use Case now supported by Xilinx WTS
  - Application Note and reference files available February, 2013

Xenomai

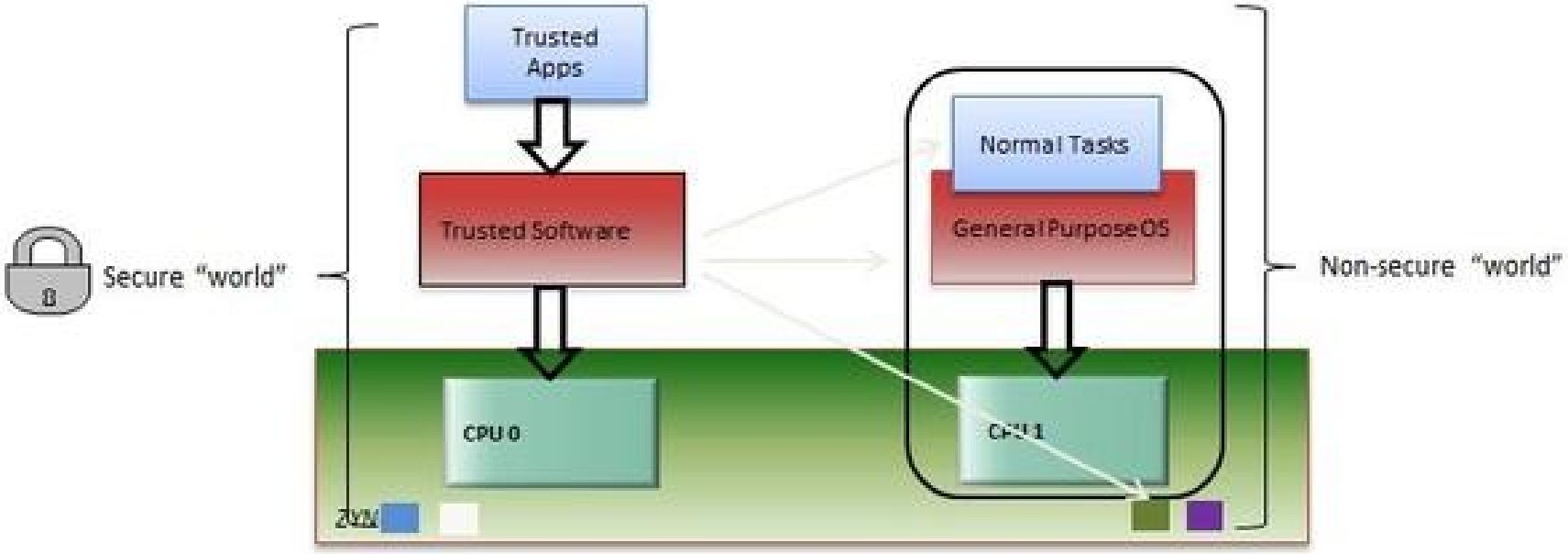


RTOS "Skin"

- Xenomai-native API
- POSIX
- uITRON
- VRTX
- VxWorks
- pSoS+

- A potential solution for commercial RTOS API support (limited)
- Real-time threads can be assigned to CPU
- RTOS is prevented from corrupting Linux
- Linux kernel is not prevented from corrupting RTOS
  - However, violations should only occur from explicit calls or through specific types of crash
  - System design and comprehensive testing may mitigate customer real-world concerns

SierraTEE



- TZ supports a "secure world" and "non secure world"
  - "Secure" and "Non Secure" run "Trusted" and "Non-Trusted" software respectively
  - Non-secure world software can access **only** non-secure resources
  - Secure world software can access ***all*** system resources
  - The Secure World is secured only *from* the TrustZone "Unsecure World" (i.e. local)
  - The Secure World is subject to external attacks
    - Trusted software often comprises of limited software functionality: DRM, encryption, password entry, etc.
  - Secure World API: Global Platform (http://www.globalplatform.org)
    - DRM, crypto, NetFlix, etc.
  - If "Secure World" software can not be safety certified then the whole system may not be certifiable
- Zynq TrustZone Protection can be applied to
  - Memory locations, Cache, AXI Devices, Interrupts