

System Protection for Multi-core Systems

Product Overview

eT-Kernel Multi-Core Edition Memory Partitioning is a real time operating system that provides the highest level of reliability and security for multi-core systems. The Memory Partitioning Option is best suited for automotive applications, aerospace instruments, high-end consumer electronics, and office automation products with memory management units (MMUs) to attain high reliability and high quality within the system.

Key Features

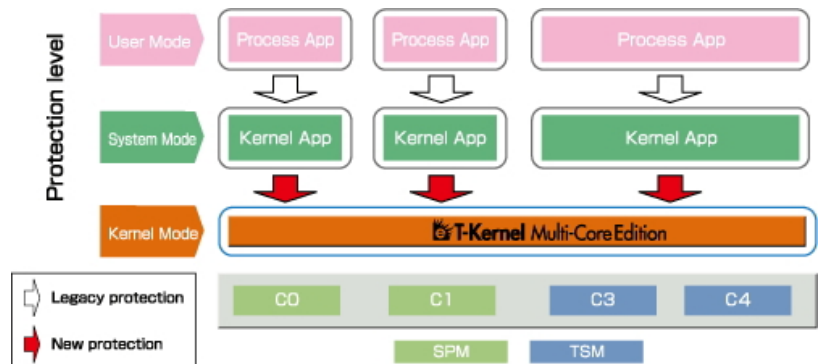
- ❑ Securely integrates sub-systems with different reliabilities
- ❑ Promotes reuse of software assets
- ❑ Provides seamless communication between partitions
- ❑ Sharing of device drivers and middleware
- ❑ Easy software migration from single core to multi-core environments
- ❑ Integrated debugging of all partitions by using the "eBinder" development environment
- ❑ Better fit for embedded systems than the Hypervisor

Architecture

Though using a process-model OS such as eT-Kernel/Extended and eT-Kernel/POSIX can prevent kernel and other process memory corruptions caused by processes, there has been no effective means to protect the kernel and process memory from kernel applications such as device drivers, interrupt handlers, and middleware that operates in the CPU kernel-mode (privileged mode). eT-Kernel Multi-Core Edition Memory Partitioning solves this problem with two technologies, "Kernel Protection" and "Core Partitioning".

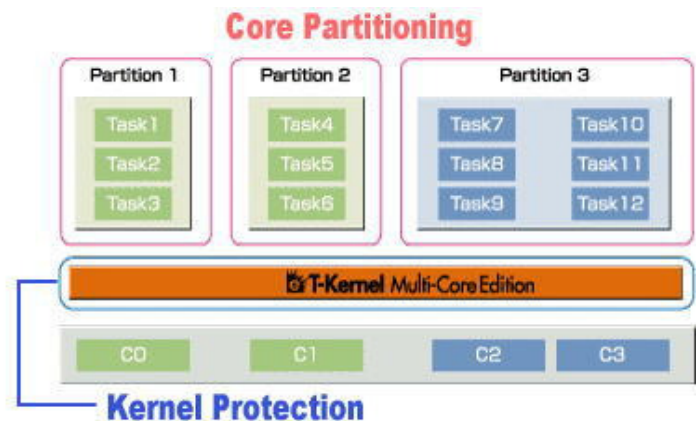
Kernel Protection

Kernel protection is a technology that protects the kernel itself from kernel applications. In addition to the kernel mode (privileged mode) and user mode (non-privileged mode) that are offered by today's CPUs, the kernel protection adds a new "system mode". Although a program that operates in system mode can access process memory, it doesn't have the authority to access the kernel memory. The kernel application is operated in this system mode.



If the kernel memory is accessed, the exception manager catches the error and executes the user-defined error processing. At this time, integrated error processing is possible by using the various types of exception information offered by the exception manager function. Because there is no need to change kernel applications, existing device drivers and middleware can be reused as-is.

As a result, because the software (except the kernel) that can destroy kernel memory doesn't exist, the kernel can achieve a completely protected state.

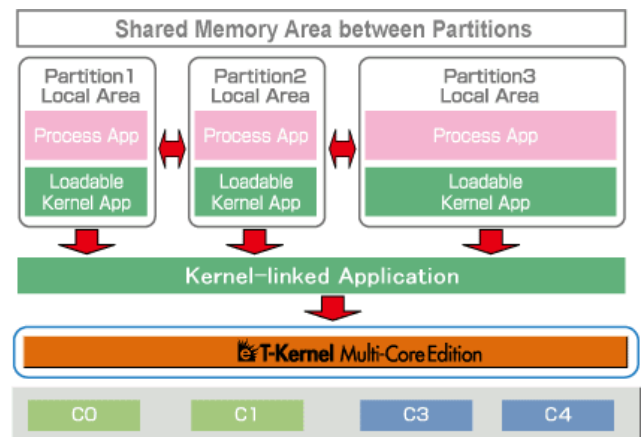


Core Partitioning

Core Partitioning is a technology to separate software into "Partitions", and to prevent memory access between partitions. As a result, memory corruption of other partitions can be prevented by the kernel applications executing in system mode. In the eT-Kernel Multi-Core Edition's Blended Scheduling Technology, which offers four

scheduling modes, a task set that is scheduled in the same scheduling mode and operating on the same CPU core (or group of CPU cores) is defined as a "scheduling unit". In Core Partitioning, the area specified by the software for the partition and scheduling unit becomes the same, since a partition is created for each scheduling unit.

Similarly, tasks and processes can only be created in the same partition, since neither can access another partition's memory. You can only load programs in the partition where the load-processing task belongs. In addition, kernel applications from other partitions cannot be started here. A method of sharing kernel applications between partitions is provided separately.



Provides seamless communication between partitions

- ❑ **APIs for communication API between tasks/processes:** Inter-task synchronous communication and exclusion APIs similar to the ones intended for single cores can be used within tasks/processes. For instance, POSIX named services (such as named pipes and named semaphores) can be used, or you can mount several additional physical file system plug-ins into a logical file system (LFS), resulting in more transparent file access. You can use a POSIX API's memory-based objects (such as mutex, condvar) and message boxes using shared memory between partitions. From this, it is easy to reuse single-core software, or even multi-core software. Moreover, another advantage is the smooth migration of software components between partitions.
- ❑ **API for sharing memory between partitions:** This is the shared memory that can be used between several partitions. A memory acquisition system call has been added in addition to the malloc library. You can make the best use of the cache-coherency feature of MPCore, etc.

Integrated debugging of all partitions by using the Binder development environment

A single instance of the OS and a single debugging environment with eBinder makes multi-core based development very easy.

- Partitioning allows smoother collaboration between partitions
- Easier software relocation/reuse between partitions
- Reuse of Kernel applications such as device drivers and middleware
- System configuration by one eT-Kernel Multi-Core Edition OS

Allied Product

- **eT-Kernel Multi-Core Edition:** T-Kernel for multi-core processors
- **eBinder:** Integrated development environment for eT-Kernel RTOSes

eSOL Co., Ltd.
Japan Headquarters
Embedded Products Division

Harmony Tower, 1-32-2 Honcho
Nakano-ku, Tokyo 164-9721, Japan
Tel: +81 3-5302-1360 Fax: +81 3-5302-1360
ep-info@esol.co.jp