

Christopher Smith discusses reliability issues effecting safety-critical systems

FAIL-SAFE FACTORS

Military and avionics systems continue to define the upper limit of the term software reliability. The consequences of the failure of a safety-critical system onboard a civil or military aircraft are immeasurably more serious than a glitch in a consumer appliance or internet service delivery. Embedded automotive systems now perform critical vehicle control functions and, therefore, have failure modes that are potentially fatal to the user. The term 'total reliability' sets software for high-performance safety-critical situations apart from the high reliability products that target 'five-nines' availability for important commercial applications.

EFFECTIVE PARTITIONING AND RESOURCE AVAILABILITY

When creating a total reliability system the designer must ensure individual components are unable to affect one another without permission. For instance, a bug in the intercom between an airliner's flight deck and passenger compartment obviously cannot be allowed to cause a failure in the fly-by-wire system. But partitioning in this way alone is not sufficient to ensure true total reliability. Resource availability must also be guaranteed so priority tasks can access the CPU and memory resources they require to meet hard real-time deadlines and to successfully perform their prescribed function. And security is an ever-present issue; even if a rogue agent is able to insert arbitrary code into a component that may be exposed to a network, this cannot be allowed to crash the system or permit theft of data.

DEFINING FEATURES OF TOTAL RELIABILITY RTOS

To meet these demands, the designer is dependent on the underlying features of the operating system. While COTS operating systems frequently display memory protection features and real-time scheduling in the case of RTOS, many are not structured to guarantee resource availability in both time and space domains. The additional functionality needed to provide these guarantees is critical to achieving total reliability.

As far as resource availability in the time domain is concerned, RTOS scheduling is commonly effected by a simple priority-based scheme where the CPU is always availed to the highest priority task, tasks at the same priority level share the CPU equally, and tasks can raise or lower their priority at will as well as spawn new tasks of arbitrary priority. But allowing a task to change its priority and to create others allows it to increase its share of the CPU, potentially starving other tasks and preventing them from performing their duties. Therefore, in order to guarantee availability of the CPU to all high priority tasks, the RTOS

must also be structured to recognise partitions, which are groupings of related tasks. Each partition is entitled to a certain percentage of the CPU, forcing it to distribute this entitlement over its incumbent tasks and never allowing it to impact the CPU bandwidth reserved for tasks in another partition.

In the space domain, any designer who has sought to manage shared resources between address spaces will know from experience that one address space can potentially consume this resource completely and deny access to the other address spaces. To eliminate this, each address space that correlates to a partition must have its own pool of resources, and the system designer must assign each address space a pool of physical spaces at boot time.

Many RTOS also lock or disable interrupts in the kernel to synchronise access to critical data structures. But these locks can cause hidden dependencies between tasks that can cause unpredictable results in the field. Disabling interrupts can also disrupt the system's real-time guarantees. To avoid this kernel calls must clean up their state and transfer control within a deterministic time frame.

It is very important that designers of total reliability systems ensure that, as well as the basic features of a high reliability COTS system, their chosen RTOS also includes comprehensive features to guarantee resource availability in both domains, a consistent interrupt structure and real-time guarantees and security.

APPLICABLE STANDARDS

The COTS marketplace is also standards-driven. This satisfies demands for flexibility in porting applications, and even changing the RTOS, as well as providing assurances certain basic functional expectations will be met. While some global standards, such as the latest POSIX standard, the 2003 edition of POSIX.1, define application programming interfaces that facilitate the designer's task, others, such as the commercial aviation software specification DO-178B, are more far-reaching and apply to the design, development, and assurance methodologies. An RTOS already certified compliant with DO-178B Level A is fundamental to achieving overall total reliability compliance. 



CHRISTOPHER SMITH,
VICE PRESIDENT, MARKETING,
GREEN HILLS SOFTWARE