LEO/P4 - A µ-Kernel Based OSEK Implementation

Robert Kaiser
Sysgo Real-Time Solutions GmbH, Klein-Winternheim. Germany
rkaiser@sysgo.de
January 18, 2000

1. Abstract

This article introduces LEO/P4, an implementation of the standardized OSEK OS API as a server on top of the new µ-kernel P4. Together with the OSEK emulation environments LEO/Lynx or LEO/Linux, it provides an OSEK run-time environment that can seamlessly support OSEK applications from the early stages of development unto the final production version in a vehicle.

The P4 µ-kernel is a new, advanced re-implementation of the L4 µ-kernel as specified by Liedtke /4/. It features a processor independent API and is internally structured for better portability. Platform specific support routines are provided to P4 by an external module (PSP), so porting P4 to another platform within the same processor family can be achieved without modifications to the µ-kernel's binary code.

In contrast to traditional monolithic implementations of OSEK, the µ-kernel-based approach allows for multiple operating system APIs and instanciations to coexist. Specifically, it is possible to have multiple instances of OSEK OS running independently in a single machine, each one in it's own protected address space.

Moreover, a Linux server can also be added to run in parallel with the OSEK OS instance(s), thus turning the system into a full-featured UNIX workstation. This does not affect OSEK's real-time characteristics, so the resulting system is a very comfortable and efficient environment for developing, debugging and testing OSEK-based code under real-world conditions.

All interaction between the servers is based on the µ-kernel's inter process communication (IPC) mechanism, which features the capability of transparent re-routing of messages across a network. Therefore, servers can be migrated from the development system to the embedded controller without any modification to their code. This not only provides an easy way to implement distributed systems, it also makes the process of moving OSEK applications from the development system to their designated target ECU far less painful than usual.

We will start with brief introductions of the OSEK OS standard and of LEO, SYSGO's OSEK OS implementation and will then focus on the benefits of the µ-kernel-based implementation, as well as on the P4 µ-kernel itself.