Symmetric Multiprocessing Guide for Windows Embedded Compact 7

Writers: Glen Langer, Jina Chan

Technical Reviewer: Glen Langer

Published: November 2011

Applies To: Windows Embedded Compact 7

Abstract

Windows Embedded Compact 7 has new features to support symmetric multiprocessing (SMP). This document briefly describes the OEM adaptation layer (OAL) interface for SMP to help you bring up a multiprocessor hardware platform using SMP.

Contents

Introduction

You can introduce support for symmetric multiprocessing (SMP) in your OS design by implementing new functions and variables in the OEM adaptation layer (OAL). SMP involves a multiprocessor architecture where multiple processors are connected to a single main memory and are controlled by a single OS instance.

New Functions for SMP

Windows Embedded Compact 7 introduces new functions and variables for SMP. They are contained in the OEM adaptation layer (OAL) structures OEMGLOBAL and NKGLOBAL.

OEMGLOBAL members for SMP:

- fMpEnabled
- pfnStartAllCpus
- pfnMpPerCPUInit
- pfnSendIpi
- pfnIpiHandler
- PfnIdleEx
- pfnMpCpuPowerFunc

NKGLOBAL members for SMP:

- pfnNKSendIPI
- pfnAcquireOalSpinLock
- pfnReleaseOalSpinLock

These structures and their members are documented in the product help for Windows Embedded Compact 7.

Handling Interprocessor Interrupts (IPIs)

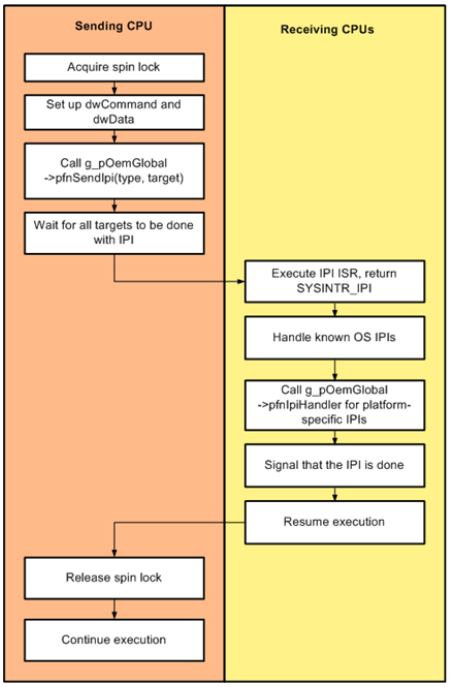
You handle interprocessor interrupts (IPIs) by implementing an **OEMIpiHandler** function and setting the **pfnIpiHandler** member of OEMGLOBAL to point to your implementation. IPI is the only type of interrupt that must be handled on every CPU. Other interrupts should only be delivered to the main CPU and handled by the main CPU.

Implementation of how IPIs are sent and handled is determined by the OAL. For example, on x86 platforms, the current implementation uses IRQ 255 as IPI, and calls `HookInterrupt(255, IpiInterruptHandle)` when there is more than one CPU.

When you handle an IPI interrupt, you only need to acknowledge (ACK) and clear the interrupt, then return SYSINTR_IPI.

IPI Sending Sequence

The following execution sequence diagram shows the order in which IPIs are sent and received.



Startup Sequence for SMP

The following execution sequence diagram shows the startup sequence for a platform with SMP.

Porting Your OAL

To introduce SMP support in an existing OAL

1. Implement a routine to start the secondary CPUs and send IPIs at the end of the OEMInit function, without calling the kernel. This routine must set the secondary CPUs spinning so they will only respond to IPIs. This makes it easier to debug the startup function and IPI handling.
2. Search for code that turns off interrupts, and replace this code with spin locks. Turning off interrupts does not guarantee exclusive access to resources on multiprocessor platforms.
3. Change any performance counters implemented with CPU local timers so that they can be called from any CPU.
4. Change any performance counters implemented with a global timer (the BSP timer) so that they are protected with spin lock.

Conclusion

Windows Embedded Compact 7 supports symmetric multiprocessing (SMP) hardware platforms. To create an OS design that supports SMP, you must implement new functions and variables in the OEM adaptation layer (OAL). This document lists the new APIs, describes IPI sending and handling, the startup sequence for a platform with SMP, and lists the steps to port your OAL to Compact 7 with SMP support.

Additional Resources

Windows Embedded website (http://go.microsoft.com/fwlink/?LinkId=183524)

© 2011 Microsoft. All rights reserved.