# Windows 7 Device Drivers Outline

## Introduction:

This course gives developers the knowledge to design, write, and debug Windows 7 device drivers.

## Prerequisites

Before taking this course, students should have the following skills:

- C Programming Language competency
- Experience with Microsoft Visual C++ (Developer Studio) environment
- Some knowledge of device driver development on other systems (HW and/or SW)
- User-level experience with Windows 7 or XP

## Goals

- Very compressed introduction on how to write drivers
- Infrastructure and management operations (drivers registration, registers, APIs, user<->driver communication)
- User space driver model – direct access to HW, management over the driver
- Memory management (including virtual memory management) – 32/64 bit env. , division of memory between user and kernel space
- Samples of file systems, how to create and operate on them on Windows
- Debugging aspects – including available tools, how to analyze logs/dumps of data, how to analyze BSOD data or backtrace - exercises are welcome
- Certification of drivers – how to do it, what process drives it
- How to include debug code into driver's code

## *Introduction to the Vista Architecture*

- Design Goals
- Windows OS layers
- Kernel mode vs. user mode
- The I/O subsystem
- Where device drivers “fit”

## *Windows Vista Driver Architecture*

- The Driver Models: Legacy, WDM, WDF, KMDF, UMDF
- Context of code running in kernel mode
- The Windows interrupt abstraction
- Deferred procedure calls
- User buffer access
- Structure of a kernel-mode driver
- The I/O processing sequence

## *First Driver Code*

- Describe the build environment for a driver
- Describe the Windows services driver code can use
- Write “core driver code” to load and unload the driver using the Windows Device Manager
- Lab: First Driver Code

## *WDF – KMDF Drivers*

- Describe the WDF Model
- The KMDF Model
- KMDF Objects
- Dispatching
- Event Handling
- Creating a KMDF Driver
- Lab: KMDF Loopback

## *The WDM Model*

- Describe the history of WDM
- The Plug & Play Architecture of Windows XP
- Hot-plug devices
- OnNow initiative
- Power management issues
- The role of the system registry
- Lab: Parallel Loopback

 

## *WDF – UMDF Drivers*

- Describe the need for User Mode Drivers
- The UMDF Model
- UMDF Objects
- Event Handling
- Creating a UMDF Driver
- Lab: Building a simple UMDF Example

## *Windows Management Instrumentation (WMI)*

- Overview of WMI and event logging
- The WMI Classes
- Becoming a WMI Provider
- The WMI IRP's
- The WBEM Object Browser
- Event logging
- Lab: Using the Event Log

## *Interrupt Driven I/O*

- Describe the sequence of events for a Programmed I/O Device operation
- Describe the use of an Interrupt Service Routine
- Describe the role of a DPC routine during an interrupt
- Examine data transfer routines
- Interrupts in Windows 7: Message Signaled Interrupts
- Lab: Writing an ISR

## *DMA Operations*

- Describe the Windows DMA abstraction
- Describe the role of the Adapter object
- Describe the purpose and structure of a Memory Descriptor List
- Explain the differences between a Slave and Master DMA device driver
- Lab: A DMA Driver Walkthrough

## Memory Management

- Virtual Address Translation
- Page Faults
- Working Set Management
- Physical Memory Management

## Debugging Drivers

- The Windows “Blue Screen of Death”
- WinDbg
- Crash dumps
- DUMPEXAM & DUMPCHK
- Interactive debugging with WinDbg
- Lab: Using WinDbg

## Timers

- What is an IO Timer?
- Uses: Device Timeouts
- Uses: Polling a device
- Custom timer DPC routines
- Time measurements
- Lab: IO Timers

## Layered & Filter Drivers

- What is an Intermediate Driver
- Layered drivers
- Filter drivers
- Coupled drivers
- Connecting to other drivers
- Lab: Layered Driver

## *Installable File System Drivers*

- File System Drivers – Simple Picture
- FSD's – The Complex Realities
- File System Filter Drivers
- File System Mini-filter Drivers
- Network Redirectors
- Security Considerations

## *Installing Device Drivers*

- Auto-install using INF files
- Manual installation
- The Driver Install Framework (DIFx)
- Windows File Protection
- Driver certification
- Digital signatures

## *Logo Testing – Introduction*

- Why Certification?
- How the testing program works
- The Windows System and Device Requirements Document
- The Test Environment
- Submitting the Test Results
- Using the Signed Driver