Windows Internals

Windows 7, 8.x, Server 2012, Server 2012 R2

Course Summary Table

Duration:	5 Day
Target Audience:	Experienced windows programmers, interested in writing better programs, by getting a deeper understanding of the internal mechanisms of the windows operating system, as exposed by the Windows API and the Kernel API.
Objectives:	Understand the underlying mechanism and advanced services of the windows OS and use that knowledge to write better and more efficient programs on windows 7, 8, Server 2008/R2 & 2012
Pre Requisites:	Basic knowledge of OS concepts and architecture. Practical experience developing windows application C/C++ knowledge is an advantage

Abstract

The Windows OS exposes many advanced services to system programmers through the Windows API, and to device driver writers through the Kernel API. The .NET framework wraps these services and runs on top of the Windows API and the Kernel.

Good knowledge of what's going on under the hood of the OS, which services are available and how to best utilize them helps in building better and more efficient software for Windows. This course discusses the internal workings of Windows and its exposed services, so they can be leveraged better by you!

Syllabus

- Module 1: System Architecture
  - Windows NT History
  - Basic Concepts
  - Windows Editions – Client, Server, Server Core
  - Tools: Windows, SysInternals, Debugging Tools
  - Processes, Threads, Virtual Memory
  - User mode vs. Kernel mode
  - Requirements and Design Goals
  - Architecture Overview
  - Key Components
  - APIs: Win32, Native, .NET, COM, WinRT
  - User/kernel transitions
  - Introduction to WinDbg
  - Lab: Task manager, Process Explorer, WinDbg

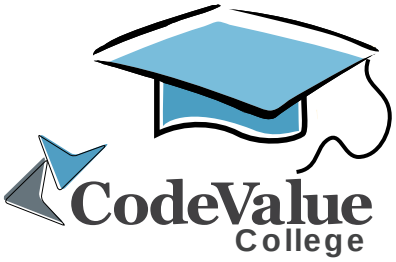
- Module 2: Kernel Mechanisms
  - Trap Dispatching
  - Interrupts & Exceptions
  - System Crash
  - Object Management
  - Objects and Handles
  - Sharing Objects
  - Synchronization
  - Synchronization Primitives
  - Signaled vs. Non Signaled
  - Windows Global Flags
  - Kernel Event Tracing
  - Wow64
  - Lab: Viewing Handles, Interrupts; creating maximum handles

- Module 3: Management Mechanisms
  - The Registry
  - Services
  - Starting and controlling services
  - Windows Management Instrumentation
  - Kernel Transaction Manager
  - Lab: Viewing and configuring services; Process Monitor

- Module 4: Processes & Threads
  - Multitasking and Multiprocessing
  - Process Internals & Data Structures
  - Prcessor Groups, 256 cores & NUMA overview
  - Creating and terminating processes
  - DLL explicit and implicit linking
  - Creating Threads
  - Thread Priorities
  - Thread Scheduling
  - Thread Stacks
  - Thread States
  - Thread Synchronization
  - Jobs
  - Lab: creating threads; thread synchronization; viewing process & thread information

- Module 5: Memory Management
  - Overview
  - Small and large pages
  - VMM Services
  - Memory states
  - Address Space Layout
  - Address Translation Mechanisms
  - APIs in User mode and Kernel mode
  - Page Faults
  - Workings Sets
  - Memory Mapped Files
  - Page Frame Database
  - Optimization Techniques
  - Lab: committing & reserving memory; using shared memory; viewing memory related information

- Module 6: Security
  - Security System Components
  - Protecting Objects
  - User access control
  - Access Rights and Privileges
  - Logon
  - Session 0 Service Isolation
  - UIPI
  - Lab: viewing security info



- Module 7: I/O System
  - I/O System overview
  - I/O Function
  - Device Drivers
  - The Windows Driver Model (WDM)
  - The Kernel Mode Driver Framework (KMDF)
  - I/O Processing and Data Flow
  - Plug & Play
  - Power Management
  - Writing a Software Driver
  - Lab: viewing driver and device information; software driver (if time permits)

- Module 8: Networking
  - Networking Architecture
  - Networking APIs
  - Redirectors
  - NDIS
  - Binding
  - Network Services

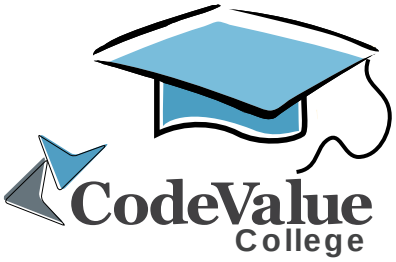
- Module 9: Introduction to Windows 8.x
  - The Windows 8 Start Screen
  - Desktop vs. Metro apps
  - Application lifecycle
  - The Windows Runtime
  - COM for WinRT
  - C++/CX
  - Application binary interface
  - Asynchrony in WinRT
  - Capabilities
  - Lab: writing simple metro apps

Course Compatibility Questionnaire

Please answer the following questions as accurately as possible:

Name: _____ Email: _____

Company: _____ Phone: _____



Language / Technology / Platform	Years of Experience						Level of Familiarity				
C	0-1	1-2	2-3	3-4	4-5	5+	1	2	3	4	5
C++	0-1	1-2	2-3	3-4	4-5	5+	1	2	3	4	5
C#	0-1	1-2	2-3	3-4	4-5	5+	1	2	3	4	5
Visual Studio Debugger	0-1	1-2	2-3	3-4	4-5	5+	1	2	3	4	5
WinDbg	0-1	1-2	2-3	3-4	4-5	5+	1	2	3	4	5
SysInternals tools	0-1	1-2	2-3	3-4	4-5	5+	1	2	3	4	5
Task Manager & Resource Monitor	0-1	1-2	2-3	3-4	4-5	5+	1	2	3	4	5
Other _____	0-1	1-2	2-3	3-4	4-5	5+	1	2	3	4	5

What is the operating system that as a user you mostly use?

What are your expectations from the course?

Thanks!

http://college.codevalue.net/