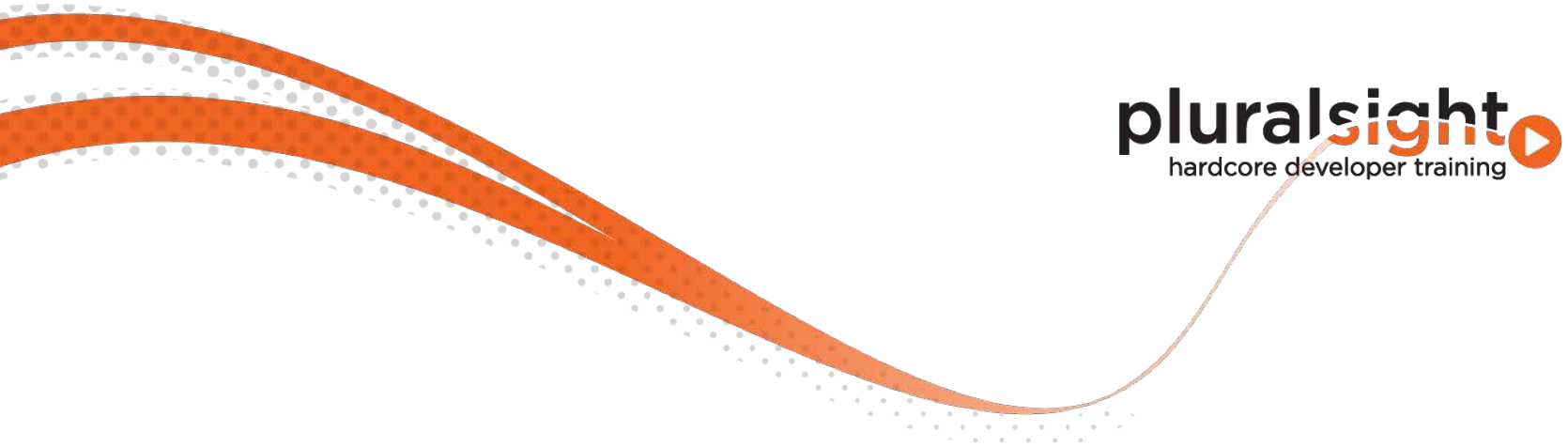
Windows Internals

Module 5: Processes & Threads

Pavel Yosifovich
CTO, CodeValue
pavely@codevalue.net
http://blogs.Microsoft.co.il/blogs/pavely



pluralsight
hardcore developer training

Contents

- Processes
- Threads
- Thread scheduling
- Thread synchronization
- Thread pools
- Jobs
- Summary

Process

- **Management and containment object**
  - Owns
    - Private virtual address space (2GB/3GB on 32 bit, 8TB on 64 bit)
    - Working set (physical memory owned by process)
    - Private handle table to kernel objects
    - Access token
  - Has a priority class (from Win32)
    - Affects all threads running in that process
  - Basic creation functions: **CreateProcess**, **CreateProcessAsUser**
  - Terminated when any of the following occurs
    - All threads in the process terminate
    - One of the threads calls **ExitProcess** (Win32)
    - Killed with **TerminateProcess** (Win32)

Process creation

- **Flow of process creation**
  - Open image file
  - Create kernel Executive Process object
  - Create initial thread
  - Create kernel Executive Thread object
  - Notify CSRSS of new process and thread
  - Complete process and thread initialization
    - Load required DLLs and Initialize
    - **DllMain** function called with **DLL_PROCESS_ATTACH** reason
  - Start execution of main entry point (**main** / **WinMain**)

Demo

Creating a Process

Demo

Process internals

Threads

- **Instance of a function executing code**
  - Owns
    - Context (registers, etc.), 2 stacks (user mode and kernel mode)
    - Optionally, message queue and Windows
    - Optional security token
  - Scheduling state
    - Priority (0-31)
    - State (Ready, Wait, Running)
    - Current access mode (user or kernel)
  - Basic creation function: **CreateThread** (Win32)
  - Destroyed when
    - Thread function returns (Win32)
    - The thread calls **ExitThread** (Win32)
    - Terminated with **TerminateThread** (Win32)

Demo

Creating threads