

Windows System Programming

Course Summary Table

Duration:	5 Days
Target Audience:	Windows developers and researchers
Objectives:	<ul style="list-style-type: none">• Understand the fundamentals of building Windows applications• Work effectively with the Windows system-level API• Leverage the capabilities of the OS, including processes, threads, memory, I/O, and much more
Pre Requisites:	<ul style="list-style-type: none">• Real-world experience programming in C• C++ experience is beneficial, but not mandatory• Basic understanding of Windows OS concepts such as processes, threads, virtual memory and DLLs

Instructor: **Pavel Yosifovich**

Abstract

The Windows system-level APIs provides a rich infrastructure for building Windows applications, whether client, server, and anything in between. This course guides the learner through the intricacies of the Windows API, while getting a deeper understanding of Windows mechanisms.

The course deals with the most important parts of the Windows OS, such as processes, threads, memory management, I/O, services, security and more. Lab exercises help put the theoretical material into practical use.

Syllabus

- Module 1: Foundations
 - Windows architecture overview
 - Windows APIs
 - Using Visual Studio
 - Common Windows types and conventions
 - Working with Strings
 - API Errors
 - 32-bit vs. 64-bit Development
 - The Windows version
 - Summary
- Module 2: Objects and Handles
 - Kernel Objects

- Handles
 - Working with Handles
 - Sharing Objects
 - Private object namespaces
 - User and GDI objects
 - Summary
- Module 3: Processes
 - Process creation
 - The main function(s)
 - Creating processes
 - Process termination
 - Enumerating processes
 - Summary
- Module 4: Jobs
 - Introduction to jobs
 - Creating jobs
 - Setting and getting limits
 - Nested jobs
 - Job notifications
- Module 5: Threads
 - Introduction to threads
 - Creating threads
 - A thread's stack
 - Terminating threads
 - Thread priorities
 - Basic thread scheduling
 - A thread's name
 - Affinity
- Module 6: Thread Synchronization
 - Synchronization basics
 - Atomic operations
 - Critical sections
 - Reader-writer locks
 - Synchronization with kernel objects
 - Mutexes, semaphores and events
- Module 7: File and Device I/O
 - The I/O system
 - The CreateFile function
 - Synchronous I/O
 - Asynchronous I/O
 - Handling async I/O completion

- I/O completion ports
 - Pipes and Mailslots

- Module 8: Memory Management
 - Process address space
 - System memory usage
 - Process memory counters
 - Reserving and committing memory
 - The heap manager
 - Memory mapped files

- Module 9: Dynamic Link Libraries
 - Why DLLs?
 - Building DLLs
 - Implicit and explicit linking
 - The DllMain function
 - Delay Load dlls
 - Injecting DLLs

- Module 10: Security
 - Windows security components
 - SIDs
 - Access tokens
 - Privileges
 - Security descriptors
 - User access control
 - Running elevated
 - Impersonation

- Module 11: Windowing
 - Windows and Threads
 - Window messages
 - Message loops
 - Enumerating windows
 - Building GUI (if time permits)