Windows Programming via WinAPIs

By Milad Kahsari Alhadi

Last Update: Monday - 2021 25 January

Part 1 Windows APIs with C/C++ – 19 Hour and 23 Min

- C++ Windows Platform Specefic Libraries
  - Static Libraries
    - File Format:
      - .lib files in Windows
      - .a files in Linux
    - Compile-time Resolution
    - Load-time Resolution
    - Creating a Static Libraries
      - Designing a Library for Division
      - Refrence to the Library
      - Copy the Library
      - Using Division Library
    - Linking Static Libraries to Projects
      - GLFW Library
      - OpenGL Library
      - SMFL Library
      - Creating a Windows Based Application
  - Dynamic Libraries
    - File Format:
      - .dll files in Windows
      - .so files in Linux
    - Resolution Time:
      - Load-Time Linking
      - Run-Time Linking
    - Linkage Methods:
      - Implicit Linking
      - Explicit Linking
    - Declaration Specification:
      - Extern as C
      - C Declaration
      - Type Defination
      - Function Pointer

- ## Windows Application Programming Interfaces
  - Windows System Programming
  - User-Mode and Kernel Mode
  - Processors Rings
    - Ring -2 – Godemode
    - Ring 0 – Kernel
    - Ring 3 – User & Application
  - Kernel Interfaces – APIs
    - WinAPIs
    - Posix Interfaces
    - Cocoa Interfaces
  - Creating a Window
    - Windows-Subsystem
    - Console-Subsystem
    - MessageBox API
    - Main and WinMain
  - Different Approaches of Programming
    - Standard Development
      - Standard Functions
      - Standard API's Wrappers
        - cout & cin
        - printf & scanf
        - puts & gets
    - Native Development
      - Linux Native System Call – Write
      - Windows Native System Call - NtWriteFile
    - System Call Process
      - Step1: Console Out
      - Step2: WriteConsole
        - Get Sandard Handle
        - Standard Output
        - String Length
      - Step3: System Call Handler
        - System Enter
        - Intterupt
- ## Characters Encoding and Characters Sets
  - Encoding Terms
    - Code Unit
    - Code Point
    - Byte Order
    - Code Pages
    - Hexadecimal Representation

- Multibyte Encoding Standards
  - ASCII
  - Shift-JIS
  - Advantages and Disadvantages
- Universal Encoding
  - Unicode 1.0
  - Unicode 2.0
  - Unicode 3.0
  - Advantages and Disadvantages
- Visual Studio Settings
  - Multibyte Character Set
  - Unicode Character Set
  - String Parsing Methods
    - Super Unicode Edit
    - wxEdit
  - UTF Comparison
- Developing an International Program
  - Change Code Page of CMD
    - SetConsoleCP
    - SetConsoleOutputCP
    - Registry
    - CHCP
  - Change Locale
    - SetLocale
    - Wide Character
    - Using UCS-3
  - Hello World in Persian
    - RTL and LTR of CMD
    - Using CMDR Environment
  - Generic Data Type
    - TCHAR Macro
    - TEXT Macro
    - ##x Placeholder

Windows Error Handling Mechanism

- How WinAPI Error Handler Works?
  - Thread Environment Block
    - !peb command
    - !teb command
    - LastErrorCode Flag

- WinError.h Library
  - Error ID
  - Error Numbers
  - Error Text Description
- GetLastError API
  - API's Return Value
  - WinError Macros
    - MessageID:
      - ERROR_SUCESS
      - ERROR_INVALID_FUNCTION
      - ERROR_FILE_NOT_FOUND
      - ERROR_ACCESS_DENIED
    - MessageText
- SetLastError API
- FormatMessage API
- Practical Project:
  - GUI based Error Handler App
  - CUI based Error Handler App
- CUI based Application
  - Subsystem: Console
  - EntryPoint: main & wmain
  - EntryPoint: mainCRTStartup & wmainCRTStartup
  - Console Host Infrastructure
    - ConHost.exe
    - Console Driver
    - IOCTL Messages
- GUI based Application
  - Subsystem: Window
  - EntryPoint: WinMain & wWinMain
  - EntryPoint: WinMainCRTStartup & wWinMainCRTStartup
  - Windows Explorer Infrastructure
    - Explorer.exe
    - cmd.exe

Windows Kernel Objects

- Windows Objects
  - GDI Objects
  - User Objects
  - Kernel Objects

- Kernel Objects
  - Creating Objects
  - Manipulting Objects
  - Handle Objects
  - Sample Kernel Objects
    - CreateThread
    - SetPriorityThread
    - ResumeThread
  - Analysis and Synthesis
    - Windows Task Manager
    - Sysinternals Process Explorer
    - Disassembly Analysis with IDA Pro
    - Decompilation Analysis with Ghidra
    - MSDN Manual Information

Process Kernel Object

- Process Instance Handle
  - Relative Addressing
    - Base Address
    - Offsets
  - ImageBase Address
  - GetModuleHandle API
  - HInstance and Hmodule
  - Executable File Resources
    - LoadIcon API
    - MAKEINTRESOURCE Macro
    - Changing an Executable Icon
- Process Environment Variables
  - Windows Registry Environment Data
    - HKEY_LOCAL_MACHINE
    - HKEY_CURRENT_USER
  - Process Environment Block
  - Environment Variable's API
    - SetEnvironmentVariable
    - GetEnvironmentVariable
    - GetEnvironmentStrings
    - FreeEnvironmentStrings
    - SetCurrentDirectory
    - GetCurrentDirectory
    - StringCbCopyN
    - strchr

- Windows Typedefs
  - PTSTR and char*
  - LPCTSTR and const char*
  - _Null_terminated_ and CONST macro
- Windows Version and Edition
  - OSVERSIONINFO Structure
  - Windows Version APIs
    - GetVersionEx – Depreceted
    - VerifyVersionInfo api
    - VER_SET_CONDITION macro
  - VersionHelper library
  - Fill Memory with Zeros
    - Zero Initialization in C++
    - ZeroMemory function in C

- Process Handle Tables
  - Handle Object
  - Process-based Handle
    - Flags
    - Security
    - Inheritance
  - Mutual Execution Object
    - Security Attributes
    - Security Descriptor
    - Object Usage Count
    - CreateMutex API
    - CloseHandle API
  - Windows Kits and SDKs
    - Windows Debuggers
      - DbgHelp.dll
      - DbgEng.dll
      - DbgModel.dll
      - Kd, ntsd, cdb, and WindDbg
    - WinDbg Mode
      - User-Mode
      - Kernel-Mode
      - Crash Dumps
    - WinDbg Commands
      - dt, dx, dd k
      - .hh and .cls
      - .ln and x
      - bp and bu
      - e and ea
      - !teb amd !peb

- Process Kernel Object
  - Process Overview
    - What is a Process?
    - What is a Thread?
  - Process Components
    - Kernel Objects
    - Address Space
  - Primary Thread
    - Thread Environment Block
      - PEB Address
      - Stack Base
      - Stack Limit
      - …
    - Process Environment Block
      - Environment Variables
      - BeingDebugged
      - LoadedModules
      - …
    - Execution Context
  - Thread Scheduling
    - Quantums
    - Round-Robin
    - Thread Scheduling Issues
      - WaitForSingleObject
      - Primary Countless Loop
  - Windows Application Subsystem
    - Console (/SUBSYSTEM:CONSOLE)
      - Main – ASCII Entry Point
      - wMain – Unicode Entry Point
    - Graphical (/SUBSYSTEM:WINDOWS)
      - wWinMain – ASCII Entry Point
      - wWinMain – Unicode Entry Point
    - Native (/SUBSYSTEM:NATIVE)
    - EFIApp (/SUBSYSTEM:EFI_APPLICATION)
    - Posix (/SUBSYSTEM:POSIX)
  - Process Property
    - Handle Instance
    - Previous Handle Instance
    - Command Line

        - Environment Variables
        - Affinity
        - Error Mode
        - Current Drive and Directory
- Creating Process Kernel Object
    - Multiprocess Application
        - Parent Process
        - Child Process
    - Process APIs
        - CreateProcess
        - CreateProcessAsUser
        - CreateProcessWithLogonW
    - Process Structures
        - STARTUPINFOEX
        - PROCESS_INFORMATION
    - Process Creation Flags
        - CREATE_BREAKAWAY_FROM_JOB
        - CREATE_SECURE_PROCESS
        - CREATE_NEW_CONSOLE
        - CREATE_NO_WINDOW
        - DETACHED_PROCESS
        - DEBUG_PROCESS
    - Process Search Paths
        - Current Directory
        - Windows Directory
        - Path Variable
        - Directory Hijacking
    - Parent-Child Debugging Issues with OllyDBG
        - Hooking CreateProcess Api
        - Creation Flags Modification
        - Debuggee Reattachment
        - Suspend Primary Thread
    - Process Termination Issue
        - CRT Clean-Up
        - ExitThread API
        - ExitProcess API
        - TerminateProcess API

- User Access Control (UAC)
  - Introduction to UAC
    - Past Windows XP
    - Post Windows Vista
      - Standard Token
      - Privileged Token
  - Executable Manifest File
    - Standalone .manifest File
    - Embedded .manifest File
  - Access Privilege Types
    - asInvoker
    - highestPrivilege
    - requireAdministrator
  - TOKEN_ELEVATION_TYPE
    - TokenElevationTypeDefault
    - TokenElevationTypeFull
    - TokenElevationTypeLimited
  - Privilege Token APIs
    - IsUserAnAdmin
    - OpenProcessToken
    - GetTokenInformation
    - CreateWellKnownSid
  - Reversing an Real Project
- **Job Kernel Object**
  - Job Object Overview
    - Why we need Job Object?
      - Security Restrictions
      - Privileges Restrictions
      - Functionalities Restrictions
      - CPU Resource Usage Restrictions
    - Linux Philosophy
      - Parent and Child Model
      - Process Groups
      - Process Forking
    - Windows Philosophy
      - Parent and Child Model
      - Process Tree
      - Job Object

- Creating Job Object
  - CreateJobObject API
- Placing Restrictions on a Job's Processes
  - Restriction Types
    - JobObjectBasicUIRestrictions
    - JobObjectBasicLimitInformation
    - JobObjectSecurityLimitInformation
    - JobObjectExtendedLimitInformation
  - SetInformationJobObject
- Placing a Process in a Job
  - CreateProcess API
  - Primary Thread Suspend
  - AssignProcessToJobObject API
- Terminating All Processes in a Job
  - TerminateJobObject
- Querying Job Statistics
  - QueryInformationJobObject API
  - GetQueuedCompletionStatus API
  - GetProcessIoCounters API
- Job Notifications
  - Job Event Notifications
    - JOB_OBJECT_MSG_ACTIVE_PROCESS_ZERO
    - JOB_OBJECT_MSG_ACTIVE_PROCESS_LIMIT
    - JOB_OBJECT_MSG_PROCESS_MEMORY_LIMIT
    - JOB_OBJECT_MSG_JOB_MEMORY_LIMIT

Thread Kernel Object

- Introduction to Threads
  - Thread Kernel Object
  - Thread Stack
  - Thread Local Storage
- Thread HandleThreads Infrastructure
  - Windows Thread:
    - std::thread
    - _beginthreadex
    - CreateThread API
    - RemoteCreateThread
  - Linux Thread:
    - std::thread
    - pthread_create

- clone syscall

- Primary Thread Invocation
  - Unicode and Ascii
    - main & wmain
    - WinMain & wWinMain
    - _tmain & _tWinMain
  - Thread Internals
    - Thread Context and Properties
    - RtlUserThreadStart API
    - Structured Exception Handler (SEH)
    - Context Switching
      - CONTEXT structure
      - GetThreadContext
      - SetThreadContext
- Preemptive Operating System
  - Special Windows Services
    - Thread Ordering Group Service
    - Multimedia Class Scheduler Service
    - JUCE Multimedia Library
  - Scheduler Priority Classes and Levels
    - Process Priority Classes
      - IDLE_PRIORITY_CLASS
      - BELOW_NORMAL_PRIORITY_CLASS
      - NORMAL_PRIORITY_CLASS
      - ABOVE_NORMAL_PRIORITY_CLASS
      - HIGH_PRIORITY_CLASS
      - REALTIME_PRIORITY_CLASS
    - Thread Priority Levels
      - THREAD_PRIORITY_IDLE
      - THREAD_PRIORITY_LOWEST
      - THREAD_PRIORITY_BELOW_NORMAL
      - THREAD_PRIORITY_NORMAL
      - THREAD_PRIORITY_ABOVE_NORMAL
      - THREAD_PRIORITY_HIGHEST
      - THREAD_PRIORITY_TIME_CRITICAL
    - Windows Priority Mappings
      - Windows Mappings Table
      - Base Priority Calculation
    - Windows Priority APIs:
      - SetPriorityClass
      - SetThreadPriority
      - GetPriorityClass
      - GetThreadPriority

- System Dynamic Boost Priority
  - Why system boost priority?
    - Priority Inversion Problem
    - Threads Starvation Problem
  - Boost Priority APIs:
    - SetProcessPriorityBoost
    - SetThreadPriorityBoost
    - GetProcessPriorityBoost
    - GetThreadPriorityBoost
  - I/O Request Priorities
- Windows Affinities
  - Affinity Model
    - Soft Affinity
    - Hard Affinity
  - Windows Hard Affinity
    - GetSystemInfo
    - SetProcessAffinityMask
    - GetProcessAffinityMask
    - GetThreadAffinityMask