Desenvolvendo KMDF Drivers

Objetivo

O objetivo deste curso é preparar os alunos que queiram entender, testar, complementar ou construir drivers para Windows empregando os conceitos do Windows Driver Foundation (WDF) aplicados ao Kernel, utilizado o Kernel Mode Driver Framework (KMDF).

Público alvo

Este curso é destinado aos desenvolvedores ou estudantes que precisam entender os conceitos fundamentais sobre implementação de drivers para Windows. Este curso não abordará implementações específicas de drivers, tais como impressoras, vídeo, SCSI, NDIS, USB, 1394 ou UMDF.

Pré-requisitos

- Os estudantes deverão ter sólidos conhecimentos da linguagem C tais como estruturas, ponteiros, heap, utilização da pilha, alocação dinâmica de memória e listas. O uso de linguagem C++ não será empregado neste curso. Os estudantes também deverão conhecer conceitos de depuração de software.
- Conhecimentos básicos da API do Windows tais como lidar com arquivos, handles, eventos, threads e processos.
- Conceitos básicos de Sistemas Operacionais tais como User-Mode versus Kernel-Mode, memória virtual, threads, processos, concorrência, registradores e interrupções.

Metodologia

Os tópicos são abordados de maneira simples e gradativa de forma a colocar em prática cada novo tópico apresentado. O curso é repleto de atividades práticas que despertam novas dúvidas enquanto o estudante ainda está em curso, tendo assim a oportunidade de consultar um profissional da área. O curso é todo apresentado em slides e acompanha material impresso.

Tópicos abordados

- Visão geral da arquitetura do Sistema
    - Processos e Threads
    - Memória Virtual e Paginação
    - Kernel Mode x User Mode
    - Subsistemas e API nativa
    - IoManager
    - Árvore de dispositivos e Plug-and-Play
    - Object Manager
    - Camada de abstração de Hardware (HAL)

- Ambiente (obtenção, instalação e utilização)
    - Windows Driver Foundation Kit
    - Microsoft Visual Studio Express
    - Microsoft Debugging Tools for Windows
    - Símbolos

- Escrevendo um Driver
    - WDF e o WDM
    - Objetos KMDF
    - Hierarquia de Objetos
    - Estruturas e Callbacks
    - Escrevendo Driver básico
    - Compilando o Driver
    - Instalando o Driver (Legacy)
    - Depurando o Driver
    - Criando Device Object
    - Symbolic Links
    - I/O Request Objects
    - Buffers e Memory Objects
    - I/O Queues
    - I/O Targets
    - I/O Event Callbacks
    - IOCTLs e DeviceIoControl
    - Implementando Dispatch Routines
    - Objetos, Handles e Ponteiros
    - Contexto Arbitrário
    - IRQLs, APCs, DPCs e WorkItems
    - Sincronização
    - Eventos e Timers

- Escrevendo Filtros
    - Filtros para Drivers Legacy
    - Repassando Requests
    - I/O Completion Callbacks
    - Tratamento de Requests Pendentes
    - Criando Requests para outros Drivers
    - Cancelamento de Requests

- Plug-and-Play Manager
    - Estados do Plug-and-Play
    - Iniciando e interrompendo devices
    - Sincronizando I/O
    - Notificações do PnP Manager
    - Escrevendo a EvtDriverDeviceAdd
    - Enumeração de Devices Filhos

- Power Manager
    - Estados do Sistema
    - Estados do Device
    - Recebendo notificações de transição
    - Desligando o Device quando Idle
    - Acordando o Device
    - Acordando o Sistema

- Interações com Hardware
    - Recebendo e Mapeando Recursos de I/O
    - Port I/O
    - Memória mapeada
    - Interrupt Objects
    - Sincronizando interrupções
    - DMA

- Instalações
    - Criando um arquivo INF
    - Instalando um Driver
    - Desinstalando um Driver

- Referências
    - Web Sites
    - Grupos de discussão
    - Livros