

Resumen del contenido

| | |
|--|-----------|
| <i>Agradecimientos</i> | xxiii |
| <i>Introducción</i> | xxvii |
| Capítulo 1. La API Win32 y las plataformas que la admiten | 1 |
| <p>Este capítulo explica las diversas APIs y plataformas de sistema operativo de Microsoft; clarifica las últimas palabras claves (Win32, Win32s, Windows CE y Windows NT); y trata los objetivos de Microsoft para cada API. Al final del capítulo, comprenderá por qué cree que es un momento muy excitante para los desarrolladores de software y por qué la API Win32 es el área en la que los ingenieros software deberían estar concentrando sus esfuerzos de desarrollo.</p> | |
| Capítulo 2. Objetos de núcleo | 9 |
| <p>Para trabajar con eficiencia con los servicios del sistema de Win32, debe tener unos conocimientos sólidos de los objetos de núcleo y los descriptores que los representan. Este capítulo explica la teoría acerca de la creación, apertura y cierre de los diversos objetos de núcleo. También se tratan los aspectos relacionados con la robustez y la seguridad. Este capítulo también muestra los tres métodos para compartir objetos de núcleo a través de los límites de proceso: herencia de descriptor, apertura de descriptores de objeto y duplicación de descriptores de objeto.</p> | |
| Capítulo 3. Procesos | 31 |
| <p>En Windows 95 y Windows NT, un ejemplar de una aplicación en ejecución es un proceso. Este capítulo explica cómo se puede crear un proceso nuevo, cómo se inicializa y cómo se destruye. También se explican varios atributos</p> | |

asociados a un proceso. Por ejemplo, en Win32 cada proceso tiene su propio espacio de direcciones, lo que significa que un proceso no puede afectar negativamente a otro proceso; esto no es cierto en Windows de 16 bits. Este capítulo también presenta los objetos de núcleo de Win32, que son la base para todas las tareas relacionadas con el núcleo tanto en Windows 95 como Windows NT. Se requieren unos conocimientos sólidos de los objetos de núcleo en cualquier desarrollador de Win32 serio.

Capítulo 4. Hilos 67

Los hilos son el corazón de las capacidades de multitarea de Windows 95 y Windows NT. Para que un proceso de Win32 haga algo realmente, debe tener hilos que ejecuten el código y que manipulen los datos contenidos en el proceso. Todos los procesos de Win32 contienen al menos un hilo, pero tanto Windows 95 como Windows NT permiten que un proceso contenga varios hilos que son planificados y apropiados por el sistema operativo. Este capítulo explica cómo se crean, planifican y destruyen los hilos.

Capítulo 5. La arquitectura de memoria de Win32 105

Los sistemas operativos avanzados como Windows 95 y Windows NT requieren arquitecturas de memoria sofisticadas. Este capítulo explica cómo gestiona el sistema la RAM y los archivos de paginación de la computadora en el disco fijo para proporcionar a cada proceso un espacio virtual de direcciones de 4 GB. El capítulo también trata cómo se particiona el espacio de direcciones de cada proceso y las diferencias de implementación entre Windows 95 y Windows NT. Una de las características que hacen única a la arquitectura de Win32 es que Win32 separa la tarea de reservar las regiones del espacio de direcciones de la tarea de comprometer el almacenamiento físico de estas regiones. Este capítulo explica cómo se realizan estas dos tareas, además de cómo se pueden asignar los atributos de protección a las páginas de almacenamiento físico comprometido.

Capítulo 6. Exploración de la memoria virtual 139

Este capítulo continúa con el Capítulo 5. Presenta varias funciones de Win32 que permiten explorar la configuración de memoria del sistema y el contenido (código y datos) del espacio de direcciones de un proceso.

Capítulo 7. Utilización de la memoria virtual en las aplicaciones 177

Este capítulo muestra cómo utilizar técnicas de gestión de memoria virtual en nuestras propias aplicaciones. Se incluye la reserva de regiones en el espacio de direcciones de un proceso y técnicas para conocer cuándo comprometer el

almacenamiento físico de estas regiones. Estas técnicas permiten que una aplicación utilice el almacenamiento físico de manera más eficiente de lo que lo haría bajo cualquier otro sistema operativo.

Capítulo 8. Archivos proyectados en memoria 215

Windows 95 y Windows NT utilizan archivos proyectados en memoria para implementar la gestión de memoria virtual. Todo el código y los datos de un proceso son respaldados por un archivo en disco: un archivo proyectado en memoria. Este capítulo muestra cómo los archivos proyectados en memoria hacen que sea casi trivial el manipular los archivos en disco. El capítulo también muestra cómo utilizar los archivos proyectados en memoria para compartir código y datos entre múltiples procesos.

Capítulo 9. Montículos 265

Los montículos son la tercera y última técnica de gestión de memoria ofrecida por la API Win32. Este capítulo muestra cómo crear múltiples montículos dentro de un solo proceso y explica por qué a un desarrollador le podría interesar hacerlo. Este capítulo también trata cómo se emulan las funciones de montículo de Windows de 16 bits en la API Win32 bajo Windows 95 y Windows NT para hacer que sea más sencillo para los desarrolladores trasladar el código fuente existente de Windows de 16 bits.

Capítulo 10. Sincronización de hilos 287

Siempre que haya múltiples hilos ejecutándose simultáneamente o cuando se les interrumpa por una apropiación, una aplicación necesitará a menudo detener a un hilo para evitar que se corrompan los datos. Windows 95 y Windows NT ofrecen varios objetos para realizar la sincronización de hilos; este capítulo trata estos objetos y describe las técnicas para utilizarlos.

Capítulo 11. Mensajes de ventana y entrada asíncrona 411

Uno de los mayores problemas en Windows de 16 bits es que es demasiado sencillo que una aplicación cuelgue a todas las aplicaciones en ejecución. Los sistemas operativos robustos, como Windows 95 y Windows NT no permiten que las aplicaciones comprometan la ejecución de los otros procesos del sistema. Para crear un entorno robusto, en Win32 los mensajes de ventana y la entrada hardware se manejan de una manera diferente de la de Windows de 16 bits. Estos cambios pueden afectar a algún código fuente escrito originalmente para Windows de 16 bits. Este capítulo explica cómo se procesan los mensajes de ventana en Windows 95 y Windows NT.

Capítulo 12. Bibliotecas de enlace dinámico 467

Las bibliotecas de enlace dinámico (DLLs) han sido siempre la piedra angular de todas las aplicaciones de Windows —y lo siguen siendo en Windows 95 y Windows NT—. Sin embargo, las DLLs se manejan de una forma bastante distinta en Win32 en comparación con la de Windows de 16 bits. Este capítulo explica cómo se proyecta una DLL en el espacio de direcciones de un proceso y cómo se inicializa correctamente una DLL. Además, este capítulo muestra cómo se puede utilizar una DLL para exportar tanto funciones como datos.

Capítulo 13. Almacenamiento local de hilo 513

En un entorno en el cual hay múltiples hilos ejecutándose concurrentemente, es importante asociar los objetos de datos y las variables a los hilos individuales de un proceso. Este capítulo describe cómo utilizar las técnicas de almacenamiento local de hilo dinámico y estático para asociar datos a hilos específicos.

Capítulo 14. Sistemas de archivo 541

Este capítulo muestra cómo puede una aplicación manipular los diversos sistemas de archivos ofrecidos por Windows 95 y Windows NT: FAT, CDFS y NTFS. También cubre cómo crear, eliminar y recorrer directorios. El capítulo muestra después cómo mover, borrar y examinar los atributos de los archivos en el sistema de archivos. Finalmente, este capítulo muestra cómo escribir una aplicación a la que se le notifiquen los cambios que se producen en el sistema de archivos.

Capítulo 15. E/S a dispositivo 611

Este capítulo examina cómo realizar E/S con dispositivo en Win32. La E/S con dispositivo incluye la lectura y escritura en archivos y en otros dispositivos como mailslots, encauzamientos, puertos serie y conectores. Concretamente, este capítulo muestra cómo realizar la E/S síncrona y varias formas de técnicas de E/S asíncrona, incluyendo la E/S avisable y los puertos de finalización de E/S.

Capítulo 16. Gestión estructurada de excepciones 675

La Gestión estructurada de excepciones (SEH) es un mecanismo que permite a los desarrolladores de aplicaciones escribir aplicaciones más robustas y fiables. La SEH consta de dos componentes que funcionan en común: la gestión de excepciones y la gestión de terminación. La gestión de excepciones es un mecanismo que permite que una aplicación capture las excepciones hardware

y software (por ejemplo, accesos inválidos a memoria). La gestión de terminación garantiza que se realizan las tareas de limpieza incluso si se produce una excepción.

Capítulo 17. Unicode 757

Los desarrolladores de software están encontrando un gran potencial en la distribución del software en los mercados internacionales. Para ayudar a los desarrolladores, Microsoft incorporó la compatibilidad con Unicode en Windows NT y una compatibilidad limitada con Unicode en Windows 95. Unicode es un conjunto de caracteres de 16 bits que permite que los desarrolladores manipulen con facilidad caracteres y cadenas para lenguajes y sistemas de escritura diferentes. Este capítulo estudia cómo se pueden aprovechar al máximo las características de Unicode de Win32 para ayudar a distribuir los proyectos de desarrollo.

Capítulo 18. Traspasando la barrera de los límites del proceso 779

La naturaleza robusta de los entornos de Windows 95 y Windows NT hace que sea muchos más difícil manipular otros procesos del sistema. Aunque no es normal que los procesos se alteren entre sí, algunos procesos, como los depuradores y otras herramientas, requieren un conocimiento íntimo de los otros procesos para poder ser útiles para el desarrollador de software. Este capítulo muestra tres técnicas que permiten que un proceso inserte una DLL en el espacio de direcciones de otro proceso. Para comprender estas técnicas se requiere conocer los procesos, los hilos, la memoria virtual, la sincronización de hilos, los mensajes de ventana, las DLLs, la gestión estructurada de excepciones y Unicode.

Apéndice A. Fibras 841

Las fibras son hilos ligeros que se han añadido a Win32 para simplificar a los desarrolladores de software el traslado de las aplicaciones basadas en UNIX existentes a Win32. Este apéndice explica lo que son las fibras y cómo aprovecharlas. También se trata el diseño de nuevas aplicaciones que aprovechen al máximo las capacidades de las fibras.

Apéndice B. Troceadores de mensajes 855

La mayor parte de los programadores de Windows no han oído nunca hablar de los troceadores de mensajes, aunque estas ayudas a la programación existen tanto en Windows de 16 bits como en Win32. Este apéndice explica cómo utilizar los troceadores de mensajes para ayudarnos a escribir, leer y mantener nuestro código fuente. Además, los troceadores de mensajes ha-

cen que sea mucho más sencillo el traslado del código fuente de Windows de 16 bits a Win32 y viceversa.

Apéndice C. El entorno de construcción 863

Este apéndice explica el archivo de cabecera CMNHDR.H que se incluye en todas las aplicaciones ejemplo presentadas en este libro. Este archivo de cabecera contiene varios *#defines*, *#pragmas* y directivas del enlazador. El Apéndice C también trata a qué valor se han establecido varios conmutadores del compilador y el enlazador en el archivo make de proyecto de cada aplicación ejemplo.

Índice 875