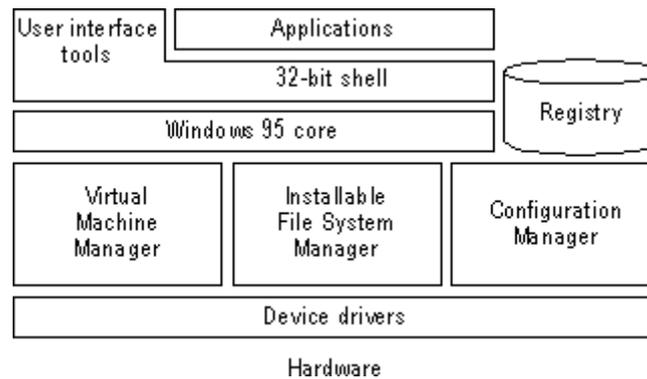


Arquitectura de Windows 95

Artículo original: <http://technet.microsoft.com/en-us/library/cc751120.aspx>
© 2013 Microsoft Corporation – Traducción al castellano de [Toto Aguiar](#).

Windows 95 presenta un nuevo modelo de controlador de dispositivo, un nuevo sistema de archivos, un nuevo motor de gráficos de 32 bits, un nuevo subsistema de impresión, comunicación y multimedia de 32 bits. Windows 95 es un sistema operativo de 32 bits con soporte de conectividad integrada. Proporciona un alto rendimiento, robustez y una compatibilidad total.

Todas estas características son compatibles con la estructura modular se muestra en el siguiente diagrama:



En este capítulo se describen los componentes clave que conforman la arquitectura de Windows 95 partiendo de su almacenamiento de información central, el registro y yendo de abajo hacia arriba.

Resumen de las Mejoras sobre Windows 3.1

A pesar de que su diseño arquitectónico se basa en la versión de Windows 3.1, Windows 95 incluye varias mejoras respecto de su antecesor:

- Un sistema operativo completamente integrado de 32 bits en modo protegido de funcionamiento, lo que elimina la necesidad de una copia separada de MS-DOS.
- Multitarea con expulsión y soporte multihilo, lo que mejora la capacidad de respuesta del sistema y ofrece un procesamiento fluido en segundo plano.
- Sistemas de archivos de 32 bits instalables como VFAT, CDFS y redirectores de red. Estos brindan un mejor rendimiento, permiten el uso de nombres de archivo largos, y otorgan una arquitectura abierta para soportar el crecimiento futuro.
- Controladores de dispositivos de 32 bits, disponibles en todo el sistema, que ofrecen un mejor rendimiento y uso de memoria inteligente.
- Un completo kernel de 32 bits que incluye la gestión de la memoria y la programación y la gestión de procesos.
- Mejora de todo el sistema de robustez y limpieza después de que una solicitud finaliza o falla. Esto proporciona un entorno operativo más estable y confiable.
- Mejora de la capacidad del sistema, que permite a múltiples aplicaciones y tareas del sistema ejecutarse simultáneamente sin inconvenientes.

El Registro de Windows 95

El registro es la base de datos central para Windows 95. Esta base de datos jerárquica simplifica mucho el sistema operativo y lo hace más adaptable. También simplifica el sistema operativo, eliminando la necesidad del AUTOEXEC.BAT, CONFIG.SYS y los archivos INI (excepto cuando las aplicaciones heredadas los requieran). Esto hace al sistema operativo más adaptable al almacenar información y configuraciones específicas del usuario, permitiendo compartir una computadora entre múltiples usuarios y tener más de una configuración para cada equipo.

La principal función del registro de Windows 95 es servir como un repositorio central de información específica del hardware para usarla en la detección de nuevos componentes de hardware y Plug and Play. Windows 95 mantiene la información sobre los componentes y dispositivos de hardware que han sido identificados a través de un proceso de enumeración en la estructura jerárquica del registro. (Éste proceso es el “Gestor de Configuración” ó “Configuration Manager” que veremos luego). Cuando un dispositivo nuevo es instalado, el sistema chequea la configuración existente en el registro para determinar los recursos de hardware (IRQs, direcciones E/S, canales de DMA, etc) que no están siendo utilizados, entonces el nuevo dispositivo se puede configurar apropiadamente, sin tener conflictos con un dispositivo ya instalado en el sistema

Windows 3.1 usaba los archivos de inicialización (INI) para almacenar información específica del estado o configuración del sistema o de algunas aplicaciones. Por ejemplo, el archivo WIN.INI almacenaba información específica de sistema sobre el hardware y drivers de configuración de dispositivos, y varios archivos INI que almacenaban información específica de las aplicaciones.

El Registro soluciona esto proveyendo una ubicación única para la configuración de la información de la computadora.

En la siguiente tabla se muestran otras dificultades o limitaciones causadas por el uso de archivos INI, que son corregidos por el uso del Registro:

Problemas usando archivos INI	Solución usando el Registro
Los archivos INI son en formato texto, y tienen una limitación de tamaño de 64K	El Registro no tiene restricción de tamaño y puede incluir valores binarios y de texto
La información almacenada en archivos INI no es jerárquica y sólo admite dos niveles de información (específicamente, encabezados de sección con una lista de nombres clave en cada uno)	El Registro está organizado jerárquicamente
Muchos archivos INI contienen una gran cantidad de switches y entradas complicadas de configurar o que son usados sólo por los componentes del sistema operativo	El Registro contiene valores mas estandarizados. Para más información vea el Capítulo 6 “debate de configuración”
Los archivos INI no proporcionan mecanismos para almacenar información específica del usuario por lo que es difícil que muchos usuarios compartan una computadora., o que usuarios que se muevan en la red tengan acceso a sus ajustes específicos de usuario	La clave de Registro HKEY_USERS almacena información específica del usuario.
La información de configuración en archivos INI es local para cada sistema, y no hay mecanismos de API disponibles para administrar remotamente la configuración, por lo que es difícil manejar varios equipos	El registro puede administrarse remotamente y las directivas del sistema pueden ser descargadas de un servidor central cada vez que un nuevo usuario se loguea.

Cuando se actualiza de Windows 3.1 a Windows 95, la información específica del sistema, como la referencia estática para cargar los drivers de dispositivos virtuales es movido, según el caso, a partir del archivo System.ini hacia el registro.

Para la compatibilidad con versiones anteriores, Windows 95 no ignora AUTOEXEC.BAT, CONFIG.SYS y los archivos INI, ya que muchas aplicaciones basadas en Win 16 todavía lo utilizan. Por ejemplo Windows 95 permite a las aplicaciones basadas en Win16 utilizar archivos INI para los parámetros, drivers de dispositivos, etc, que las aplicaciones necesitan para correr. Además Windows 95 sigue examinando la sección [386Enh] de System.ini al inicio para comprobar si hay controladores de controladores de dispositivos virtuales.

Una de las ventajas del registro para las aplicaciones basadas en Win32, es que muchas de las APIs del Registro basado en Win32 puede ser usado remotamente a través de llamadas a procedimiento remoto (RPC - Remote Procedure Call) que es un mecanismo de Windows 95 para facilitar el acceso a la información del registro a través de una red. Esto significa que los administradores de red pueden utilizar las herramientas de administración del sistema para acceder al contenido del registro desde cualquier computadora en la red (por supuesto, la computadora remota debe estar configurada para permitir administración remota, y debe tener seguridad de niveles de usuarios)

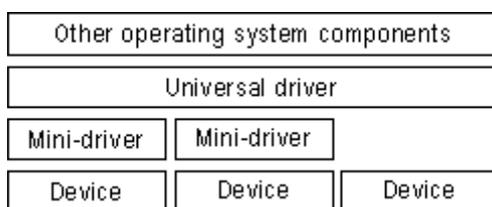
con la administración remota de Windows 95, los mecanismos de gestión de la industria, tales como el Protocolo Simple de Administración de Red (SNMP – Simple Network Management Protocol) puede ser integrado fácilmente a Windows 95, simplificando la administración y soporte de carga al administrador de la red . Para más información vea el Capítulo 16, “administración remota”

Controladores de Dispositivo

Windows 95 proporciona soporte mejorado para dispositivos de hardware y periféricos, incluyendo discos, adaptadores de pantalla, mouse y otros dispositivos señaladores, módems, máquinas de fax e impresoras.

En Windows 3.1 los drivers de dispositivos eran en mayor parte monolíticos y complejos de desarrollar. Windows 95 utiliza una arquitectura de driver/mini-driver universal que hace que sea mas sencillo para los fabricantes de hardware proveer código de drivers específicos para su dispositivo.

Un driver universal incluye la mayor parte del código necesario para los dispositivos de una clase particular (como impresoras o módems) para comunicarse con los componentes correctos del sistema operativo (como impresión o subsistemas de comunicación) Un mini-driver es un controlador relativamente pequeño y sencillo que contiene las instrucciones adicionales necesarias para un dispositivo específico. Sin embargo en muchos casos el driver universal para una categoría de dispositivos, también contiene el código necesario para operar dispositivos diseñados para el estándar más común de esa categoría (por ejemplo el driver unimodem funciona con todos los módems que soportan comandos AT)



Un controlador de dispositivo virtual (VxD) es un controlador de 32 bits, de modo protegido que gestiona un recurso del sistema, tal como un dispositivo de hardware o software instalado, de modo que más de una aplicación puede utilizar el recurso al mismo tiempo. VxD se refiere a un controlador de dispositivo virtual en general -la x representa el tipo de controlador de dispositivo. Por ejemplo, un controlador de dispositivo virtual para un dispositivo de visualización se lo conoce como VDD, un controlador de dispositivo virtual para un dispositivo temporizador es un ETV, un controlador de dispositivo virtual para un dispositivo de impresora es un VPD, y así sucesivamente.

En Windows 3.1 los VxD se cargaban estáticamente y tomaba una gran cantidad de espacio en memoria. En cambio en Windows 95 éstos se cargan dinámicamente (sólo aquellos que son necesarios en un momento dado son los que se cargan en la memoria). Además, los nuevos VxD no requieren ser alojados completamente en memoria, aumentando aún más la cantidad de memoria disponible en el sistema.

Los drivers VxD brindan soporte a todos los dispositivos de hardware típicos de un ordenador, incluyendo controladores de disco serie y paralelo, dispositivos de visualización, teclados, etc. si el estado del dispositivo de hardware puede ser interrumpido por el cambio entre múltiples aplicaciones, el dispositivo debe tener un dispositivo virtual correspondiente, y VxD. El dispositivo virtual realiza un seguimiento del estado del dispositivo para cada aplicación y se asegura que el dispositivo está en el estado correcto cada vez que una aplicación continúa.

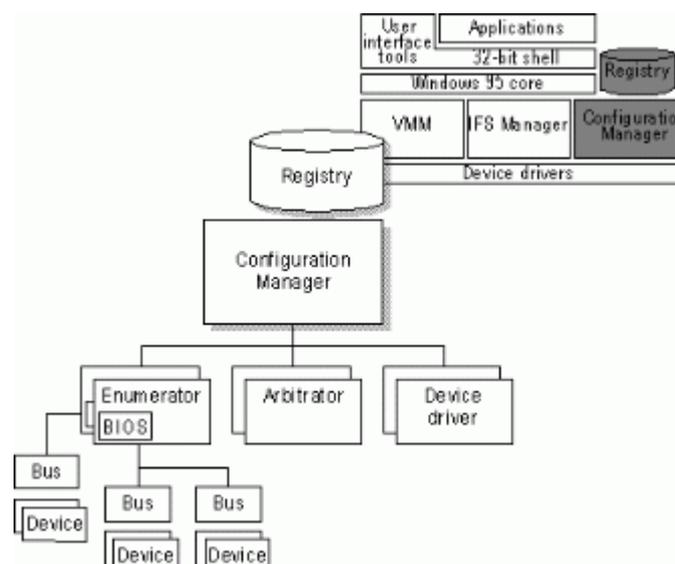
Aunque la mayoría de los dispositivos administran hardware, algunos sólo administran software instalado, como por ejemplo un driver de dispositivo de MS-DOS o un programa TRS. Estos dispositivos virtuales contienen código para emular el software o asegurarse que dicho software utilice datos que se aplican únicamente para la aplicación que está en ejecución. También se utilizan VxD para mejorar el rendimiento del software.

(Los archivos de drivers de dispositivos virtuales en Windows 95 tienen extensión .vxd; los controladores de Windows 3.1 utilizan la extensión .386.)

Administrador de Configuración

Para dar soporte a la funcionalidad Plug and Play, la arquitectura de Windows 95 incluye un nuevo componente llamado Administrador de Configuración (Configuration Manager), el cual orquesta el proceso de configuración. Éste proceso puede involucrar muchos buses y arquitecturas de dispositivos coexistentes en un único sistema, con mas de un tipo de dispositivo utilizando la misma arquitectura de bus, aunque cada dispositivo tenga un conjunto de requisitos de configuración separado. (un bus es el mecanismo que permite que la información sea transferida entre la computadora y el dispositivo) Por ejemplo, un mouse y un teclado pueden utilizar el mismo controlador de bus, una unidad de CD-ROM y un disco rígido pueden utilizar el mismo bus SCSI.

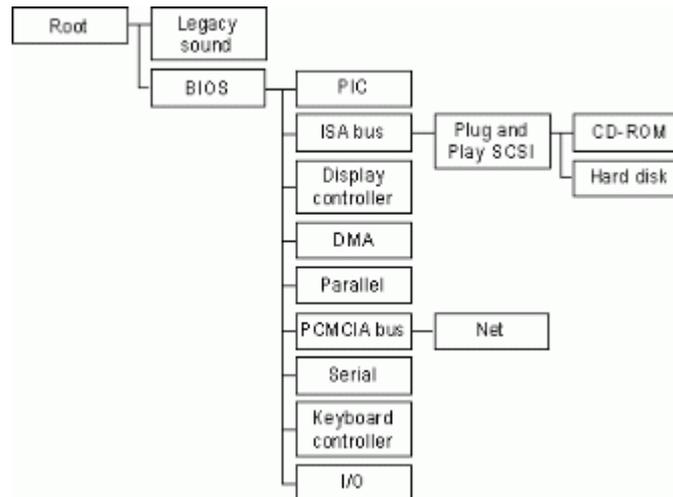
Como se muestra en la siguiente ilustración, el Administrador de Configuración trabaja con una serie de subcomponentes para identificar cada autobús y cada dispositivo en el sistema, y para identificar las opciones de configuración para cada dispositivo. El Administrador de Configuración asegura que cada dispositivo en el equipo puede utilizar una IRQ, direcciones de puerto de E / S y otros recursos sin conflictos con otros dispositivos.



El Administrador de configuración también ayuda a monitorear en el equipo por cambios en el número y tipo de dispositivos presentes y administra la reconfiguración de los dispositivos, según sea necesario, cuando estos cambios tienen lugar. A medida que ocurren, El Administrador de Configuración informa de estos eventos a las aplicaciones.

Para llevar a cabo su función, el Administrador de Configuración (implementado como parte del Administrador de memoria virtual, VMM32) pide a los enumeradores de bus identificar todos los dispositivos en los buses específicos y sus respectivas necesidades de recursos.

Los enumeradores bus son nuevos drivers responsables de crear el árbol de hardware de Windows 95. Un árbol de hardware es la representación jerárquica de todos los buses y dispositivos en una computadora. Cada bus y cada dispositivo está representado como un nodo. La siguiente es una representación gráfica de un árbol de hardware.



Durante el proceso de enumeración de dispositivos, el enumerador de bus localiza y reúne información de cada controlador de dispositivo o de los servicios de BIOS para ese tipo de dispositivo en particular. Por ejemplo, el enumerador de bus SCSI llama a los controladores SCSI para obtener información. (Algunos enumeradores de bus pueden comprobar el hardware directamente.)

Para cada dispositivo, se carga un controlador. Cuando se carga, el driver espera al Administrador de configuración para asignar recursos específicos (como IRQ) para el dispositivo. Éste último llama a los árbitros de recursos para asignar los recursos para cada dispositivo.

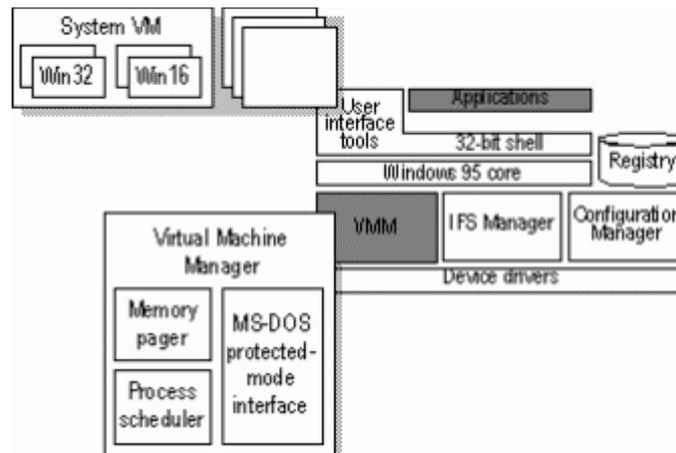
Los árbitros de recursos resuelven conflictos entre los dispositivos que soliciten asignaciones de recursos idénticos. Windows 95 proporciona árbitros para la memoria estándar de E/S, interrupciones de hardware y DMA. (Los árbitros son componentes separados, en lugar de una parte del Administrador de Configuración, para garantizar la extensibilidad a nuevos tipos de recursos, como la asignación de potencia o las identificaciones automáticas de SCSI)

Para completar el proceso de configuración, el Administrador de Configuración informa a los drivers de dispositivo acerca de la configuración del dispositivo. Este proceso se repite cuando la BIOS o uno de los enumeradores de bus informan al Administrador de Configuración de un evento que requiere un cambio en la configuración del sistema, tal como la eliminación o inserción de una tarjeta Plug and Play-PCMCIA compatible. Para obtener más información acerca de Plug and Play, consulte el cap 18 “Introducción a la configuración del sistema”

Gestor de Máquina Virtual

Así como Administrador de Configuración proporciona todos los recursos necesarios para cada dispositivo en el equipo, otro componente, el Gestor de Máquinas Virtuales (Virtual Machine Manager), proporciona los recursos necesarios para cada aplicación y proceso del sistema que se ejecuta en el ordenador. Gestor de Máquinas Virtuales crea y mantiene los entornos de máquinas virtuales en los que las aplicaciones y procesos del sistema se ejecutan.

Una máquina virtual (VM) es un entorno en el que la memoria, desde la perspectiva de la aplicación, se ve como si fuera una computadora independiente, completa con todos los recursos disponibles en el equipo físico que una aplicación necesita para ejecutarse. El Gestor de Máquinas Virtuales proporciona a cada aplicación los recursos del sistema que necesita. Este componente reemplaza WIN386.EXE en Windows 3.1.



Windows 95 tiene una sola VM llamada máquina virtual del sistema, en el que se ejecutan todos los procesos del sistema. Todas las aplicaciones basadas en Win32 y Win16 se ejecutan en ésta máquina virtual. Cada aplicación basada en MS-DOS se ejecuta en su propia máquina virtual. (Para obtener información sobre las máquinas virtuales, consulte "Componentes básicos del sistema" más adelante en este capítulo.)

El administrador de máquinas virtuales es responsable de tres áreas clave de servicios:

- Programación de procesos
- Paginación de memoria
- Soporte para modo MS-DOS, para que las aplicaciones basadas en MS-DOS puedan tener acceso exclusivo a los recursos del sistema

En las siguientes secciones hablaremos de estas tres áreas de servicio

Programación de Procesos y Multitarea

El planificador de procesos es el componente responsable de proporcionar los recursos del sistema para las aplicaciones y otros procesos que se ejecutan, y planificar los procesos para permitir que varias aplicaciones se ejecuten simultáneamente.

También permite planificar procesos de manera de permitir a múltiples aplicaciones y otros procesos ejecutarse concurrentemente. Windows 95 usa dos métodos para programación de procesos concurrentes: multitarea cooperativa y multitarea con expulsión

Con Windows 3.1 las aplicaciones corrían concurrentemente a través de un método conocido como multitarea cooperativa. Usando este método, el sistema operativo necesita una aplicación para comprobar la cola de mensajes periódicamente y ceder el control del sistema a otras aplicaciones en ejecución. Las aplicaciones que no comprueben la cola de mensajes con frecuencia podrían monopolizar la CPU y evitar que el usuario cambie a otra aplicación. Por razones de compatibilidad, Windows 95 utiliza multitarea

cooperativa para las aplicaciones basadas en Win16.

Windows 95 utiliza multitarea con expulsión para aplicaciones basadas en Win32. Esto significa que el sistema operativo lleva el control desde, o da el control a otra tarea en ejecución, en función de las necesidades del sistema.

A diferencia de las aplicaciones basadas en Win16, las aplicaciones basadas en Win32 no tienen que ceder el paso a otras tareas en ejecución para realizar múltiples tareas correctamente. Las aplicaciones basadas en Win32 pueden utilizar multithreading, un mecanismo que Windows 95 proporciona para facilitar la capacidad de ejecutar aplicaciones simultáneamente. Una aplicación basada en Win32 que se ejecuta en el sistema se denomina proceso, en términos del sistema operativo. Cada proceso se compone de por lo menos un hilo de ejecución que identifica el flujo de ruta del código que se ejecuta en el sistema operativo. Un hilo es una unidad de código que puede obtener un intervalo de tiempo desde el sistema operativo para ejecutarse concurrentemente con otras unidades de código, y debe estar asociado a un proceso. Además, una aplicación basada en Win32 puede iniciar varios subprocesos para un proceso dado para mejorar la aplicación para el usuario, mejorando el rendimiento y la capacidad de respuesta y el procesamiento en segundo plano. Debido a la naturaleza de la multitarea con expulsión de Windows 95, los hilos de ejecución permiten que el código sea procesado sin problemas en segundo plano.

Nota: Algunas aplicaciones basadas en Win16 pueden haber proporcionado una funcionalidad similar a esta, sin embargo, debido a que Windows 3.1 no proporciona un mecanismo de soporte a las aplicaciones multihilos, correspondía a los proveedores de aplicaciones implementar sus propios esquemas de hilos. El uso de hilos en Windows 95 hace que sea fácil para los proveedores de aplicaciones agregar el procesamiento asíncrono de información a sus aplicaciones.

Paginación de Memoria

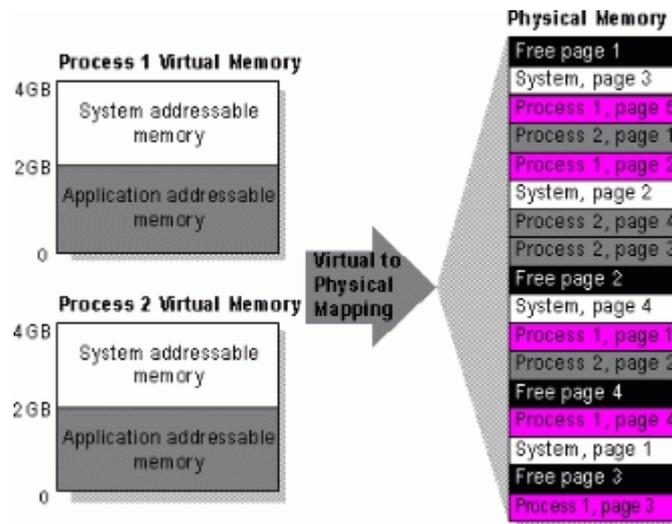
Windows 95, al igual que Windows NT, utiliza un sistema de memoria virtual de paginación bajo demanda. Este sistema se basa en un plano, un espacio de direcciones lineales accedido utilizando direcciones de 32 bits.

A cada proceso se le asigna un espacio de direcciones virtual único de 2 GB. Mientras que los 2 GB superiores son para memoria compartida, los dos 2 GB más bajos son privados para la aplicación. Este espacio de direcciones virtuales se divide en bloques iguales o páginas.

La paginación por demanda es un método por el cual el código y los datos se mueven de las páginas de la memoria física a un archivo de paginación temporal en el disco. A medida que se necesite la información de un proceso, es buscado de nuevo en la memoria física.

El paginador de memoria mapea las direcciones virtuales desde el espacio de direcciones del proceso a las páginas físicas en la memoria de la computadora. Al hacerlo, se esconde la organización física de la memoria de los threads del proceso. Esto asegura que el thread pueda acceder a la memoria de su proceso, según sea necesario, pero no a la memoria de otros procesos.

Por lo tanto, como se muestra en la siguiente ilustración, una vista de thread de la memoria virtual de su proceso (como se muestra a la izquierda) es mucho más simple que la disposición real de las páginas en la memoria física.



Para brindar soporte a un entorno operativo de 16 bits, incluyendo Windows 3.1 y MS-DOS, la arquitectura del procesador Intel® utiliza un mecanismo llamado un segmento. Los segmentos referencian memoria utilizando una “dirección de segmento” de 16 bits, y una dirección de desplazamiento de 16 bits dentro del segmento. Un segmento es de 64 KB de tamaño, y las aplicaciones y el sistema operativo sufre una penalización en el rendimiento por acceder a la información a través de segmentos.

Windows 95 resuelve este problema mediante el uso de las capacidades de 32 bits de la arquitectura del procesador 80386 (y superiores) para soportar un modelo de memoria plana, modelo de memoria lineal para la funcionalidad del sistema operativo de 32 bits y las aplicaciones basadas en Win32. Un modelo de direccionamiento lineal simplifica el proceso de desarrollo de los proveedores de aplicaciones, y elimina las penalizaciones de rendimiento impuestos por la arquitectura de memoria segmentada.

Con este modelo de direccionamiento, Windows 95 permite el uso completo de los 4 GB de espacio de memoria direccionable para todos los componentes y aplicaciones del sistema operativo de 32 bits. Cada aplicación de 32 bits puede acceder hasta 2 GB de espacio de memoria direccionable, que es suficiente como para soportar incluso la más grande de las aplicaciones de escritorio.

Soporte para Modo MS-DOS

Aunque la mayoría de las aplicaciones basadas en MS-DOS ejecutan bien en Windows 95 y se pueden ejecutar simultáneamente con otras aplicaciones basadas en Win32 y Win16, un número pequeño de aplicaciones basadas en MS-DOS requiere acceso exclusivo a los recursos del sistema para funcionar. En este caso, es el Administrador de memoria virtual quien crea este ambiente de operación exclusiva para la aplicación, llamada “modo MS-DOS”. Cuando una de estas aplicaciones se ejecuta en modo MS-DOS, no se permite a otras aplicaciones o procesos competir por los recursos del sistema - todos los recursos son de acceso exclusivo de la aplicación basada en MS-DOS. Para obtener información relacionada, consulte el Capítulo 22, "Soporte de aplicaciones".

Sistemas de Archivos Instalables

Windows 95 cuenta con una arquitectura de sistema de archivos en capas que soporta múltiples sistemas de archivos, incluyendo VFAT y CDFS.

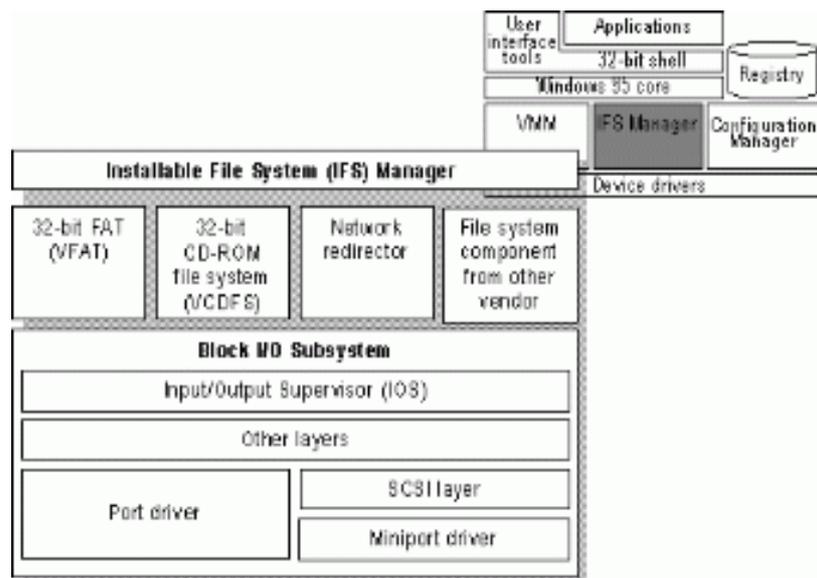
La nueva arquitectura de sistema de archivos hace que el equipo sea más fácil de usar y mejora el rendimiento de E/S de disco. Las características de la nueva arquitectura de sistema de archivos incluyen compatibilidad con nombres largos y un sistema dinámico de caché para el archivo y la E/S por red.

La compatibilidad con nombres largos de archivos mejora la facilidad de uso ya que los usuarios no necesitan hacer referencia a los archivos por la estructura de MS-DOS. En su lugar, los usuarios pueden especificar un máximo de 255 caracteres para identificar sus documentos. Además, los nombres de los archivos parecen menos crípticos y por lo tanto más fácil de leer porque Windows 95 oculta las extensiones de los archivos de los usuarios.

El rendimiento de E/S de Windows 95 ha mejorado drásticamente respecto de Windows 3.1. Esto se debe a las características de Windows 95: código de 32 bits, modo protegido para leer información desde y escribir información en el sistema de archivos, y para leer información desde y escribir información en el dispositivo de disco. También incluye mecanismos de caché dinámica de 32 bits, y código de 32 bits desde el sistema de archivos hasta el dispositivo de disco

Windows 95 incluye una arquitectura de sistema de archivo abierto para soportar futuros sistemas. También proporciona compatibilidad de controladores de dispositivo de disco con Windows NT.

La siguiente figura muestra la arquitectura del sistema de archivos utilizado por Windows 95.



La arquitectura del sistema de archivos de Windows 95 se compone de los siguientes componentes:

- *Administrador de Sistema de Archivos Instalable (IFS)*. El Administrador de IFS es responsable de arbitrar el acceso a los diferentes componentes del sistema de archivos.
- *Controladores del sistema de archivos*. La capa de controlador de sistema de archivos incluye el acceso a los dispositivos basados en FAT del disco, sistemas de archivos de CD-ROM y soporte de redirección de dispositivos de red.
- *Subsistema de E/S en bloque*. El subsistema de E/S en bloque se encarga de interactuar con el dispositivo de disco físico.

Las siguientes secciones describen estos componentes.

Administrador de Sistemas de Archivos Instalable.

En Windows 95, la clave de acceso de disco y dispositivos redirigidos es el administrador de sistema de archivos instalable (IFS). El administrador IFS arbitra el acceso a dispositivos que contengan file systems (sistemas de archivo).

En MS-DOS y Windows 3.1, la INT 21 proveía el acceso al file system manipulando la información del archivo en un dispositivo de disco. Para soportar el redireccionamiento de dispositivos de disco (por ejemplo, una unidad de red o una unidad de CD-ROM), otros componentes del sistema, como el redirector de red se conectaría a la función INT 21 para poder examinar la solicitud del sistema de archivos y determinar si debe manejar solicitud de E/S o dejar que el sistema de archivos base lo haga. Aunque este mecanismo proporciona la posibilidad de añadir más controladores de dispositivos, algunos componentes adicionales corrían mal e interferían con otros controladores instalados.

Otro problema con el file system basado en MS-DOS fue la dificultad en la carga de varios redirectores de red para proporcionar acceso simultáneo a diferentes tipos de redes. Windows para Trabajo en Grupo provee soporte para el funcionamiento del redirector de red de Windows y, al mismo tiempo, funciona como un redirector de red adicional, como Novell® NetWare® o Banyan® VINES®, sin embargo, no ofrece soporte para el funcionamiento de más de dos redirectores de red al mismo tiempo. En Windows 95, los redirectores de red se implementan como sistemas de archivos bajo el IFS Manager, por lo que se puede utilizar un número ilimitado de redirectores de 32 bits.

Drivers de Sistema de Archivos

En Windows 3.1 el los drivers de sistemas de archivos eran soportados por MS-DOS. Con Windows 95 los drivers del sistema de archivos son componentes de Anillo 0 del sistema operativo. Windows 95 posee soporte para todos los sistemas de archivos (otros FS pueden ser agregados por otros fabricantes)

- Driver de 32-bit FAT (VFAT)
- Driver de Sistema de archivos de CD-ROM de 32 bits (CDF5)
- Redirector de red de 32 bits para la conexión a los servidores de red de Microsoft, como Windows NT Server, junto con un redirector de red de 32 bits para conectarse a servidores Novell o NetWare

Para obtener información sobre los redirectores de red, vea el capítulo 32 “Arquitectura de red de Windows 95”

Sistema de Archivos VFAT

El controlador VFAT DE 32 bits proporciona una ruta de código en modo protegido para la manipulación del sistema de archivo almacenado en un disco. Debido a que es reentrante y multihilo, proporciona una mejora en el rendimiento de la multitarea.

Este controlador interactúa con el subsistema de E/S en bloque para proporcionar acceso al disco para más tipos de dispositivos que los que eran soportados en Windows 3.1. Windows 95 también admite el mapeo de cualquier driver de disco de modo real que pueda estar instalado. La combinación de acceso a archivos de 32 bits y acceso a drivers de disco de 32 bits resultan una mejora significativa en el rendimiento de la E/S de disco y archivo.

Los beneficios de los drivers de acceso a archivos de 32 bits, respecto de las soluciones de drivers basados en MS-DOS incluyen:

- Dramática mejora del rendimiento y software de almacenamiento en caché de disco en modo real
- Uso de la memoria no convencional (se ha sustituido el SMARTDrive en modo real)
- Mejora en la multitarea cuando se accede a información en el disco
- Soporte de caché dinámica

Tanto MS-DOS como Windows 3.1 utilizan código de 16 bits en modo real para manipular la tabla de asignación de archivos (FAT), y para leer y escribir desde el disco. Ser capaz de manipular el sistema de archivos del disco en modo protegido elimina o reduce la necesidad de cambiar al modo real para escribir información en el disco a través de MS-DOS, esto resulta en un aumento del rendimiento de acceso de E/S a archivo.

El VFAT de 32 bits funciona con un driver de caché en modo protegido de 32 bits (VCACHE). Este driver reemplaza el software de caché de disco de 16 bits en modo real proporcionado con MS-DOS y Windows 3.1. El driver de VCACHE ofrece mejores algoritmos de caché que SMARTDrive, para cachear la información leída desde, o escrita en la unidad de disco. Este driver también gestiona el banco de caché para el sistema de archivos de CD-ROM (CDFS) y los redirectores de red de 32 bits provistos con Windows 95.

Otra gran mejora de VCACHE sobre SMARTDrive es que el banco de memoria utilizado para la caché es dinámico y se basa en la cantidad de memoria libre del sistema. Los usuarios ya no tienen que asignar un bloque de memoria como una caché de disco. El sistema automáticamente asigna o des asigna memoria utilizada para la caché, basándose en el uso del sistema.

Por ejemplo, a medida que realiza un gran número de actividades en la red, Windows 95 aumenta el tamaño de la caché. Cuando la actividad de la red disminuye y más aplicaciones se inician, Windows 95 disminuye el tamaño de la caché.

Sistema de Archivos de CD-ROM

El CDFS de 32 bits en modo protegido implementado en Windows 95 mejora el acceso y performance del CD-ROM respecto del driver MSCDEX en modo real de Windows 3.1 (CDFS se ajusta a la norma ISO 9660). El caché del driver del CDFS también es dinámico y no requiere configuración o ubicación estática por parte del usuario. Para más información consulte el capítulo 17 "Ajuste de rendimiento"

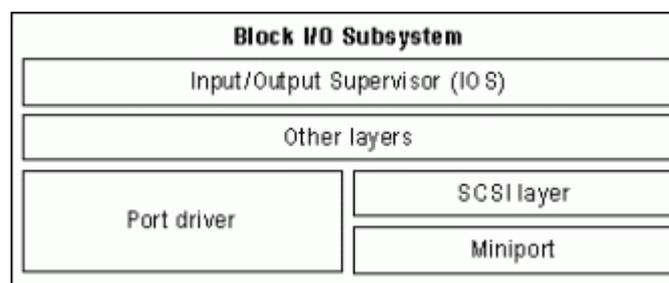
Las ventajas del nuevo controlador CDFS 32 bits incluyen:

- Uso de memoria no convencional (se ha sustituido MSCDEX en modo real)
- Mejora el rendimiento respecto a MSCDEX basado en MS-DOS y la cache en modo real
- Mejora la multitarea cuando se accede a la información del CD-ROM
- Soporta caché dinámica para proporcionar un mejor equilibrio entre el suministro de memoria para ejecutar aplicaciones y la asignación memoria para servir de caché de disco

MSCDEX ya no es necesario en Windows 95, y es es removido de la memoria y del autoexec.bat, utilizándose en su lugar el driver CDFS de 32 bits.

Subsistema de E/S en Bloque

El subsistema de E/S en bloque de Windows 95 mejora el acceso a la arquitectura del dispositivo de disco "FastDisk" de 32 bits utilizado en Windows 3.1 para mejorar el rendimiento de todo el sistema de archivos y proporciona una gama más amplia de soporte de dispositivos.



Como puede apreciarse en el diagrama anterior, el subsistema de E/S en bloque posee los siguientes componentes

Supervisor de E/S (IOS)

Este componente proporciona servicios a los sistemas de archivos y drivers. El IOS es responsable de poner en cola las solicitudes de servicio de archivos y de rutear las peticiones al controlador de sistema de archivo adecuado. El Supervisor de E/S también proporciona notificación asíncrona de los eventos del sistema de archivos a los drivers. Este componente se describe con más detalle en la sección siguiente.

Driver de Puerto

Es un driver monolítico de 32 bits en modo protegido que se comunica con un dispositivo de disco específico, como un controlador de disco duro. Este driver es específico de Windows 95 y es parecido al driver de acceso a disco de 32 bits que se utiliza en Windows 3.1 (por ejemplo, WDCTRL para controladores de disco duro compatibles con Western Digital™). En Windows 95, el driver que se comunica con controladores de disco duro IDE o ESDI y los controladores de disquete se implementa como un controlador de puerto. Un controlador de puerto proporciona la misma funcionalidad que la combinación del administrador SCSI y el controlador de minipuerto.

Capa SCSI

Este componente aplica un modelo de arquitectura de controlador universal de 32 bits, de modo protegido, para la comunicación con los dispositivos SCSI. La capa SCSI proporciona toda la funcionalidad de alto nivel que es común a SCSI y dispositivos similares, y luego utiliza un controlador de minipuerto para manejar las llamadas de dispositivos específicos E/S. El Administrador de SCSI es también parte de este sistema y proporciona el soporte de compatibilidad para el uso de drivers de minipuerto de Windows NT.

Driver de Minipuerto

El modelo de controlador de minipuerto de Windows 95 hace que sea más fácil para un proveedor de dispositivos de disco escribir un controlador de dispositivo. Debido a que el stub de SCSI proporciona la funcionalidad de alto nivel para comunicarse con los dispositivos SCSI, el proveedor de dispositivos de disco de hardware sólo tiene que crear un driver de minipuerto adaptado a su propio dispositivo de disco. El driver de minipuerto de Windows 95 es de código de 32 bits en modo protegido, y compatible con los controladores de minipuerto de Windows NT. Sin embargo, los antiguos drivers de minipuerto escritos para Windows NT no incluyen información Plug and Play y, por lo tanto, no van a funcionar bien en Windows 95.

Además de estas capas, el subsistema de E/S en bloque proporciona una capa de asignación en modo real. Esta capa proporciona compatibilidad con los con los drivers de dispositivos en modo real, basados en MS-DOS para los cuales no existe una contraparte en modo protegido. También permite que el sistema de archivos en modo protegido se comunique con un controlador de modo real como si se tratara de un componente en modo protegido.

Las capas que están por encima de la E/S en bloque y el mapeador de modo real, son código en modo protegido; el mapeador de modo real traduce el requerimiento de E/S de archivo desde el modo protegido, al modo real, de manera que el driver de dispositivo basado en MS-DOS puede realizar la operación de escribir o leer desde o hacia el dispositivo de disco. Por ejemplo. El mapeador de modo real es usado cuando el software de modo real de compresión de disco esta corriendo y un driver de compresión de disco en modo protegido no está disponible.

Nota: El uso de controladores de dispositivos basados en MS-DOS pueden ser un cuello de botella potencial, porque todas las E/S deben ser serializadas. También, debido a que todos estos VxD deben bloquear las páginas de memoria, esto aumenta el trabajo del sistema operativo.

Supervisor y Cargador de Driver de E/S

El supervisor de E/S es un VxD del sistema necesario que lleva a cabo todo el control y gestión de tareas del sistema de archivo en modo protegido, y los dispositivos de bloque en Windows 95. Además carga e inicializa los drivers de dispositivos en modo protegido y provee los servicios necesarios para las operaciones de E/S

El supervisor de E/S recibe el requerimiento desde un file system VFAT y CDFS, y carga los drivers para el acceso a los drivers y dispositivos locales de disco, incluyendo SCSI e IDE. Soporta drivers compatibles con WD1003, toma el control de los drivers de modo real y provee un mapeador para dichos drivers.

El mapeador de modo real en el supervisor de E/S provee compatibilidad con los drivers de dispositivos en modo real de MS-DOS, para los cuales no existen equivalentes en modo protegido. Por ejemplo, el mapeador de modo real va a trabajar cuando el software en modo real de compresión de disco esta corriendo, y un driver de compresión de disco en modo protegido no está disponible. Este componente asegura la compatibilidad entre los dispositivos de disco basados en MS-DOS y los drivers de dispositivo de Windows 95

El supervisor de E/S fue implementado primero en Windows 3.x como *BLOCKDEV, y Windows 95 también proporciona servicios *BLOCKDEV para antiguos drivers de acceso a disco de 32 bits. Las nuevas responsabilidades para el Supervisor de E/S incluyen:

- Registrar drivers
- Enrutar y enconlar los requerimientos de E/S, y enviar notificaciones asincrónicas a los drivers que lo necesiten
- Proveer servicios a esos drivers para que puedan utilizar memoria y completar sus requerimientos de E/S

Windows 95 carga e inicializa el Supervisor de E/S como especificado en device=entry en System.ini. El Supervisor de E/S se inicializa antes que los clientes y controladores de dispositivos virtuales como APIX y *INT13, para que los clientes y drivers de dispositivos virtuales puedan llamar a los servicios en el Supervisor de E/S para registrar y llevar a cabo las tareas.

Para cargar e inicializar los controladores de puerto, de minipuerto y los drivers de valores agregados, el Supervisor de E/S requiere los archivos de estos drivers se almacenen en el directorio SYSTEMIOSUBSYS con las siguientes extensiones de nombre de archivo.

Extensión de nombre de archivo	descripción
PDR	Driver de puertos, como SCSIPOINT, ESDI_506, y NEC
MPD	Drivers de minipuertos
386 O VXD	Drivers de valores agregados, como seguimiento de volumen y drivers suministrados por el proveedor.

El directorio SYSTEMIOSUBSYS está reservado para los drivers de dispositivos diseñados específicamente para usarse con el Supervisor de E/S. Otros clientes o drivers de dispositivos virtuales deben ser almacenados en otros directorios y cargados explícitamente usando devices=entry en SYSTEM.INI

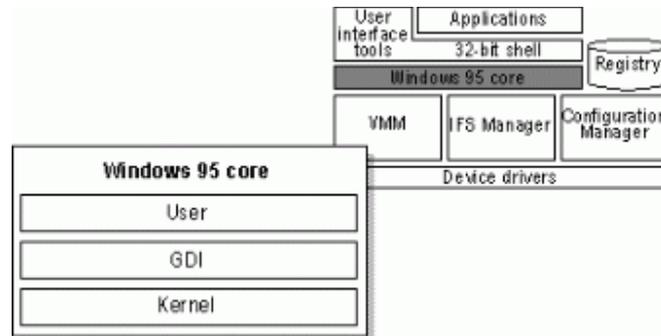
El Supervisor de E/S inicializa los drivers de los dispositivos (como se explicará en las siguientes secciones) desde la capa más baja hacia arriba, entonces los drivers de puertos serán inicializados antes que los drivers de tipo específicos, y así.

Los drivers de Valor Agregado, son inicializados en grupos, capa por capa. Cuando todos los drivers de una capa son inicializados, se continúa con la siguiente. El orden de inicialización dentro de una capa no está definido, por lo que no se puede depender de que los drivers en un grupo sean inicializados en un orden específico, o que el orden se mantenga entre operaciones de inicio.

Para la detección Plug and Play, el supervisor de E/S carga un driver de puerto o minipuerto específico sólo si el Administrador de Configuración requiere que ese driver sea cargado después de la detección de hardware, localiza un adaptador.

Componentes de Núcleo del Sistema

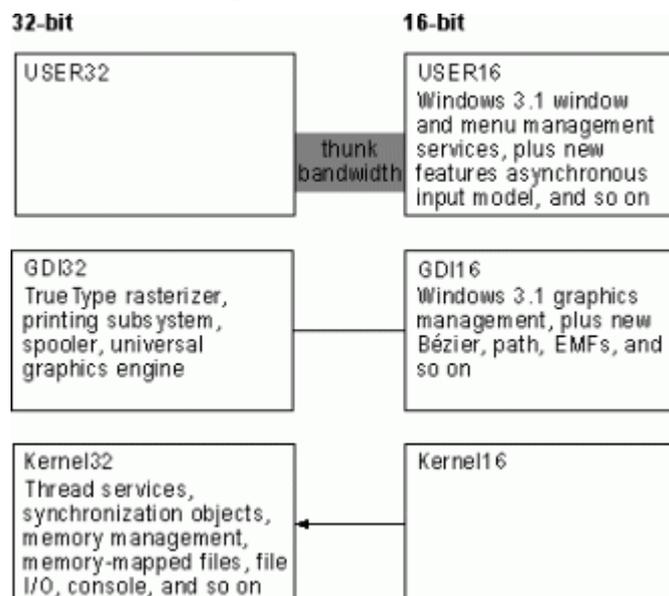
Al igual que las versiones de Windows 3.1 y Windows para trabajo en Grupo 3.1, Windows 95 incluye un núcleo compuesto por tres componentes: User, Kernel e Interfaz de Dispositivo Gráfica (GDI)



Cada uno de estos componentes incluye un par de DLLs (una de 32 bits y otra de 16 bits) las cuales proveen servicios para las aplicaciones que corran. Windows 95 fue diseñado para usar código de 32 bits, mejorando significativamente su rendimiento sin sacrificar la compatibilidad de aplicaciones. También mantiene el código existente de 16 bits donde sea requerido, para mantener la compatibilidad, o donde el código de 32 bits pueda llegar a incrementar los requerimientos de memoria sin mejorar significativamente la performance.

Todos los subsistemas de E/S (como por ejemplo sistemas de red y archivos) y los drivers dispositivos son de 32 bits, al igual que toda la gestión de memoria y componentes de programación, incluyendo el Kernel y el Administrador de memoria virtual.

Como se muestra en la siguiente ilustración, los servicios de más bajo nivel que ofrece el núcleo de Windows 95 se implementan como código de 32 bits para asegurar un núcleo de alto rendimiento. La mayor parte del código de 16 bits restante consiste en lenguaje ensamblador a mano alzada, que ofrece un rendimiento que compite con algo de código de 32 bits usado por otros sistemas operativos.



Muchas de las funciones provistas por el GDI - que son mayormente funciones complejas y CPU-intensivas- ahora están reescritas en código de 32 bits para mejorar el rendimiento. La mayor parte del código de gestión de ventanas en los componentes de User -que son funciones pequeñas y rápidas- siguen siendo de 16 bits, por lo que conserva la compatibilidad de aplicaciones.

Windows 95 también implementa mejoras en el entorno de MS-DOS y Windows 3.1 implementando muchos drivers de dispositivos como código en modo protegido de 32 bits

Los drivers de dispositivos virtuales en Windows 95 asumen la funcionalidad provista por muchos drivers de dispositivos de modo real basados en MS-DOS. Esto resulta en una huella mínima de memoria convencional, mejora en el rendimiento y mayor fiabilidad y estabilidad del sistema a través de los controladores de dispositivo en modo real.

Las siguientes secciones describen los servicios prestados por estos componentes del núcleo.

User

El componente USER controla la entrada desde teclado, mouse y otros dispositivos de entrada a la interfaz de usuario (ventanas, iconos, menús, etc). También gestiona la interacción con el driver de sonido, el timer y los puertos de comunicación

Windows 95 usa un modelo de entrada asíncronico para todas las entradas al sistema y a las aplicaciones. Como varios dispositivos de entrada generan interrupciones, el manejador de interrupciones convierte esas interrupciones en mensajes y los envía al área del thread de entradas en crudo, la cual por turnos pasa cada mensaje a la cola de mensajes correcta. Si bien cada thread basado en Win32 puede tener su propia cola de mensajes, todas las aplicaciones basadas en Win16 comparten una cola de mensaje común.

Kernel

El kernel proporciona la funcionalidad básica del sistema operativo, incluyendo el servicio de E/S de archivos, manejo de memoria virtual, y planificación de tareas. Cuando un usuario quiere iniciar una aplicación, el kernel carga los archivos EXE y DLL para la aplicación.

El manejador de excepciones es otro servicio del kernel. Las excepciones son eventos que ocurren mientras un programa está corriendo y requiere software fuera de la corriente normal de control para mantenerse en marcha. Por ejemplo, si una aplicación genera una excepción, el kernel está capacitado para comunicar esta excepción a la aplicación y realizar las funciones necesarias para resolver el problema.

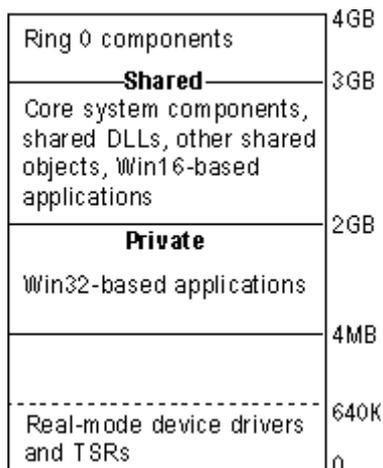
El kernel también asigna memoria virtual, resuelve referencias de importación, y soporta paginación por demanda para las aplicaciones. Cuando una aplicación se ejecuta, el kernel planifica y ejecuta threads para cada proceso de la aplicación.

También el kernel provee servicios para aplicaciones de 16 y de 32 bits, usando un proceso de traducción llamado thunking para mapear entre formatos de 16 y 32 bits. El thunking convierte un valor de 16 bits en su equivalente de 32 bits

Gestión de Memoria Virtual

La memoria virtual es un término que se refiere al hecho de que el sistema operativo puede asignar mas memoria de la que posee físicamente la computadora. Cada proceso es alocado en un espacio de dirección virtual único, el cual es un conjunto de direcciones disponibles para su uso por los threads del proceso. Este espacio de direcciones virtuales parece ser de 4 GB de tamaño – 2GB reservados para el almacenamiento del programa, y 2GB reservados para el almacenamiento del sistema.

El siguiente diagrama ilustra que los componentes del sistema y aplicaciones de Windows 95 residen en memoria virtual.



De arriba a abajo, los componentes de sistemas y de aplicaciones residen en memoria virtual de la siguiente manera:

Todos los componentes Ring 0 residen en el espacio de direcciones superiores a los 3GB

Los componentes del núcleo de Windows 95 y las DLLs compartidas residen en el espacio de direcciones compartidas, entre los 2, y los 3 GB. Esto hace que estén disponibles para todas las aplicaciones.

Las aplicaciones basadas en Win32 residen entre los 4MB y los 2GB. Cada aplicación tendrá su propio espacio de direcciones, lo que significa que otros programas no podrán dañar, o perturbar de otro modo a la aplicación y viceversa.

Archivo Swap de Windows 95

El archivo swap para la implementación de memoria virtual en Windows 95, trata los problemas y limitaciones impuestas por Windows 3.1

Bajo Windows 3.1 el usuario tenía que decidir entre usar un archivo swap temporal, o uno permanente, cuanta memoria asignarle a este archivo y si usar acceso a disco de 32 bits, para acceder al archivo swap. Los usuarios se beneficiaban con un archivo swap temporal, ya que este archivo no necesitaba ser continuo, Windows asignaba espacio en el disco duro cuando el usuario iniciaba el SO y liberaba el espacio cuando se salía del mismo. Un archivo swap permanente provee mejor performance, pero requiere bloques contiguos de espacio en el disco rígido a nivel físico, y era estático, por lo que no liberaba espacio cuando el usuario salía de Windows.

La implementación del archivo swap en Windows 95 simplifica la tarea de configuración para el usuario y combina lo mejor de los archivos temporales y permanentes de swap, mejorando los algoritmos y métodos de acceso a memoria virtual. El archivo swap en Windows 95 es dinámico; puede agrandarse o achicarse basado en las operaciones que se realizan en el sistema. Además puede ocupar una región fragmentada del disco rígido sin disminuir el rendimiento. El archivo swap también puede residir en un volumen comprimido

Usted puede ajustar los parámetros para definir el archivo swap en Windows 95, sin embargo la necesidad de hacer esto se reduce por el uso de los valores por defecto del sistema. Para mas información acerca de las opciones de configuración del archivo swap, vea el Capítulo 17 "Ajustes de rendimiento"

Interfaz de Dispositivo Gráfico

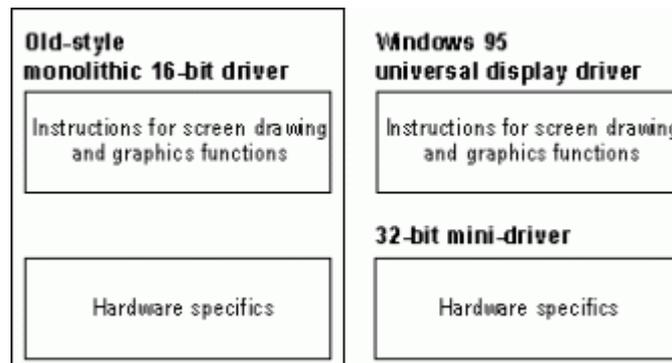
La interfaz de dispositivo gráfico (GDI) es el sistema gráfico que controla que aparece en la pantalla. También provee soporte gráfico para impresoras y otros dispositivos de salida. Realiza gráficas primitivas, manipula mapas de bits e interactúa con los drivers de dispositivos gráficos independientes, incluyendo los de impresoras y otros dispositivos de salida.

Subsistema de Gráficos

Éste subsistema proporciona soporte de gráficos para dispositivos de entrada y de salida. Para ganar confiabilidad y mejorar el rendimiento, Microsoft desarrolló un nuevo motor de gráficos de 32 bits (también conocido como motor de Dispositivo Independiente de Mapas de Bits o motor DIB). Éste motor controla directamente la salida de gráficos a la pantalla, logrando más confiabilidad y mejor rendimiento del sistema. También provee un conjunto de funciones de dibujo genéricas optimizadas para dispositivos gráficos monocromo, color de 16 bits, color de 16 bits de alta calidad, 256 colores, y color verdadero de 24 bits, y soporte para Bézier.

El subsistema de gráficos soporta Image Color Matching, para mejorar la coincidencia entre el color de la pantalla y el de los dispositivos de salida. Con Image Color Matching en Windows 95, los usuarios podrán ver una mejor coincidencia entre los colores mostrados en pantalla y el color impreso en un dispositivo de impresión a color. Image Color Matching se implementa en Windows 95 como un servicio. Las aplicaciones pueden utilizar las APIs y DLL que correspondan para sacar provecho de las características de ICM.

Al igual que con otras partes del sistema operativo, el subsistema de gráficos de Windows 95 contiene un modelo de driver/minidriver universal. Como se ilustra en la siguiente gráfica, los drivers de visualización para Windows 3.1 incluían instrucciones específicas del hardware, además de las instrucciones generales para el sistema operativo. Ahora, todas las instrucciones para dibujar en pantalla o en dispositivos de salida, están incluidas en el driver universal de visualización. Los minidrivers de Windows 95 definen solamente las instrucciones específicas del hardware. Éstos están disponibles para la mayoría de los adaptadores VGA y aceleradoras gráficas, incluyendo S3, ATI, Tseng, Paradise, Western Digital y Cirrus Logic.



Consejo: Puede haber problemas cuando Windows 95 asuma que el adaptador de pantalla puede soportar cierta funcionalidad que, de hecho, no pueda. Si esto ocurre, los efectos secundarios pueden ser, desde pequeñas irregularidades, hasta el bloqueo del sistema.

Para determinar si los problemas pueden estar relacionados con el adaptador de pantalla, puede ir deshabilitando progresivamente las funcionalidades de visualización, usando la opción del sistema en el Panel de Control. Para más información, vea el Capítulo 17 “Ajuste de rendimiento”

Subsistema de Impresión

El subsistema de impresión de 32 bits de Windows 95 mejora el rendimiento a través de la impresión en segundo plano, y un tiempo de retorno a la aplicación más rápido. El “spooler” de Windows transmite los datos a la impresora utilizando procesamiento de threads en segundo plano.

Windows emplea spools con archivos de metadatos mejorados (EMF), en lugar de datos de impresión en bruto (raw), para asegurar un tiempo de retorno bajo (hasta dos veces más rápido que en Windows 3.1). Cuando el archivo EMF llega al spooler, la información es interpretada en segundo plano y la salida enviada a la impresora.

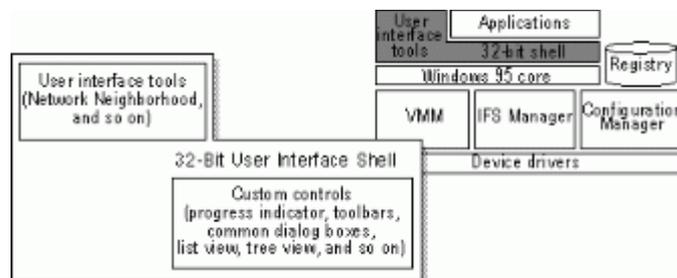
El subsistema de impresión soporta Point and Print. Cuando el usuario navega por la red para elegir la impresora a utilizar, Windows 95 instala automáticamente el driver apropiado desde el servidor de Windows 95 o Windows NT.

Otra característica del subsistema de impresión de Windows 95 es la impresión diferida. Si no hay un dispositivo de impresión disponible, el usuario aún puede “imprimir” el trabajo. Windows 95 genera el trabajo de impresión, y lo guarda para imprimirlo en cuanto un dispositivo de impresión esté disponible. Entonces, cuando el usuario conecte su computadora portátil al regresar a la oficina, el trabajo de impresión que fue “impreso” comenzará a la generación de páginas en el dispositivo de impresión.

El subsistema de impresión también provee soporte a nivel de sistema de protocolos de comunicación bidireccional para impresoras que adhieran al “Puerto de Comunicación Extendido” (Extended Communication Port – ECP), estándar de comunicación con impresoras, desarrollado por Microsoft y Hewlett-Packard. Esto permite a las impresoras enviar mensajes no solicitados a Windows 95 y a las aplicaciones. Por ejemplo, la impresora puede enviar un mensaje de “impresora sin papel” o “impresora fuera de línea”. La comunicación bidireccional permite un reporte de estado mucho más detallado, reportando mas variedad de información, tales como condición del tóner, detalles acerca del atasco de papel, etc.

Interfaz de Usuario

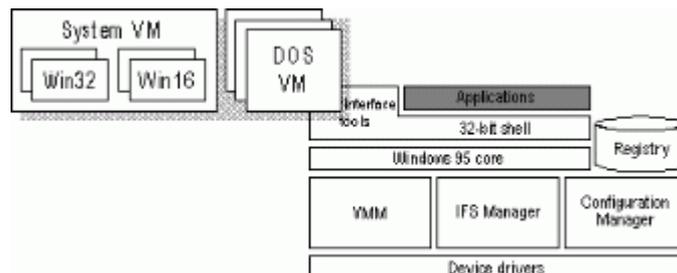
Windows 95 incluye una interfaz de usuario de 32 bits, basado en el Explorador de Windows. Este shell contiene varias herramientas de escritorio, incluyendo el entorno de red. Como se muestra en la siguiente figura, estas herramientas se ejecutan al mismo nivel que otras aplicaciones basadas en Win32, Win16 y MS-DOS



Todas las aplicaciones y herramientas pueden aprovechar los controles comunes que ofrece la shell, tales como cuadros de diálogo comunes, vistas de árbol y vistas de lista

Soporte de Aplicaciones

Windows 95 soporta aplicaciones basadas en Win32, Win16 y MS-DOS



Tal como se aprecia en la figura, las aplicaciones basadas en Win32 y Win16 corren en la System VM (Máquina virtual del sistema) Cada aplicación basada en Win32 se ejecuta en un espacio de direcciones separado, mientras que las aplicaciones basadas en Win16 se ejecutan juntas en un espacio de memoria compartido. Cada aplicación basada en MS-DOS corre en su propia Máquina Virtual.

Para detalles acerca del uso en Windows 95 de aplicaciones basadas en Win32, Win16 y MS-DOS vea el capítulo 22 “Soporte de Aplicaciones” para más información acerca de las VMs, vea “Gestor de Máquina Virtual” desarrollado anteriormente en este capítulo.