

“Las computadoras personales proporcionaron al mundo una forma totalmente nueva de trabajar, jugar y comunicarse.”

Bill Gates

Objetivos de aprendizaje

Después de completar este capítulo, el lector debe ser capaz de describir:

- Cómo acceder a emuladores del MS-DOS desde otros sistemas operativos.
 - La forma en que MS-DOS proporcionó una base para las primeras versiones de Microsoft Windows.
 - Los fundamentos de los sistemas controlados por comandos y cómo construir archivos por lotes simples.
 - Cómo varios procesos pueden compartir un procesador.
 - Las limitaciones del MS-DOS.
-

MS-DOS, también conocido simplemente como DOS, fue desarrollado para su ejecución en computadoras de escritorio de usuario único e independiente. Cuando el mercado de computadoras personales explotó en la década de 1980, MS-DOS era el sistema operativo normal entregado con millones de estas máquinas.

Este sistema operativo es uno de los más sencillos de comprender. En muchas formas, MS-DOS ejemplifica los primeros sistemas operativos porque administra trabajos en forma secuencial desde un usuario único. Sus ventajas son su operación fundamental y sus comandos de usuario directos. Con sólo unas cuantas horas de instrucción, un usuario por primera vez puede aprender a manipular exitosamente una computadora personal y sus archivos y dispositivos.


MS-DOS Fue escrito para una sola familia de microprocesadores: los chips Intel 8086, 8088, 80186 y 80286. Cuando estos microprocesadores dominaron el mercado de las computadoras personales, también lo hizo MS-DOS. Cuando aparecieron nuevos chips, MS-DOS perdió su ventaja en el mercado sobre otros sistemas operativos más sofisticados. MS-DOS fue el primer sistema operativo para una generación de microcomputadoras, por lo que en este texto se incluye como un ejemplo.

Merece la pena observar que las primeras versiones de Windows (versiones 1.0 a 3.1) eran meras interfaces gráficas de usuario que se ejecutaban en la parte superior del sistema operativo MS-DOS. Consulte el capítulo 15 sobre Windows para más información.

Historia

MS-DOS fue el sucesor del sistema operativo CP/M. CP/M (de Control Program for Microcomputers) se ejecutaba en las primeras computadoras personales, máquinas de 8 bits comercializadas por Apple Computer y Tandy Corporation. Pero cuando las primeras computadoras personales de 16 bits fueron desarrolladas en 1980, requerían un sistema operativo con más capacidad que CP/M, y muchas compañías se apresuraron a desarrollar el sistema operativo que se volvería la norma para la nueva generación de hardware.

IBM fue el catalizador. Cuando investigaba un sistema operativo para su próxima versión a entregar de las computadoras personales de 16 bits, Digital Research ofreció el nuevo sistema operativo CP/M-86 y Softech ofreció su sistema P. IBM analizó con cuidado ambos sistemas y comenzó a negociar con Digital Research para adquirir el “nuevo y mejorado sistema CP/M”. Mientras tanto, Microsoft, una naciente compañía iniciada por Bill Gates y Paul Allen, descubrió un sistema operativo innovador, denominado 86-DOS, diseñado por Tim Patterson de Seattle Computer Products para ejecutar esa línea de productos de computadoras personales de 16 bits. Microsoft lo compró, lo denominó MS-DOS que significa *Microsoft Disk Operating System*, y lo puso a disposición para IBM.



No debe confundirse MS-DOS con software que tiene acrónimos semejantes, como “distributed operating systems” o un “disk operating system” comercializado por otros fabricantes como IBM.


IBM escogió MS-DOS en 1981, lo denominó PC-DOS y lo proclamó la norma para su línea de computadoras personales. Finalmente, con el peso del endoso de IBM, MS-DOS se volvió el sistema operativo normal para la mayor parte de las computadoras personales en la década de 1980.

Este sistema operativo experimentó varias modificaciones después de su aparición en Seattle. Unas fueron necesarias para reparar algunas deficiencias, mientras otras se realizaron para permitir cambios fundamentales en el hardware, como capacidades aumentadas de la unidad de disco o formatos diferentes. En la tabla 14.1 se muestran algunas de las versiones más importantes.

(Tabla 14.1)

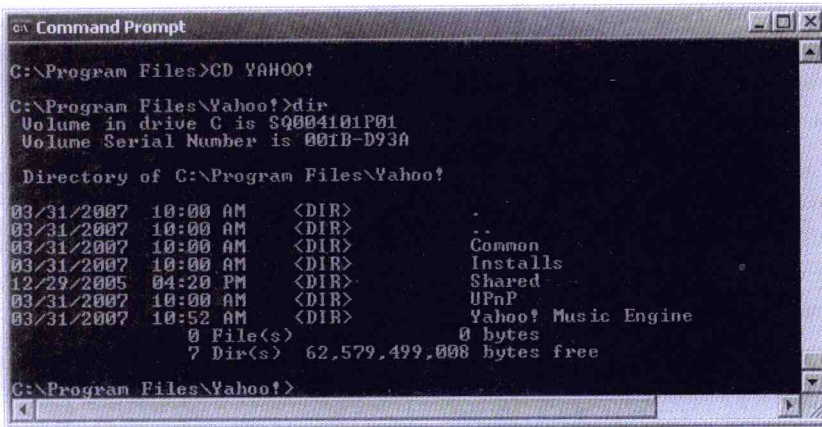
Evolución de MS-DOS.

Versión no.	Fecha de lanzamiento	Características
1.0	1981	Compatible con CP/M, sólo atendía un directorio
1.1	1982	Permitía discos de 5¼ pulg con dos lados
2.0	1983	Eliminó algunos defectos de la primera versión
3.0	1984	Aumentó el requerimiento de memoria a 36K, atendía PC/AT
3.1	1984	Primera versión que atendía redes
3.2	1986	Atendía token ring y discos de 3½ pulg
3.3	1987	Atendía la computadora IBM PS/2
4.0	1988	Atendía grandes discos de más de 32 MB
5.0	1991	Mejor uso de la memoria extendida
6.0	1993	Mejor uso de la memoria convencional
6.22	1994	Proporciona a los usuarios capacidades previamente disponibles sólo como aplicaciones de un tercer partido

 Aunque los emuladores de MS-DOS encontraron que Windows permite nombres de archivo superiores de 8 caracteres en los comandos auténticos de MS-DOS no.

Cada versión de MS-DOS era normal, de modo que versiones posteriores de éste fueron compatibles con las primeras. En consecuencia, los programas escritos para su ejecución en la versión 5.0 también podían ejecutarse en la versión 6.2. También significaba que, entre diferentes fabricantes, con los mismos comandos se obtenía la misma respuesta del sistema operativo sin importar quién había fabricado el hardware para ejecutarlo; esta característica era importante en esa época.

Aunque MS-DOS ya no se usa ampliamente, muchos sistemas operativos Windows ofrecen un emulador DOS para permitir que cualquier persona introduzca algunos comandos semejantes a los de DOS. Para encontrarlos en Windows, haga clic en el botón de inicio, luego todos los programas, accesorios, símbolos del sistema para ver una pantalla como la que se muestra en la figura 14.1.

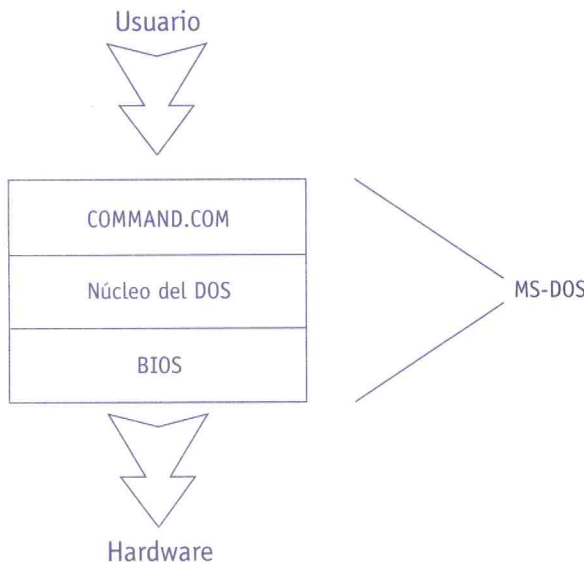


(Figura 14.1)

Ventana de símbolos del sistema abierta desde un sistema operativo Windows. A partir de los símbolos del sistema, los usuarios pueden introducir comandos MS-DOS.

Objetivos de diseño

MS-DOS fue diseñado para atender a un usuario novato único en un entorno de proceso único, como se muestra en la figura 14.2. Su soporte normal de E/S incluye teclado, monitor, impresora y unidad de almacenamiento secundario. Sus comandos de usuario están basados en palabras y frases del inglés y son indicativos de la acción a realizar. Estos comandos son interpretados por el procesador de comandos, en general la única porción del sistema operativo con la que interactúa la mayoría de los usuarios.



(Figura 14.2)

Las tres capas del MS-DOS. La capa superior es el procesador de comandos, un programa denominado COMMAND.COM, que proporciona independencia de dispositivo. El núcleo de software de DOS proporciona servicios de administración de archivos. El software BIOS proporciona servicios de administración de dispositivos.

El enfoque por capas es fundamental para el diseño de todo el sistema MS-DOS, que es “proteger” al usuario de tener que trabajar con los bits y los bytes de la máquina desnuda

que integran la capa inferior; el hardware que incluye los circuitos eléctricos, los registros y otros componentes básicos de la computadora. Cada capa está construida sobre la capa precedente, empezando de abajo arriba.

La capa en la parte inferior del MS-DOS es el **BIOS** (Basic Input/Output System). Esta capa del sistema operativo se interconecta directamente con los diversos dispositivos de E/S como impresoras, teclados y monitores. BIOS contiene los **controladores de dispositivos** que controlan el flujo de datos hacia y desde cada dispositivo excepto las unidades de disco. Recibe información de estado sobre el éxito o fracaso de cada operación de E/S y la pasa al procesador. BIOS se ocupa de las pequeñas diferencias entre unidades de E/S, de modo que el usuario puede comprar una impresora de cualquier fabricante sin tener que escribir un controlador de dispositivo para ella: BIOS hace que trabaje como debe hacerlo.

La capa de en medio, el núcleo del DOS, contiene las rutinas necesarias para conectarse con las unidades de disco. Se lee en la memoria en el momento de inicio desde el archivo MSDOS.SYS que reside en el disco de arranque. El núcleo del DOS es un programa registrado proporcionado por Microsoft Corporation que implementa MS-DOS. Al núcleo tienen acceso los programas de aplicación y el núcleo proporciona una colección de servicios independientes del hardware, como administración de la memoria y administración de archivos y registros. Éstas se denominan funciones del sistema. Como BIOS, el núcleo DOS compensa diferencias de un fabricante a otro, de modo que todas las unidades de disco funcionan del mismo modo. En otras palabras, el núcleo hace que la administración de archivos sea transparente para el usuario, de modo que no es necesario recordar en qué pistas y sectores están almacenados sus archivos; y qué sectores del disco están dañados y deben evitarse. El núcleo hace esto por el usuario; administra el almacenamiento y la recuperación de archivos y asigna y libera dinámicamente el almacenamiento secundario según sea necesario.

La tercera capa, el procesador de comandos, algunas veces se denomina shell. Esta es la parte del sistema que envía desplegados al usuario, acepta los comandos escritos, ejecuta los comandos y emite las respuestas idóneas. El procesador de comandos reside en un archivo denominado COMMAND.COM, que consta de dos partes almacenadas en dos secciones diferentes de la memoria principal. Algunos usuarios consideran de manera errónea que el archivo COMMAND.COM es todo el sistema operativo porque es la única parte que aparece en el directorio público. En realidad, es sólo uno de varios programas que integran el MS-DOS; los demás están ocultos.

El trabajo del procesador de comandos consiste en llevar a cabo los comandos del usuario introducidos desde el **signo del sistema** sin tener que esperar las instrucciones de algún controlador de dispositivos. Por ejemplo, cuando un usuario emite un comando PRINT, el procesador de comandos dirige el resultado hacia la línea de impresión mediante el BIOS. En forma semejante, cuando un usuario ordena ESCRIBIR un archivo, el procesador de comandos dirige el resultado hacia el monitor. En estos casos, el usuario no necesita compensar la baja velocidad de la impresora y la alta velocidad de la terminal; el usuario puede interactuar con dispositivos y con archivos de la misma manera.

La debilidad del procesador de comandos es que no es interpretativo. Es decir, los programadores no pueden tomar atajos para abreviar los comandos. Los nuevos usuarios deben aprender a introducir cada comando de manera completa y correcta. Es implacable para quienes no pueden escribir, deletrear o construir comandos perfectamente.

La versión 4.0 de MS-DOS introdujo un shell DOS controlado por menú para facilitar la interacción del usuario con el sistema, aunque no tuvo gran aceptación. Cuando se lanzó la versión 5.0, IBM y Microsoft también lanzaron un nuevo sistema operativo, OS/2, que fue diseñado para sustituir a MS-DOS. Aunque OS/2 tenía varias ventajas sobre MS-DOS, como usar toda la memoria disponible y atender multiprogramación, fracasó en generar el interés que esperaban ambas compañías.

En su apogeo, MS-DOS ejecutó una gran colección de paquetes de software. De las aplicaciones notables fueron Lotus 1-2-3 (un popular programa de cálculo en ese momento) y WordPerfect (un procesador de palabras). La adopción de la hoja de cálculo de estos dos productos ayudó a los usuarios novatos a apreciar la potencia de una computadora personal y fue un acicate para el crecimiento de la industria.

Microsoft ha continuado incorporando acceso a un emulador MS-DOS con un “símbolo del sistema DOS”, una capacidad semejante a un comando DOS, sobre muchos de sus productos Windows (mostrado en la figura 14.1) aún cuando MS-DOS ya ha sido retirado oficialmente del mercado.

Administración de la memoria

El administrador de la memoria tiene un trabajo relativamente simple porque administra un solo trabajo para un usuario único. Para ejecutar un segundo trabajo, el usuario debe cerrar o hacer una pausa en el primer archivo antes de abrir el segundo. El administrador de la memoria usa un **esquema de asignación de memoria del primer ajuste**. El primer ajuste fue escogido para las primeras versiones de DOS porque es la estrategia más eficiente en un entorno de usuario único. (Algunas versiones permiten características de memoria extendida y **multitarea**, características disponibles con hardware y software que puede agregarse, aunque para que este análisis sea breve no los incluiremos aquí.)

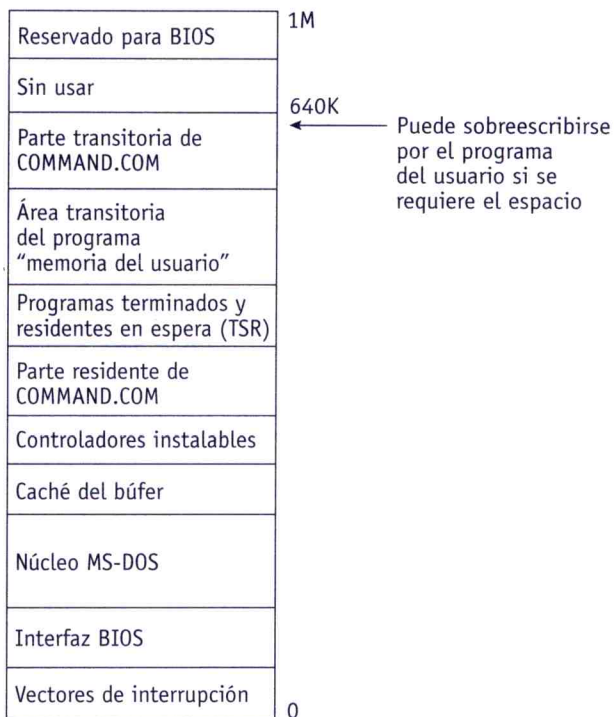
Antes de ver cómo se asigna la memoria, veamos cómo está estructurada. La memoria principal viene en dos formas: memoria de sólo lectura (ROM) y memoria de acceso aleatorio (RAM).

La ROM suele ser de tamaño muy pequeño y contiene un programa, una sección de BIOS, con la única tarea de iniciar el sistema. El proceso de inicialización es llamado rutina de arranque debido a que efectivamente el sistema sale adelante por sí mismo por medio de sus rutinas de arranque. Este programa en ROM inicia la computadora. También recupera el resto de la porción residente del sistema operativo de almacenamiento secundario y la carga en RAM.

RAM es la parte de la memoria principal donde se cargan y ejecutan los programas. La disposición de la RAM para una computadora con 1 MB de memoria se muestra en la figura 14.3.

(Figura 14.3)

Un megabyte de memoria principal RAM en MS-DOS. Los vectores de interrupción están ubicados en la memoria baja direccionable y el COMMAND.COM está superpuesto en la memoria alta direccionable.



La parte inferior de la RAM: denominada memoria baja direccionable porque es donde empieza la memoria: 0, 1, 2, . . ., está ocupada por 256 vectores de interrupción y sus tablas. Un vector de interrupción especifica dónde está ubicado el programa **manejador de interrupciones** para un tipo de interrupción específico. El uso de manipuladores de interrupciones se analizó en el capítulo 4. Lo anterior es seguido por tablas BIOS y tablas DOS, y por el núcleo DOS con controladores adicionales instalables, en caso de haber alguno, que se especifican en el archivo de configuración del sistema denominado CONFIG.SYS. Esto es seguido por la parte residente del intérprete de comandos COMMAND.COM; la sección necesaria para ejecutar programas de aplicación.

Cualquier programa de aplicación del usuario puede cargarse ahora en el área transitoria de programas (TPA). Si un gran programa requiere más espacio, el área superpuesta de COMMAND.COM, localizada en la ubicación de la memoria con numeración alta, también puede usarse por el programa de aplicación. Los programas COMMAND.COM en esta área se consideran transitorios porque se usan para ejecutar comandos, como FORMAT, que no puede ejecutar el usuario cuando está ejecutándose un programa de aplicación, de modo que es posible que otros programas se superpongan (o sobrescriban).

Asignación de la memoria principal

Las primeras versiones de MS-DOS proporcionaban toda la memoria disponible al programa de aplicación residente, pero esto demostró ser insuficiente porque el esquema simple de asignación de memoria contigua no permitía que los programas de aplicación asignaran y liberaran de manera dinámica bloques de memoria. Con la versión 2.0, MS-DOS comenzó a atender asignación y modificación dinámica, y a liberar de la memoria principal bloques por medio de programas de aplicación.

La cantidad de memoria que realmente pertenece a cada programa de aplicación depende tanto del tipo de archivo como del programa que se cargue y del tamaño del área transitoria de la TPA.

Toda la TPA se proporciona a los programas con la **extensión COM**, la necesiten o no.

A los programas con la extensión EXE sólo se les proporciona la cantidad de memoria que necesitan. Estos archivos tienen un encabezado que indica la cantidad mínima y máxima de memoria necesarias para la ejecución de un programa. Idealmente, MS-DOS proporciona al programa la cantidad máxima de memoria solicitada. En caso de no ser posible, trata de satisfacer el requerimiento mínimo. Si el mínimo es más que la cantidad de espacio de memoria disponible, entonces no es posible ejecutar el programa.

Excepto por archivos COM, en la TPA puede haber cualquier número de archivos a la vez. Pero esto plantea una pregunta interesante: ¿Por qué un sistema puede tener dos programas en la memoria cuando sólo puede ejecutar uno? Respuesta: al contar con varios archivos en la memoria a la vez, un usuario puede abrir rápidamente uno, trabajar con él y cerrarlo antes de abrir el siguiente. No es posible abrirlos al mismo tiempo, pero al abrirlos y cerrarlos de manera alterna, el usuario puede usar ambos programas rápida y fácilmente.

Por ejemplo, un programa procesador de palabras podría permitir que un usuario exhibiera dos archivos en la pantalla al mismo tiempo al abrir un área de trabajo por separado para cada uno. Estas dos áreas de trabajo dividen en secciones la pantalla; en este ejemplo una mostraría el archivo activo y la otra el archivo dormido. Si el usuario indica cuál trabajo debe empezar en el segundo archivo (dormido), entonces el primer archivo (activo) se cierra rápidamente y se activa el segundo archivo.

He aquí un segundo ejemplo: suponga que su programa principal del procesador de palabras incluye el código requerido para componer e imprimir un documento, pero si usted desea verificar la ortografía, es necesario cargar en el disco el programa de revisión ortográfica. Una vez que se ha cargado, la porción principal del procesador de palabras se mantiene en la memoria y el segundo programa se agrega sin borrar nada del primero que ya estaba ahí. Ahora usted tiene dos programas en la memoria, pero sólo uno se ejecuta en un momento dado. Este hecho se analizará en la sección de administración de procesos después en este capítulo.

Si un programa que ya está en ejecución requiere más memoria, como para *búfers* adicionales de E/S, el administrador de la memoria verifica si queda suficiente memoria. En caso afirmativo, la asigna al programa mientras actualiza la tabla de asignación de bloques de

memoria para ese programa. En caso negativo, entonces se regresa un mensaje de error al usuario y el programa se detiene. Aunque la asignación inicial de memoria es manipulada de manera automática por los programas escritos en BASIC, Pascal o en cualquier lenguaje atendido por MS-DOS, la reducción y ampliación de la asignación de la memoria durante la ejecución puede hacerse sólo desde programas escritos en lenguaje ensamblador o en C.

Asignación de bloques de memoria

El administrador de la memoria asigna la memoria usando un algoritmo del primer ajuste y una lista enlazada de bloques de memoria. Pero con la versión 3.3 y versiones posteriores, es posible escoger la estrategia del primero o del último ajuste. Cuando se usa el último ajuste, DOS asigna el bloque de memoria direccionable más alto suficientemente grande para satisfacer la solicitud del programa.

El tamaño de un bloque puede variar desde tan pequeño como de 16 bytes (denominado “párrafo”) hasta tan grande como la máxima memoria disponible. Cuando se forma un bloque, sus cinco primeros bytes contienen la información mostrada en la tabla 14.2.

(Tabla 14.2)

Los cinco primeros bytes de un bloque de memoria definen las características estructurales del bloque.

Byte	Contenido
Byte 0	ASCII 90h si es el último bloque, o ASCII 77H en caso de no serlo.
Bytes 1-2	Se incluye el número cero para indicar un bloque ocupado, y el apuntador hacia al prefijo del segmento del programa (PSP) que se crea durante la función EXEC cuando se carga el programa.
Bytes 3-4	Proporciona el número de párrafos contenido en el bloque.

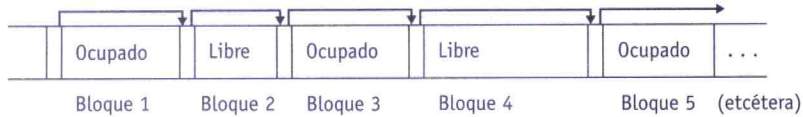
En consecuencia, si un bloque contiene cuatro párrafos y es el primer bloque, entonces su código sería 7700000004h (como se explica en la tabla 14.3) La letra h que aparece al final del valor anterior es una notación hexadecimal y no está registrado.

(Tabla 14.3)

Bloque de memoria muestra con los cinco primeros bytes que contienen 7700000004h.

Byte	Contenido	Significado
Byte 0	77	Indica que este no es el último bloque.
Bytes 1-2	0000	Indica que este es un bloque ocupado y que su apuntador hacia la PSP es cero.
Bytes 3-4	0004	Indica que este bloque contiene cuatro párrafos.

Siempre que llega una solicitud por memoria, DOS busca en la lista de bloques libre/ocupado, como se muestra en la figura 14.4, hasta que encuentra un bloque libre que se ajusta a la solicitud. Si la lista de bloques se desconecta, se genera un mensaje de error y el sistema se detiene. Para recuperar el sistema, es necesario reiniciarlo.



(Figura 14.4)

La lista enlazada de bloques de memoria.

Un programa de aplicación bien diseñado libera el bloque de memoria que ya no necesita. Si dos bloques de memoria libres son contiguos, de inmediato se fusionan en un bloque y se enlazan a la lista. Un programa que no está bien diseñado, sin embargo, amontona sus bloques de memoria hasta que termina su ejecución; sólo entonces MS-DOS puede liberar los bloques de memoria usados por ese programa.

Administración del procesador

El administrador del procesador tiene la tarea relativamente fácil de asignar el procesador al trabajo residente cuando está listo para su ejecución.

Administración de procesos

MS-DOS no fue escrito en código reentrante, analizado en la sección sobre memoria virtual en el capítulo 3, porque fue diseñado para un entorno unitarea de usuario único. El código reentrante es la base de la multitarea, y MS-DOS no lo atiende; en consecuencia, los programas no pueden irrumpir en medio de una rutina interna DOS y luego volver a empezar la rutina a partir de cualquier parte.

En nuestro ejemplo del procesador de procesamiento/verificación de ortografía, el programa padre del procesador llamaba al programa hijo verificador de ortografía. El padre pasaba a dormir y así permanecía mientras se ejecutaba el hijo. No hay intercalamiento, por lo que no hay necesidad de que algoritmos o políticas sofisticadas determinen el siguiente trabajo a ejecutar o durante cuánto tiempo. Cada trabajo se ejecuta en segmentos completos y no se interrumpe a la mitad. En otras palabras, no hay necesidad de mantener una buena mezcla de trabajos para equilibrar la utilización del sistema.

Sin embargo, aunque no es posible ejecutar dos trabajos al mismo tiempo, algunos programas de software dan esa ilusión. Tanto Microsoft Windows como *SideKick* de Borland, por ejemplo, parecen interrumpir el programa padre, cambiar lo que aparece en pantalla, ejecutar programas no relacionados y luego regresar al padre; pero esto no es multitarea. (Multitarea es el sinónimo de la industria de microcomputadoras para multiprogramación.) Estos programas se ven y sienten como operaciones multitarea porque retienen su área de la memoria y ejecutan programas ejecutables, pero no están en el estado de ejecución al mismo tiempo. En cada caso el programa padre duerme mientras el hijo es ejecutado en sí. Esta sincronización es posible porque los manejadores de interrupciones integrados en MS-DOS proporcionan a los programadores la capacidad de guardar toda la información sobre su programa padre, lo cual permite su reinicio idóneo una vez que termina el programa hijo.

Manejadores de interrupciones

Los manejadores de interrupciones son una parte crucial de los sistemas. Podría decirse que son responsables de la sincronización de procesos. Una computadora personal cuenta con 256 interrupciones y manejadores de interrupciones, a los que se tiene acceso por medio de la tabla de vector de interrupciones residente en los bytes inferiores de la memoria, como se muestra en la figura 14.3. Las interrupciones pueden dividirse en tres grupos: interrupciones internas de hardware, interrupciones externas de hardware e interrupciones de software. Las interrupciones internas de hardware son generadas por ciertos eventos que ocurren durante la ejecución de un programa, como división entre cero. La asignación de tales eventos a números específicos de interrupciones se conecta electrónicamente hacia el procesador y no es modificable por instrucciones de software.

Las interrupciones externas de hardware son ocasionadas por controladores de dispositivos periféricos o por coprocesadores como el 8087/80287. La asignación de dispositivos externos a niveles de interrupciones específicos la hace el fabricante del sistema de cómputo o el fabricante del dispositivo periférico. El software no puede modificar estas asignaciones porque están conectadas directamente; implementadas como conexiones eléctricas físicas.

Las interrupciones de software son generadas por el sistema y por programas de aplicación. Tienen acceso a funciones del DOS y del BIOS, las que, a su vez, tienen acceso a los recursos del sistema.

Algunas interrupciones de software se usan para activar programas de aplicación especializados que asumen el control de la computadora. *SideKick* de Borland es uno de estos programas. Este tipo de manejador de interrupciones se denomina *Terminate y Stay Resident* (TSR). Su función consiste en terminar un proceso sin liberar su memoria, proporcionando así facilidades de programación residentes en la memoria. El TSR suele ser usado por bibliotecas de subrutinas que son llamadas una vez desde el nivel de comandos MS-DOS y luego son puestas a disposición para proporcionar servicios a otras aplicaciones a través de una interrupción de software. Cuando un TSR empieza a funcionar, establece sus tablas de la memoria y se prepara para su ejecución al conectarse a una interrupción DOS; cuando todo está listo, el programa determina cuánta memoria debe mantener. Luego, cuando el programa sale, se hace pasar un código de regreso al padre.

¿Cómo se sincronizan estas interrupciones? Cuando la CPU detecta una interrupción, hace dos cosas: 1) coloca en una pila el contenido del PSW (programa de estado de palabra), el segmento del código de registro y el registro del apuntador de instrucciones; y 2) deshabilita el sistema de interrupciones de modo que todas las otras interrupciones se apaguen hasta que se haya resuelto la interrupción actual. La CPU usa el número de 8 bits colocado sobre el bus del sistema por el dispositivo de interrupciones a fin de obtener la dirección del manejador de interrupciones idóneo de la tabla del vector de interrupciones y obtiene ejecución en esa dirección.

Finalmente, el manejador de interrupciones reactiva la interrupción del sistema para permitir que interrupciones de alta prioridad ocurran, salvando cualquier registro que éste necesite usar y procesando la interrupción lo más rápido posible.

Resulta evidente que este procedimiento es delicado. La sincronización de las actividades del TSR con funciones del DOS ya en progreso debe diseñarse e implementarse con cuidado para evitar modificar cosas que no deben modificarse o la caída del sistema.

Administración de dispositivos

La capacidad de reordenar solicitudes para optimizar el tiempo de espera y el tiempo de búsqueda no es una característica del MS-DOS porque está diseñado para un entorno de usuario único. Todas las solicitudes se manejan sobre la base de primero en llegar, primero en servirse. Pero, desde la versión 3.0, BIOS puede atender *spooling* al permitir que los usuarios planifiquen la impresión de varios archivos uno después de otro. Para hacer esto, BIOS transfiere de manera continua datos desde un *búfer* específico de la memoria hacia la impresora hasta que se vacía el *búfer*.

MS-DOS fue escrito para sistemas simples que usan teclado, monitor, impresora, ratón, uno o dos puertos en serie y tal vez una segunda impresora. Para almacenamiento, la mayor parte de los sistemas de cómputo personales usan dispositivos de almacenamiento con acceso directo, por lo general disquetes o discos duros. Algunos sistemas también atienden un sistema de archivamiento de cinta magnética con acceso secuencial. El administrador de dispositivos del MS-DOS puede trabajar con cualquiera.

Estos sistemas sólo usan un tipo de dispositivo de E/S para cada puerto, de modo que los canales de dispositivos no forman parte del MS-DOS. Y debido a que cada dispositivo cuenta con su propia unidad de control dedicada, los dispositivos no requieren administración especial del sistema operativo. En consecuencia, los manejadores de dispositivos son los únicos dispositivos necesarios para que el administrador de dispositivos haga funcionar el sistema. Un controlador de dispositivos es un módulo de software que controla un dispositivo de E/S aunque maneja sus interrupciones. Cada dispositivo tiene su propio controlador de dispositivos. BIOS es la porción del administrador de dispositivos que manipula el software del controlador de dispositivos.

El BIOS está almacenado en ROM y en RAM. En muchos sistemas MS-DOS, las partes más primitivas de los controladores de dispositivos están ubicadas en ROM, de modo que pueden usar las aplicaciones y diagnósticos independientes, así como el programa *bootstrapping* del sistema. Una segunda sección se carga desde el disco hacia la RAM y extiende las capacidades de las funciones básicas almacenadas en ROM, de modo que BIOS puede manejar todas las solicitudes de entrada y salida al sistema.

Normalmente, BIOS es proporcionado por el fabricante del sistema apegado a las especificaciones de Microsoft para MS-DOS y, puesto que es el enlace entre el hardware y DOS, usa núcleos normales de sistema operativo sin importar el hardware. Esto significa que los programas con las interfaces normales DOS y BIOS para sus funciones dependientes del sistema pueden usarse en toda máquina DOS sin importar el fabricante.

El BIOS responde a interrupciones generadas por hardware o software. Por ejemplo, una interrupción de hardware es generada cuando un usuario oprime la tecla *Print Screen*; lo que ocasiona que BIOS active una rutina que envía el contenido ASCII de la pantalla a la impresora.

En forma semejante, una interrupción de software es generada cuando un programa emite un comando para leer algo desde un archivo de disco. Esto hace que la CPU indique a BIOS que active una rutina para leer datos desde el disco y le proporciona lo siguiente: el número de sectores a transferir, el número de pista, el número de sector y el número de unidad. Una vez que la operación ha sido completada con éxito, indica a BIOS el número de sectores transferidos y envía un comando para limpiar el código. En caso de que durante la operación ocurra un error, el código de error se regresa de modo que BIOS puede exhibir el mensaje de error idóneo en la pantalla.

La mayor parte de los controladores de dispositivos forman parte de un MS-DOS normal. Por supuesto, siempre es posible que usted escriba su propio controlador de dispositivos. Todo lo que se requiere son conocimientos del lenguaje ensamblador, información sobre el hardware y algo de paciencia. Esta opción puede ser necesaria si se está usando un sistema con una combinación inusual de dispositivos. Por ejemplo, en sus primeros años de disponibilidad comercial, no había una gran demanda para interconectar una computadora con un reproductor de discos, de modo que sus controladores de dispositivos no estaban incorporados en BIOS. En consecuencia, los usuarios que deseaban usar un disco como dispositivo de E/S debían escribir o comprar sus propios controladores de dispositivos y cargarlos cuando se arrancaba el sistema. Estos controladores de dispositivos se denominan instalables porque pueden incorporarse en el sistema operativo según sea necesario sin tener que parchar o modificar el sistema operativo existente. Los controladores de dispositivos instalables son una característica sobresaliente del diseño del MS-DOS.

Administración de archivos


MS-DOS atiende organizaciones de archivos secuenciales, directos e indexados secuenciales. Los archivos secuenciales pueden tener registros de longitud variable o fija. Sin embargo, los archivos directos y los indexados secuenciales sólo pueden tener registros de longitud fija.

Convenciones de nomenclatura de archivos

Un nombre de archivo no contiene espacios en blanco y consta de la identificación de la unidad, el directorio, cualquier subdirectorio, un nombre base y una extensión opcional. (DOS no es sensible a mayúsculas ó minúsculas, de modo que los nombres de archivo pueden escribirse con letras mayúsculas, minúsculas o una combinación de ambas.)

La identificación de la unidad (normalmente A, B, C o D), está seguida por dos puntos (:). Los directorios o los subdirectorios pueden tener entre uno y ocho caracteres de longitud y van precedidos por un backslash (\). El nombre de archivo base puede tener entre uno y ocho caracteres de longitud y la extensión, de uno a tres caracteres. El nombre base y las extensiones van separadas por un punto. Una extensión de archivo puede tener un significado especial para DOS; el usuario debe conocer las extensiones y sus usos.

Si en el nombre no se incluyen directorios o subdirectorios, se supone que el archivo está en el **directorio de trabajo** actual. En caso que no se designen unidades, se supone que el archivo

 MS-DOS no es sensible a mayúsculas o minúsculas, de modo que una letra mayúscula en un comando o nombre de archivo es equivalente a una minúscula.

está en la unidad actual. El directorio raíz (consulte la siguiente sección, Administración de archivos, para un análisis de esto) se llama con un simple *backslash* (\). Los nombres de otros directorios van precedidos por el símbolo *backslash*, que también se usa como delimitador entre nombres.

Un nombre relativo de archivo consta de su nombre base y una extensión, en caso de usarse. Un nombre absoluto de archivo consta de su identificación de unidad y la ubicación de su directorio (denominada **ruta**) seguidas por su nombre relativo. Cuando un usuario está trabajando en un directorio o en un subdirectorío, el directorio en cuestión se denomina directorio de trabajo y a cualquier archivo en ese directorio puede tenerse acceso mediante su nombre relativo. Sin embargo, para tener acceso a un archivo que está en otro directorio se requiere el nombre absoluto.

Por ejemplo, si su directorio de trabajo incluye un archivo denominado JORDAN.DOC, entonces es posible identificar el archivo al escribir:

```
JORDAN . DOC
```

No obstante, si usted ha cambiado a otro directorio de trabajo, entonces tal vez deba incluir el nombre del directorio cuando llame al archivo:

```
\JOURNAL\CHAP9\JORDAN . DOC
```

Y si ha cambiado a otra unidad, como la unidad A, y desea llamar al archivo, quizá también deba incluir la identificación de la unidad:

```
C : \JOURNAL\CHAP9\JORDAN . DOC
```

Los nombres de archivos DOS no pueden tener espacios entre ellos. En consecuencia, para copiar un archivo de la unidad C a la unidad B, este es el comando:

```
COPY C:\MEMO\DEAN.DOC B:DEAN.DOC
```

Una forma más simple para tener acceso a archivos consiste en escoger primero un directorio y luego acceder a los archivos dentro de ese directorio mediante sus nombres relativos. Después, cuando haya terminado con un directorio, puede emitir el comando de cambio de directorio (consulte la tabla 14.4 más tarde en este capítulo) para dirigirse a otro directorio de trabajo.

Por supuesto, hay muchas variantes. Para detalles completos, consulte un manual técnico para su versión de MS-DOS.

Administración de archivos

Las primeras versiones de MS-DOS mantenían cada archivo en un solo directorio. Esto era lento y engorroso, especialmente a medida que los usuarios agregaban más y más archivos.

Para recuperar un solo archivo, el administrador de archivos buscaba desde el principio de la lista hasta encontrar el archivo o llegar al final de la lista. Si un usuario olvidaba el nombre del archivo, había muchas posibilidades de no volver a verlo jamás.

Para resolver este problema, Microsoft implementó una estructura jerárquica de directorios en la versión 2.0; una estructura de árbol invertida. (Es invertida porque la raíz está en la parte superior y las hojas, en la parte inferior.)

Cuando se formatea un disco (usando el comando `FORMAT`) sus pistas se dividen en sectores de 512 bytes cada uno. (Esto corresponde a un tamaño de *búfer* de 512 bytes). Los discos de un solo lado tienen una superficie grabable; los de dos lados tienen dos superficies grabables, y los discos duros tienen de dos a cuatro platos, cada uno con dos superficies grabables. El concepto de cilindros, presentado en el capítulo 7, es válido para estos discos duros porque las cabezas de lectura/escritura se mueven al unísono.

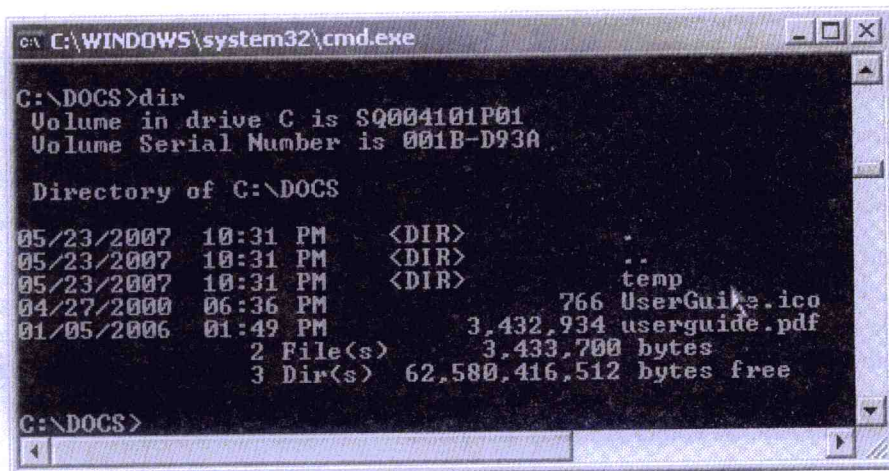
Los sectores (de dos a ocho) están agrupados en *clusters*, y esa es la manera en que el administrador de archivos asigna espacio a los archivos. Cuando un archivo requiere espacio adicional, DOS le asigna más *clusters*. Además de dividir el espacio del disco, `FORMAT` crea tres áreas especiales sobre el disco: el registro de arranque, el directorio raíz y la FAT, que significa **tabla de asignación de archivos**.

El *registro de arranque* es el primer sector de todo disco lógico, ya sea una unidad-entidad totalmente física (como un disquete o un disco duro) o sólo un disco virtual (como un disco RAM). Empezando con la versión 2.0, el registro de arranque contiene el programa del disco de arranque y una tabla con las características del disco.

El *directorio raíz* es donde el sistema empieza su interacción con el usuario cuando arranca. El directorio raíz contiene una lista de los subdirectorios y archivos primarios del sistema, incluyendo cualquier archivo de configuración generado por el sistema y cualquier instrucción de arranque generada por el usuario que pudiera estar incluida en un archivo `AUTOEXEC.BAT`. Se trata de un **archivo por lotes** que contiene una serie de comandos definidos por el usuario. Cada que se enciende o restablece la CPU, el sistema ejecuta de manera automática los comandos en este archivo. Después, en este capítulo se analiza un archivo muestra `AUTOEXEC.BAT`.

La información que contiene el directorio raíz es: 1) el nombre del archivo, 2) la extensión del archivo, 3) el tamaño del archivo en bytes, 4) la fecha y hora en que se hizo la última modificación al archivo, 5) el número del *cluster* inicial para el archivo y 6) los códigos de atributos del archivo. De estos seis elementos, los cuatro primeros se exhiben en respuesta al comando `DIR`, como se muestra en la figura 14.5.

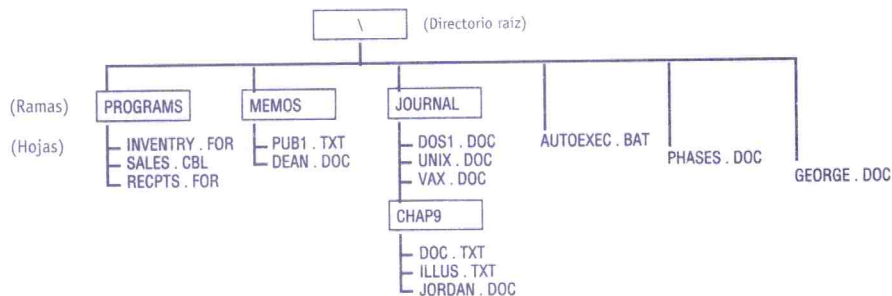
El número de entradas en el directorio raíz es fijo. Por ejemplo, para un disco duro de 20 MB sólo se permiten 512 entradas. El tamaño del directorio raíz es limitado porque DOS requiere saber dónde empieza el área de datos del disco. Empezando con la versión 2.0, los usuarios pueden evitar esta limitación al crear subdirectorios sin límite de tamaño.



(Figura 14.5)
Lista de un directorio DOCS que muestra tres subdirectorios y dos archivos.

La lista del directorio que se muestra en la figura 14.5 fue generada por el comando: DIR. Observe la manera en que los tres subdirectorios se distinguen de los dos archivos. Observe también que el sistema mantiene la fecha y la hora en que fue modificada la última vez. Algunos programas de seguridad del software usan estos datos para detectar cualquier virus o alguna modificación no autorizada o inusual del software del sistema.

Cada subdirectorio puede contener sus propios subdirectorios y/o archivos, como se muestra en la figura 14.6.



(Figura 14.6)
Sistema de directorios: la lista del directorio raíz tiene seis entradas. La lista del directorio para JOURNAL tiene cuatro entradas: sus tres archivos y su subdirectorio CHAP9. La lista del directorio para CHAP9 tiene tres entradas para sus tres archivos.

MS-DOS atiende archivos ocultos; archivos que son ejecutables aunque no se exhiben en respuesta a comandos DIR. Algunos archivos del sistema MS-DOS son ocultos; se usan para ejecutar el sistema operativo pero no aparecen en las listas de directorios. COMMAND.COM es el único archivo del sistema que no es oculto, de modo que siempre aparece en directorios públicos.

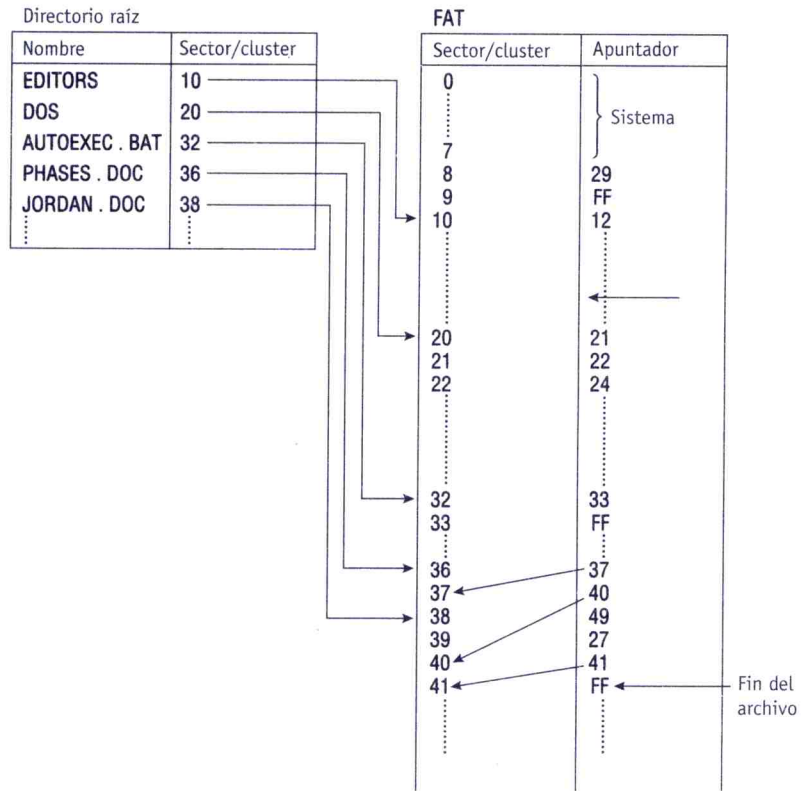
La tabla de asignación de archivos (FAT) contiene información crítica de estado, incluyendo la capacidad de cada sector para ser asignado o no, lo que puede ser el resultado de errores de formateo que interpretan al sector como no utilizable.

El directorio nota el número del primer sector o *cluster* del archivo; este número se registra en el directorio cuando se crea el archivo. Todos los sectores o *clusters* sucesivos asignados a

ese archivo se registran en la FAT, y se enlazan entre sí para formar una cadena, donde cada entrada de la FAT proporciona el número de sector/*cluster* a la siguiente entrada. La última entrada de cada cadena contiene el valor hexadecimal FF para indicar el final de la cadena. Como puede verse en la figura 14.7, los sectores de un archivo no tienen que ser contiguos.

(Figura 14.7)

Para cada archivo, el directorio incluye la ubicación del primer sector/*cluster* en la tabla de asignación de archivos, de modo que puede tenerse acceso rápido a éste. La FAT enlaza cada sector para cada archivo. Observe que los sectores para el archivo PHASES.DOC no son contiguos (las flechas son una ayuda visual para mostrar su enlace).



MS-DOS busca en los datos en un archivo del disco como una cadena continua de bytes. En consecuencia, las operaciones de E/S solicitan los datos por byte relativo (relativo con respecto al inicio del archivo) en vez de por sector relativo. La transformación de un sector físico (o *cluster*) en una dirección relativa de byte la efectúa el administrador de archivos de modo que parece que el acceso a los datos sobre un disco es justo como el acceso a los datos en la memoria principal.

Como ya mencionamos, MS-DOS atiende el almacenamiento de archivos no contiguo y asigna de manera dinámica espacio de disco a un archivo, en el supuesto de que en el disco hay suficiente espacio. Desafortunadamente, a medida que en el disco se agregan y borran archivos, un archivo puede volverse demasiado fragmentado, haciendo cada vez más engorrosa y consumidora de tiempo su recuperación.

La **compactación** se volvió una característica de la versión 6.0 de MS-DOS con la inclusión de DEFRAG.EXE, una utilidad usada para desfragmentar discos al escoger los fragmentos de un archivo y repositonarlos como una sola pieza en espacio contiguo.

Para determinar la necesidad de compactación puede usarlas otro comando. Dado CHKDSK (nombre del archivo), el sistema responde con el número de bloques no contiguos donde está almacenado el archivo. Es decisión del usuario compactar al archivo, en caso de ser necesario, de modo que se almacena en el menor número de bloques no contiguos para acelerar el tiempo de acceso y reducir el mantenimiento sobre el mecanismo de búsqueda.

La habilidad para restringir el acceso del usuario al sistema de cómputo y sus recursos no está incluida en MS-DOS. Se cuenta con software de seguridad que puede agregarse aunque, para la mayoría de los usuarios, los datos se mantienen seguros al mantener la computadora cerrada físicamente o al retirar los discos y guardarlos en un sitio seguro.

Interfaz de usuario

MS-DOS usa una **interfaz controlada por comandos**. En la tabla 14.4 se muestran algunos de los comandos más comunes. Los usuarios escriben sus comandos en el símbolo del sistema. El símbolo del sistema por defecto es el indicador de la unidad (como C: y el caracter >; en consecuencia, C:> es el símbolo del sistema normal para un sistema con disco duro y A:> es el símbolo del sistema para una computadora con una unidad de disquete. El símbolo del sistema predeterminado puede cambiarse usando el comando PROMPT.

Comando	Significa	Acción a ejecutar
DIR	Directorio	Hacer una lista de lo que está en el directorio.
CD o CHDIR	Cambiar directorio	Cambiar el directorio de trabajo.
COPY	Copiar	Copiar un archivo. Agregar uno a otro.
DEL o ERASE	Borrar	Borrar el archivo o siguiente o los archivos siguientes.
RENAME	Renombrar	Renombrar un archivo.
TYPE	Escribir	Mostrar en la pantalla el texto del archivo.
PRINT	Imprimir	Imprimir uno o más archivos.
DATE	Fecha	Mostrar y/o cambiar la fecha del sistema.
TIME	Hora	Mostrar y/o cambiar la hora del sistema.
MD o MKDIR	Hacer directorio	Crear un nuevo directorio o subdirectorio.
FIND	Encontrar	Encontrar una cadena. Buscar en archivos una cadena.
FORMAT	Formatear disco	Preparar lógicamente un disco para almacenamiento de archivos.
CHKDSK	Comprobar disco	Comprobar disco respecto a estado de disco/archivo/directorio.
PROMPT	Símbolo del sistema del sistema	Cambiar el símbolo del sistema.
DEFRAG	Desfragmentar disco	Compactar archivos fragmentados.
(Nombre del archivo)		Ejecutar el archivo.

(Tabla 14.4)

Algunos comandos MS-DOS de usuarios comunes. Los comandos pueden escribirse en letras mayúsculas o minúsculas, aunque en este texto sólo usamos mayúsculas por consistencia en la notación. Consulte documentación técnica para ortografía y sintaxis correctas para su sistema.

Cuando el usuario oprime la tecla Enter, el shell denominado COMMAND.COM interpreta el comando y llama a la siguiente rutina de nivel inferior para satisfacer la solicitud.

Comandos de usuario incluyen algunos o todos los siguientes elementos en este orden:

`command source-file destination-file switches`

El comando es cualquier comando MS-DOS legal. El archivo fuente y el archivo destino se incluyen cuando es válido y, dependiendo de la unidad y el directorio actuales, podría ser necesario incluir la ruta completa del archivo. Los cambios comienzan con un *slash* (es decir: /P/V/F) y son opcionales; proporcionan detalles específicos sobre cómo debe efectuarse el comando. La mayor parte de los comandos requieren un espacio entre cada uno de sus elementos.

Los comandos son ejecutados por el archivo COMMAND.COM, que forma parte de MS-DOS, como se muestra en la figura 14.2. Como ya mencionamos, cuando COMMAND.COM se carga durante el inicio del sistema, una de sus secciones se almacena en la sección baja de la memoria: esta es la porción residente del código. Contiene el intérprete de comandos y las rutinas necesarias para atender un programa activo. Además, contiene las rutinas necesarias para procesar CTRL-C, CTRL-BREAK y los errores críticos.

El código transitorio, la segunda sección de COMMAND.COM, se almacena en las direcciones más altas de la memoria y pueden sobrescribirse por programas de aplicación si necesitan usar su espacio de la memoria. Después, cuando termina el programa, la porción residente de COMMAND.COM verifica si el código transitorio sigue intacto. En caso negativo, carga una nueva copia.

A medida que un usuario escribe un comando, cada carácter se almacena en la memoria y aparece en la pantalla. Cuando se oprime la tecla Enter, el sistema operativo transfiere el control a la porción del intérprete de comandos de COMMAND.COM, que entonces tiene acceso a la rutina que lleva a cabo la solicitud o muestra un mensaje de error. Si la rutina reside en la memoria, entonces el control se le proporciona de manera directa. Si la rutina reside en el almacenamiento secundario, se carga en la memoria y después se le proporciona el control.

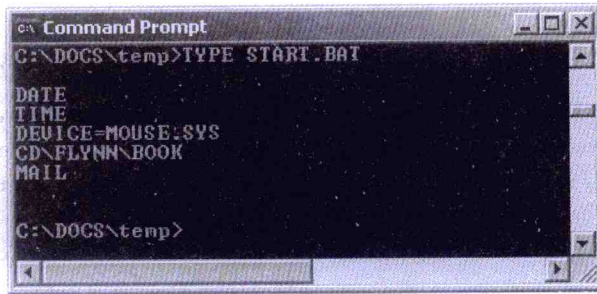
Aunque no es posible describir todos los comandos que hay en MS-DOS, merece la pena destacar algunas características para mostrar la flexibilidad de este sistema operativo.

Archivos por lotes

Al crear archivos por lotes personalizados, los usuarios pueden ejecutar rápidamente combinaciones de comandos DOS para configurar sus sistemas, realizar tareas de rutina, o facilitar que usuarios no técnicos puedan ejecutar software.

Por ejemplo, si un usuario verifica rutinariamente la fecha y hora del sistema, carga un controlador de dispositivos para un ratón, se desplaza a cierto subdirectorío y carga un pro-

grama denominado MAIL.COM, entonces este programa, denominado START.BAT, realiza cada uno de esos pasos a su vez como se muestra en la figura 14.8.



(Figura 14.8)

Contenido del programa START.BAT.

Para ejecutar este programa, el usuario sólo necesita escribir START en el símbolo del sistema. Para que este programa se ejecute automáticamente cada que se reinicia el sistema, entonces el archivo debe volverse a nombrar AUTOEXEC.BAT y cargarse en el directorio raíz del sistema. Al usar archivos por lotes, cualquier combinación tediosa de teclas puede reducirse a unos cuantos comandos personalizados fáciles de recordar.

Redirección

MS-DOS puede redirigir la salida desde un dispositivo de entrada o salida normal hacia otro. Por ejemplo, el comando DATE envía el resultado directamente a la pantalla pero al usar el símbolo de redirección (>), en lugar de lo anterior la salida se redirige a otro dispositivo o archivo.

La sintaxis es:

```
command > destination
```

Por ejemplo, si se desea enviar una lista del directorio a la impresora, debe escribirse DIR > PRN y la lista aparece en la página impresa, en lugar de en la pantalla. En forma semejante, si se desea que la unidad predeterminada sea redirigida a un archivo en el disco en la unidad B, debe escribirse DIR > B:DIRFILE y en la unidad B se crea un nuevo archivo denominado DIRFILE, que contiene una lista del directorio.

Es posible redirigir y agregar nuevos resultados hacia un archivo existente al usar el símbolo de agregar (>>). Por ejemplo, si ya se ha creado el archivo DIRFILE con el comando de redirección y se desea generar una lista del directorio y agregarla en el archivo DIRFILE creado previamente, debe escribirse:

```
DIR >> B:DIRFILE
```

Ahora DIRFILE contiene dos listas del mismo directorio.

La redirección también funciona de manera opuesta. Si se quiere modificar la fuente a un dispositivo o archivo específico, debe usarse el símbolo <. Por ejemplo, suponga que tiene en

desarrollo un programa denominado INVENTORY.EXE que espera datos de entrada desde el teclado, aunque para efectos de prueba y depuración se requiere que acepte datos de entrada desde un archivo de prueba de datos. En este caso debe escribirse lo siguiente:

```
INVENTORY < B:TEST.DAT
```

Filtros

Los **comandos filtro** aceptan datos de entrada desde el dispositivo predeterminado, manipulan los datos de alguna manera y envían los resultados al dispositivo de salida predeterminado. Un filtro de uso común es SORT, que acepta datos de entrada desde el teclado, ordena esos datos y los muestra en la pantalla. Este comando filtro se vuelve aún más útil al usar parámetros de redirección. Por ejemplo, si usted desea ordenar un archivo de datos denominado STD.DAT y almacenarlo en otro denominado SORTSTD.DAT, debe escribir:

```
SORT <STD.DAT> SORTSTD.DAT
```

El archivo almacenado puede estar en orden creciente (numérica o alfabéticamente) empezando con el primer caracter en cada línea del archivo. Si desea que el archivo esté almacenado en orden inverso, debe escribir:

```
SORT /R <STD.DAT> SORTSTD.DAT
```

Es posible ordenar el archivo por columnas. Por ejemplo, suponga que un archivo denominado EMPL contiene datos que siguen este formato: los números de identificación empiezan en la columna 1, los números telefónicos empiezan en la columna 6, y los apellidos empiezan en la columna 14. (Una columna se define como caracteres delimitados por uno o más espacios.) Para ordenar el archivo por apellido, el comando sería:

```
SORT /+14 <EMPL.DAT> SORTEEMPL.DAT
```

El archivo estaría ordenado en orden creciente por el campo empezando en la columna 14.

Otro filtro común es MORE, que hace que los resultados aparezcan en la pantalla en grupos de 24 líneas y espera hasta que el usuario oprime la tecla Enter antes de mostrar las 24 líneas siguientes.

Tuberías

Una **tubería** puede ocasionar que la salida normal de un comando sea utilizada como entrada normal para otro comando; este símbolo es una barra vertical (|). Es posible ordenar alfabéticamente el directorio y mostrar la lista ordenada en la pantalla al escribir:

```
DIR | SORT
```

También es posible combinar tuberías y otros filtros. Por ejemplo, para mostrar en la pantalla el contenido del archivo INVENTORY.DAT a la vez, el comando sería:

```
TYPE INVENTORY.DAT | MORE
```

El mismo resultado puede obtenerse usando sólo redirección al escribir:

```
MORE < INVENTORY.DAT
```

Es posible ordenar el archivo y mostrar el resultado en la pantalla a la vez al usar tuberías con este comando:

```
DIR | SORT | MORE
```

O bien, el mismo resultado puede obtenerse al usar tanto tuberías como filtros con estos dos comandos:

```
DIR | SORT > SORTFILE
```

```
MORE < SORTFILE
```

Comandos adicionales

Tres comandos adicionales que se usan a menudo en MS-DOS son FIND, PRINT y TREE. Observe que estos comandos son “tradicionales” de MS-DOS, y que algunos de los cambios u opciones mencionados aquí pueden no funcionar en emuladores semejantes a Windows DOS.

FIND

FIND es un comando filtro que busca una cadena específica en un archivo o archivos dados y despliega todas las líneas que contienen la cadena de esos archivos. La cadena debe estar entre comillas dobles (“”) y debe escribirse exactamente como se busca; las letras mayúsculas y minúsculas se toman según se introducen.

Por ejemplo, el comando para mostrar todas las líneas en el archivo PAYROLL.COB que contienen la cadena AMNT-PAID es:

```
FIND "AMNT-PAID" PAYROLL.COB
```

El comando para contar el número de líneas en el archivo PAYROLL.COB que contiene la cadena AMNT-PAID y mostrar el número en la pantalla es:

```
FIND /C "AMNT-PAID" PAYROLL.COB
```

El comando para mostrar el número de línea relativo, así como la línea en el archivo PAYROLL.COB que contiene la cadena AMNT-PAID es:

```
FIND /N "AMNT-PAID" PAYROLL.COB
```

El comando para mostrar todas las líneas en el archivo PAYROLL.COB que *no contiene* la cadena AMNT-PAID es:

```
FIND /V "AMNT-PAID" PAYROLL.COB
```

El comando para mostrar los nombres de todos los archivos en el disco en la unidad B y que *no contienen* la cadena SYS es:

```
DIR B: | FIND /V "SYS"
```

PRINT

Este comando permite que el usuario establezca una serie de archivos para imprimir mientras libera al COMMAND.COM para aceptar otros comandos. En efecto, se trata de un *spooler*. A medida que la impresora imprime sus archivos, usted puede escribir otros comandos y trabajar en otras aplicaciones. El comando PRINT tiene muchas opciones, pero para usar las dos siguientes, éstas deben proporcionarse la primera vez que se usa el comando PRINT después de arrancar el sistema.

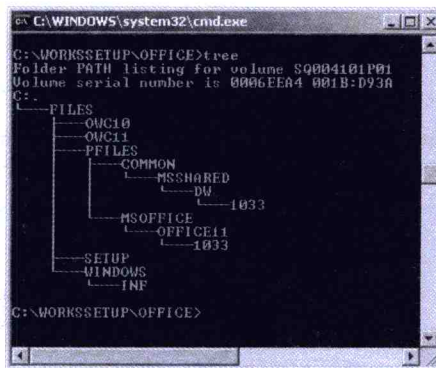
- El comando PRINT/B permite cambiar el tamaño del *búfer* interno. Por defecto, es de 512 bytes, pero aumentar su valor acelera el proceso PRINT.
- El comando PRINT/Q especifica el número de archivos permitidos en la siguiente cola de impresión. El valor mínimo para Q es 4 y el máximo, 32.

TREE

El comando TREE muestra los directorios y subdirectorios en una lista jerárquica y con sangría. También tiene opciones que permiten que el usuario borre archivos mientras se está generando el árbol. El desplegado empieza con el directorio actual o especificado con los subdirectorios con sangría bajo el directorio que los contiene. Por ejemplo, si se emite el comando TREE, la respuesta sería semejante a lo que se muestra en la figura 14.9.

(Figura 14.9)

Resultados muestra del comando TREE.



Para mostrar los nombres de los archivos en cada directorio, debe agregarse el conmutador /F:

```
TREE /F
```

El comando TREE también puede usarse para borrar un archivo que esté duplicado en diferentes directorios. Por ejemplo, para borrar el archivo PAYROLL.COB en cualquier parte del disco, el comando sería:

```
TREE PAYROLL.COB /D /Q
```

El sistema muestra el árbol como se acostumbra pero siempre que encuentra un archivo denominado PAYROLL.COB hace una pausa y pregunta si usted quiere que se borre. Si usted escribe Y, entonces borra el archivo y continúa. Si escribe N, entonces prosigue como antes.

Para efectos ilustrativos, aquí sólo hemos incluido unos cuantos comandos MS-DOS. Para una lista completa de comandos, su sintaxis exacta y más detalles sobre lo que se ha analizado aquí, consulte la página www.microsoft.com.

Conclusión

MS-DOS fue escrito para rendir servicio a los usuarios de computadoras personales en la década de 1980, incluyendo las primeras versiones de las computadoras personales de IBM. Como tal, tuvo éxito pero su flexibilidad limitada lo hicieron inutilizable a medida que evolucionaban las computadoras personales.

MS-DOS se recuerda como el primer sistema operativo normal que fue adoptado por muchos fabricantes de máquinas de cálculo personales. Como estándar también atendía, y es atendido, por legiones de grupos de diseño de software.

La debilidad de MS-DOS fue su diseño de sistema de usuario único de tarea única que no podía atender multitarea, redes y otras aplicaciones sofisticadas requeridas por computadoras de todos tamaños. Hoy es una reliquia del pasado, pero su estructura simple e interfaz de usuario lo hicieron una herramienta de aprendizaje accesible para estudiantes de sistemas operativos.

Términos clave

Archivo por lotes: Archivo que incluye una serie de comandos que se ejecutan en secuencia sin ningún dato de entrada de parte del usuario. Contrasta con una sesión interactiva.

Asignación de memoria del primer ajuste: Esquema de asignación de memoria que busca desde el principio del bloque de árbol y escoge para su asignación el primer bloque de memoria suficientemente grande para satisfacer la solicitud. Contrasta con la *asignación de memoria del mejor ajuste*.

BIOS: Acrónimo de *basic input output system* (sistema básico de entrada salida), un conjunto de programas codificados en un chip para cargarse en ROM al inicio.

Bootstrapping: Proceso de arrancar una computadora inactiva al usar un pequeño programa de inicio para cargar otros programas.

Comando filtro: Comando que dirige los datos de entrada desde un dispositivo o archivo, los modifica y después envía el resultado a una impresora o una pantalla.

Compactación: Proceso de reunir fragmentos de espacio de memoria disponible en bloques contiguos al mover programas y datos en la memoria o en almacenamiento secundario de una computadora.

Controlador de dispositivos: Módulo de programa específico que maneja las interrupciones y controla un tipo particular de dispositivo.

Directorio de trabajo: Directorio o subdirectorio que actualmente es el que está usando el directorio principal.

Extensión: Parte del nombre del archivo que indica cuál compilador o paquete de software es necesario para ejecutar los archivos.

Interfaz controlada por comandos: Interfaz que acepta comandos por escrito, una línea a la vez, del usuario. Contrasta con una interfaz controlada por menú.

Manejador de interrupciones: Programa que controla la acción que debe emprender el sistema operativo cuando se interrumpe una secuencia de eventos.

Multitarea: Sinónimo de multiprogramación, una técnica que permite que un solo procesador procese varios programas que residen simultáneamente en la memoria principal e intercala su ejecución al superponer solicitudes de E/S con solicitudes de CPU.

Redirección: Instrucción que dirige al sistema operativo para enviar los resultados de un comando hacia o desde un archivo u otro dispositivo que no sean el teclado o el monitor. En MS-DOS, los símbolos de redirección son < y >.

Ruta: Secuencia de directorios y subdirectorios que deben seguir el sistema operativo para encontrar un archivo específico.

Símbolo del sistema del sistema: Señal desde el sistema operativo que está lista para aceptar un comando del usuario, como C:\> o C:\Documents>.

Tabla de asignación de archivos (FAT): Tabla usada para seguir la pista de segmentos de un archivo.

Tubería: Símbolo que dirige al sistema operativo para diversificar los resultados de un comando, de modo que se vuelvan los datos de entrada de otro comando. En MS-DOS, el símbolo de una tubería es |.

Búsquedas de interés

- Emulador MS-DOS
- Archivo por lotes autoejecutable
- Basic Input/Output System (BIOS)

- Interfaz de usuario controlada por comandos
- Sintaxis de comandos MS-DOS

Ejercicios

Temas de investigación

- A. Explore el mundo de la computación al inicio de la década de 1980 e identifique varias razones de la popularidad de MS-DOS en ese momento. Haga una lista de los sistemas operativos competidores y las marcas de computadoras personales que estaban disponibles. Cite sus fuentes.
- B. De acuerdo con www.microsoft.com, la empresa sigue manteniendo MS-DOS debido a que este sistema operativo está en uso en sitios alrededor del mundo. Dirija su propia investigación para encontrar un sitio en el que aún se esté ejecutando MS-DOS y explique por qué este es el sistema operativo elegido ahí.

Ejercicios

1. Describa el propósito de todas las interfaces de usuario, ya sea guiadas por comandos o por menús.
2. Mencione cinco ventajas que una interfaz de usuario guiada por comandos tiene sobre una interfaz de usuario guiada por menús.
3. ¿Cómo se construye un nombre de archivo legal en MS-DOS? Describa la longitud máxima y los roles de los caracteres especiales, mayúsculas/minúsculas, slashes, etcétera.
4. ¿Cómo se comparan los tamaños de los búfers de un sistema con los sectores de un disco? ¿Cuál es mayor? Explique por qué esto es así.
5. Proporcione ejemplos de los comandos CD, DIR y TREE y explique por qué decidiría usar cada uno.
6. Abra el emulador de MS-DOS de un sistema operativo Windows (quizá bajo el Menú de Accesorios y llamado "Command Prompt"). Cambie a un directorio con varios archivos y subdirectorios y ejecute un comando DIR. ¿Cómo está ordenado el directorio resultante (alfabética, cronológicamente, otra)?
7. Abra el emulador de MS-DOS de un sistema operativo Windows y haga un listado de directorios del directorio raíz (utilice el comando CD\ y luego el comando DIR). Después utilice el sistema operativo Windows y abra la carpeta C. Compare las dos listas y explique en qué son similares y en qué son diferentes.
8. Abra el emulador de MS-DOS de un sistema operativo Windows y ejecute el comando TREE. Explique si tener o no acceso a este comando MS-DOS podría ser valioso para un usuario de Windows.

9. Describa el rol de una tabla de asignación de archivos (FAT) y cómo ésta administra los archivos.
10. ¿Cómo trabaja el directorio descrito en este capítulo comparado con el grupo de trabajo detallado en el capítulo 3?

Ejercicios avanzados

11. Si usted fuera a configurar una oficina pequeña con 10 computadoras personales ejecutando sólo MS-DOS, describa cómo podría conectarlas en red y cuántas copias del sistema operativo necesitaría comprar.
12. El comando FORMAT borra cualquier dato existente en el disco al formatearlo o reformatearlo. Describa qué características de seguridad tendría que agregar al sistema para prevenir el uso inadvertido de este comando.
13. Explique por qué es necesaria una rutina de arranque como elemento de todo sistema operativo.
14. Describa cómo MS-DOS ejecuta un arranque en frío y en qué orden este controlador de disco es ingresado. Explique por qué este orden hace o no sentido.
15. Describa cómo agregaría un control de acceso para proteger datos sensibles en una computadora al ejecutar MS-DOS. ¿Puede describir las soluciones de hardware y de software? Describa otras características que usted desearía agregar al sistema para hacerlo más seguro.
16. La rutina de arranque está almacenada en el primer sector del disco del sistema hasta que esta rutina puede ser el primer software cargado en la memoria cuando se enciende la computadora. Conduzca su propia investigación para descubrir los elementos esenciales de la rutina de arranque y explique por qué este software es necesario para “iniciar” el sistema.

