



UNIVERSIDAD NACIONAL
AUTONOMA DE MEXICO

Facultad de Estudios Superiores
"Cuautitlán"



**"PROCEDIMIENTO DE PROTECCION DEL
SOFTWARE Y DEL SISTEMA OPERATIVO MS-DOS
DE IBM O COMPATIBLES"**

1992
FALLA DE ORIGEN

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO MECANICO ELECTRICISTA

P R E S E N T A,

IVAN ERIC BELTRAN GUERRERO

Asesor: Ing. Jorge Buendía Gómez



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

ÍNDICE

OBJETIVOS	1
-----------	---

INTRODUCCIÓN	I
--------------	---

CAPÍTULO I

ESTRUCTURA DEL SISTEMA OPERATIVO	I
----------------------------------	---

UNA COMPUTADORA PERSONAL	1
HARDWARE BÁSICO DE UN SISTEMA DE COMPUTACIÓN	2
SOFTWARE DE LA COMPUTADORA	4
MÁQUINA VIRTUAL	8
SISTEMA OPERATIVO COMO MANEJADOR DE RECURSOS	9
MÓDULOS QUE COMPONEN EL SISTEMA OPERATIVO	11
ADMINISTRADOR DE MEMORIA	14
ADMINISTRADOR DEL PROCESADOR	15
PROGRAMADOR DE TAREAS	15
CONTROLADOR DE TRÁFICO	18
ADMINISTRADOR DE DISPOSITIVOS	18
ADMINISTRADOR DE INFORMACIÓN	20

CAPÍTULO II

ESTRUCTURA DEL SISTEMA OPERATIVO MS-DOS	22
---	----

EL MS-DOS	23
MÓDULOS DEL SISTEMA OPERATIVO	25
EL CARGADOR DEL SISTEMA OPERATIVO	26
LAS RUTINAS DE INICIALIZACIÓN	28
EL SISTEMA DE INICIALIZACIÓN DE MS-DOS	37
EL MÓDULO DE INICIALIZACIÓN BIOS	37
MS-DOS KERNEL	39
EL COMMAND.COM	43
MANEJADOR DE MEMORIA	46
MEMORIA CONVENCIONAL	47
EL TPA	49
MANEJADORAS DE ARCHIVOS	54
FUNCIONES MANEJADORAS DE ARCHIVOS	56

ÍNDICE

CAPÍTULO II

EL CONTROL DE PROCESOS	67
DISPOSITIVOS PERIFÉRICOS	69
EL TECLADO	72
EL MONITOR	74
LA IMPRESORA	76

CAPÍTULO III

ARQUITECTURA DEL DISCO	77
EL DISCOS	78
LA ESTRUCTURA GENERAL	80
EL BOOT	82
LA FAT	85
CAMPOS DE LA FAT	87
EL DIRECTORIO DEL DISCO	92
EL DISCO DURO	95

CAPÍTULO IV

PROGRAMAS DE APLICACIÓN	98
EL PSP	99
LOS PROGRAMAS .COM	103
LOS PROGRAMAS .EXE	105
LA FUNCIÓN EXEC	113
SUMARIO DE DIFERENCIAS	116
PROGRAMAS RESIDENTES	117

ÍNDICE

CAPÍTULO V

DESARROLLO DEL PROGRAMA DE PROTECCIÓN	119
LOS MÓDULOS PRINCIPALES	120
LA AYUDA	122
EL TECLADO	123
LOS DISCOS FLEXIBLES	126
EL DISCO DURO	128
EL MÓDULO INDEPENDIENTE	129

CAPÍTULO VI

GUÍA DEL USUARIO	130
INTRODUCCIÓN	131
AMBIENTE	131
INSTALACIÓN	132
TECLAS UTILIZADAS	133
DESCRIPCIÓN DE MENÚS	134
MENÚ PRINCIPAL	134
PASSWORD DE TECLADO	134
PASSWORD DE FLOPPYS	134
DISCO FIJO	135
DESACTIVACIÓN DEL TECLADO	136
NUEVA CLAVE DE TECLADO	136
DESACTIVAR DISCOS FLEXIBLES	137
ACTIVAR DISCO FLEXIBLE	137
RESTAURAR DISCO FLEXIBLE	137
LA ACTIVACIÓN POR DISCO FLEXIBLE	137
CONCLUSIONES	138
REFERENCIA BIBLIOGRÁFICA	140

OBJETIVOS

A) Impedir el acceso a la computadora personal, a un usuario no autorizado.

B) Evitar el acceso a programas que tienen como fin, dañar la información almacenada en el disco de la computadora.

INTRODUCCIÓN

Los sistemas de archivos con frecuencia contienen información muy valiosa para los usuarios. La protección de esta información contra el uso no autorizado es de interés fundamental de todos los sistemas de archivos.

Los términos "seguridad" y "protección" a menudo se emplean en forma intercambiable. No obstante, con frecuencia es de utilidad hacer una distinción entre los problemas generales que intervienen al asegurar que los archivos no sean leídos o modificados por personas no autorizadas, entre los que se encuentran aspectos técnicos administrativos, legales y políticos por un lado y los mecanismos específicos del sistema operativo que se utilizan para brindar seguridad.

Para evitar confusión, utilizaremos el término SEGURIDAD para referirnos al problema general y el término mecanismos de PROTECCIÓN para referirnos a los mecanismos específicos que utiliza el usuario para proteger su información en la computadora.

Los mecanismos de protección aprovechan las características tanto de hardware como de software de la computadora para proteger la información.

La seguridad tiene muchas facetas. Dos de la más importantes son:

- La pérdida de información.
 - Los intrusos.
-

INTRODUCCIÓN

Algunas causas de la pérdida de información son:

1.- Fuerza mayor:

Incendios, inundaciones, terremotos, guerras, ratas que mordisquean cintas, discos flexibles, etc.

2.- Errores de hardware o software:

Mal funcionamiento de la unidad central de procesamiento, discos o cintas ilegibles, errores de telecomunicación, errores ocultos en los programas etc.

3.- Errores humanos:

Captación de datos incorrectos, montaje de cinta o el disco equivocado, disco o cinta extraviado etc.

La mayoría de estas causas se pueden controlar conservando respaldos adecuados, preferentemente que estén alejados de datos originales.

INTRODUCCIÓN

Un problema más interesante es que hacer con los intrusos. Estos presentan dos variedades:

- Los intrusos pasivos, que desean solamente leer archivos a los cuales no tienen autorización.
- Los intrusos activos, desean hacer cambios no autorizados a los datos.

Un problema aunado a esto, son los programas maliciosos.

Este tipo de programas maliciosos se pueden introducir a la computadora voluntaria o involuntariamente. Cuyas consecuencias pueden ser solo molestas o la pérdida de toda la información almacenada.

Modifican de alguna manera el funcionamiento normal y producen alguno de los siguientes efectos:

- Destruyen toda la información almacenada.
- Algunos se hacen presentes a través de la pantalla.
- Contaminan a la computadora en cuestión.

INTRODUCCIÓN

La gente que desarrolla estos programas, muchas veces modifican un programa comercial y le pasan a sus amistades las copias, que a su vez pasan a sus amistades y así sucesivamente. Al correr estos programas, independientemente de funcionar adecuadamente, modifican ciertos parámetros del disco duro o de la memoria de la máquina y de esta forma se afecta el equipo, pero aquí no acaba, un buen programa malicioso se copia en cualquier diskette que se use y cuando es usado en otra máquinas sigue copiándose.

Es sabido que en todo el mundo existe una gran cantidad de personas que copian y utilizan ilegalmente programas de computo.

De esta forma, puede llegar a una computadora casi en cualquier programa que se corra, sin conocer la procedencia del mismo y no necesariamente tiene que ser un programa comercial copiado ilegalmente.

Con tantas máquinas con este problema y sin encontrar respuesta a la procedencia de los programas maliciosos, se llegó a manejar la teoría de que los grandes fabricantes de software incluían en sus copias originales un código que detectaba si el programa había sido copiado ilegalmente, en caso afirmativo, se copiaba a la computadora y destruida la información de esta. Esto sigue siendo una teoría y no se sabe a ciencia cierta que tan válida sea.

INTRODUCCIÓN

Más tarde surgió otra teoría; las casas de software que desarrollan sistemas de protección contra copias ilegales, que son contratadas por los fabricantes, sin autorización del fabricante original, incluían en sus esquemas de protección programas maliciosos para adicionarse a las copias ilegales.

Todo esto se comenta, se platica y se critica, pero realmente no hay ninguna definición oficial del asunto.

La única realidad es que los programas maliciosos están invadiendo más máquinas cada día, fomentados por el acceso de usuarios ilegales a las computadoras.

De lo anterior se concluye que cualquier tipo de persona no autorizada en nuestra computadora es indeseable por los numerosos problemas que podría acarrear.

En la presente tesis se desarrolla un mecanismo de protección el cual evitará el acceso a personas no autorizadas por medio de restricciones al teclado, acceso a los discos flexibles y por último la deshabilitación del disco fijo.

Los primeros cuatro capítulos nos dan los conceptos para entender el funcionamiento de la computadora a un nivel interno. En el capítulo cinco tenemos los diagramas de flujo de cada módulo del proyecto y por último el capítulo seis es la guía de usuario.

INTRODUCCIÓN

Capítulo 1.- Estructura de un Sistema Operativo.

En este capítulo se describirá que es un Sistema Operativo, de que módulos está compuesto y los tipos de Sistema que hay. Esto nos servirá para irnos introduciendo al funcionamiento de una computadora y ver de que tipo de Sistema Operativo es el MS-DOS.

Capítulo 2.- Estructura del Sistema Operativo MS-DOS.

Se verá como trabaja este Sistema y de que módulos se compone. Esto nos indicará, el por qué se pensó en deshabilitar el teclado, evitar el acceso a los discos flexibles y deshabilitar el disco fijo.

Capítulo 3.- Estructura de Disco.

En este capítulo se dará la estructura de la unidad de almacenamiento (disco) para poder conocer como funciona el almacenamiento de datos y sugerirnos una forma de protección.

INTRODUCCIÓN

Capítulo 4.- Programas de aplicación.

Aquí veremos como están estructurados los archivos ejecutables y la diferencia entre estos.

Capítulo 5.- Desarrollo del programa de protección.

"PROSOFT" (Protección de Software), se dan los diagramas de flujo de cada módulo y su descripción.

Capítulo 6.- Guía de usuario.

Es un documento el cual podrá el usuario utilizar para la instalación, manejo y cuidados de la herramienta.

CAPITULO UNO



EL HARDWARE DE LA COMPUTADORA

Una computadora personal constará de un CPU (Unidad Central de Proceso), un monitor y el teclado (Fig. -1.1-), el CPU va a ser la parte fundamental de la computadora, sin esto no habría computadora. El monitor lo tomaremos como un dispositivo de salida, con el cual podremos ver el resultado de nuestras aplicaciones. Con el teclado tendremos acceso para comunicarnos con nuestras aplicaciones. En la figura el dispositivo apuntador se pone para ejemplificar uno de muchos dispositivos de entrada/salida que puede tener una computadora, como una impresora, un modem, un graficador, etc.

Una computadora sin ningún soporte lógico (conjunto de instrucciones que desarrollan una tarea específica, almacenada en un archivo el cual también es llamado programa -software-) no hace nada. No puede desplegar ningún carácter por el monitor, no podría cargar ni mucho menos ejecutar un programa.



Figura -1.1-
Dispositivos básicos de una computadora personal.

El diagrama de bloques del hardware básico de un sistema de computación puede ser dividido en cuatro secciones funcionales:

- Unidad de entrada (Input unit)
- Unidad de salida (Output unit)
- Unidad Central de Proceso (Central procesing unit -CPU-)
- Unida de memoria (Memory unit)

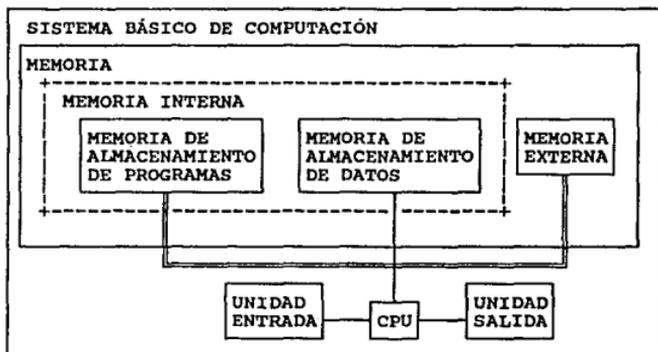


Figura -1.2-

El CPU es la parte de la computadora que ejecuta las instrucciones del programa, procesa los datos, es responsable de la realización de las operaciones aritméticas y decisiones lógicas iniciadas por el programa que está corriendo en la computadora, también controla todos los dispositivos de entrada / salida que integran la computadora.

Por medio de las unidades de entrada, por ejemplo el teclado y salida, como puede ser el monitor, el CPU se comunica con el mundo exterior o con otros dispositivos.

La unidad de memoria en la computadora se ajusta para almacenar la información como números o caracteres.

Los programas que definen como la computadora va a procesar la información almacenada, también residen en memoria.

Después de la descripción del diagrama de bloques del hardware de la computadora (Fig. -1.2-), esto nos conduce al tópico de ¿Qué es el software?

Una computadora no puede pensar acerca de como procesar los datos. Se le debe decir exactamente que hacer, donde conseguir los datos, que hacer con los datos y donde poner los resultados, cuando se ha realizado una tarea. Este es el trabajo del software de la computadora.

Software es un nombre general usado para referirse a una gran variedad de programas que corren en la computadora. Algunos ejemplos son:

- El Sistema Operativo
- Lenguajes de programación
- Manejadores de bases de datos
- Procesadores de palabras
- Editores
- Programas de aplicación, etc.

En el presente trabajo del software de la computadora, la parte que nos interesa es el Sistema Operativo.

ENFOQUES DEL SISTEMA OPERATIVO

La escritura de programas que lleven el control de todos los dispositivos y los utilicen correctamente y óptimamente es un trabajo difícil. Si el programador tuviera que involucrarse en la forma que trabajan las unidades de disco o algún otro dispositivo de la computadora, sería muy laborioso escribir algún programa ya que el programador primero tendría que controlar todos los dispositivos antes mencionados y después desarrollar su programa de aplicación.

Hace muchos años, se hizo evidente que tenía que descubrirse alguna manera de resguardar a los programadores de la complejidad para el manejo del hardware.

La estrategia que ha evolucionado en forma gradual consiste en colocar una capa de software en la parte superior del hardware al descubierto, con el objeto de manejar todas las partes que integran la computadora y presentar al usuario una interfase que es más fácil de entender y programar.

Esta capa de software es el SISTEMA OPERATIVO(Fig. -1.3-).

Entonces se puede decir que la función del Sistema Operativo consiste en ocultar la complejidad del manejo del hardware de la computadora para dar al programador una interfase o conjunto de instrucciones más adecuado con el cual trabajar.

En la parte superior del Sistema Operativo se encuentra el resto del software de la computadora.

Aquí se encuentran el interprete de comandos (SHELL como una interfaz entre el Sistema Operativo y el usuario), compiladores, procesadores de palabras, editores y programas de aplicación.

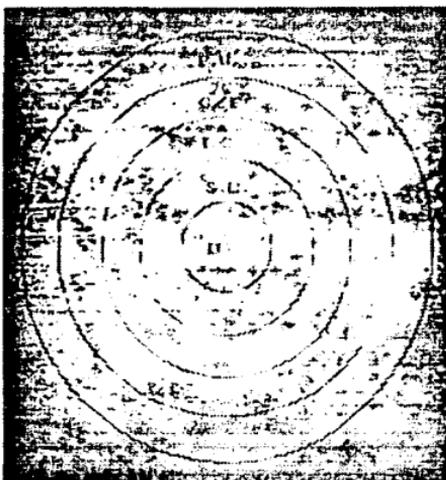


Figura -1.3-
Capas de un sistema de computación

Donde: HW - Hardware (Dispositivos físicos)

S.O.- Sistema Operativo

I.C.- Interprete de Comandos

C/E - Compiladores / Editores

P.U.- Programas de Usuario

Podemos concluir que un Sistema Operativo es la parte intermediaria entre el hardware y el usuario. Nos servirá para facilitar el uso de los dispositivos periféricos, tales como: almacenamiento de información en discos, cintas, cassettes, etc, u otro medio de almacenamiento, también para ver lo que se esta tecleando como: documentos, gráficas, etc., para la comunicación con otros dispositivos o interfases, para controlar Sistemas analógicos, la adquisición de datos para manejos estadísticos y la impresión de resultados.

Ya que se ha adquirido una idea de la ubicación y la función del Sistema Operativo dentro de un sistema de computación. Vamos a considerar dos enfoques diferentes :

- Una máquina ampliada (máquina virtual)
- Manejador de recursos

**EL SISTEMA OPERATIVO COMO UNA
MÁQUINA VIRTUAL**

Como se menciona antes, el lenguaje que maneja la máquina es de bajo nivel, es decir, que las instrucciones van a estar directamente ligadas con el hardware y por lo tanto hay que definir para cada dispositivo un número de instrucciones y esto lo hace difícil de programar, especialmente con la entrada/salida de información.

El programador en promedio no desea verse envuelto en la programación del hardware para poder realizar un programa; por ejemplo, el acceso de los discos flexibles. En su lugar, lo que el programador desea es una abstracción simple de alto nivel con la que se desea trabajar.

En el caso de los discos, una abstracción podría ser que el disco contenga un conjunto de archivos. Cada archivo puede abrirse para escritura y/o lectura, después de ser leído y/o escrito, cerrarlo. El programa que oculta la programación del hardware al programador y presenta una vista simple y agradable de los archivos que pueden ser leídos y/o escritos, es desde luego, el Sistema Operativo.

Así es como el Sistema Operativo resguarda al programador del hardware del disco y presenta una interfase simple, orientada al archivo, también disimula mucho del trabajo displicente concerniente a las interrupciones, reloj, manejo de la memoria y otras características de bajo nivel.

En cada caso, la abstracción que presenta al usuario el Sistema Operativo es más simple y sencilla de utilizar que la programación del hardware implícito. En esta visión, la función del Sistema Operativo es la de presentar al usuario una máquina virtual o máquina ampliada que sea más fácil de programar que la programación del hardware implícito.

**EL SISTEMA OPERATIVO COMO
MANEJADOR DE RECURSOS**

Una visión alternativa sostiene que el Sistema Operativo se encuentra ahí para controlar todas las piezas de un sistema de computación. Las computadoras, como ya se mencionó, constan de procesadores, memoria, reloj, discos, unidades de cinta magnéticas, impresoras y una amplia variedad de dispositivos adicionales.

En la visión alternativa, el trabajo del Sistema Operativo consiste en ofrecer una distribución ordenada de los procesadores, memoria y dispositivos de entrada/salida, entre los diversos programas que compiten por la atención de estos.

Imagínese que sucedería, si tres programas que se ejecutan en alguna computadora tratan de imprimir su información de salida en forma simultánea en la misma impresora. Las primeras líneas de la impresión podrían ser del programa uno, las siguientes del programa dos, después algunas del programa tres, etc. El resultado sería el caos.

El Sistema Operativo, puede poner orden al caos potencial, separando todas las salidas destinadas a la impresora. Cuando un programa termina su ejecución, el Sistema Operativo puede copiar entonces los resultados a un archivo del disco donde se almacenará para ser impreso, mientras que al mismo tiempo el otro programa puede seguir ejecutándose, sin pensar en el hecho de que su información de salida en realidad no se dirige (todavía) a la impresora.

En resumen, esta visión del Sistema Operativo sostiene que su tarea principal consiste en llevar el control de quien utiliza cual recurso, conceder requisiciones de recursos, interceder en peticiones conflictivas de diferentes programas y usuarios.

Partiendo de los enfoques anteriores; el que nos interesa, para el presente trabajo, es el Sistema Operativo como manejador de recursos debido a que este se puede aplicar al Sistema Operativo MS-DOS, como se verá en el capítulo 2. Como parte final de este capítulo vamos a describir el Sistema Operativo como manejador de recursos y de una manera general los módulos de los cuales se compone dicho Sistema.

**DESCRIPCIÓN DEL SISTEMA
OPERATIVO COMO MANEJADOR DE
RECURSOS**

El software de la computadora puede dividirse de modo general en dos clases:

- Los programas del sistema.
Manejan la operación de la computadora misma

- Los programas de aplicación.
Resuelven los problemas del usuario

De los programas del sistema los programas más importantes son los programas que integran el Sistema Operativo.

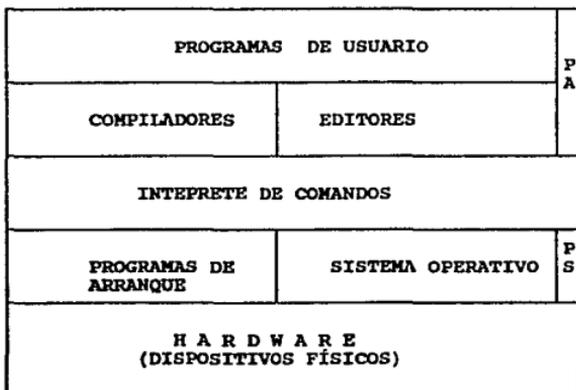


Figura -1.4-

Donde: PA -> PROGRAMAS DE APLICACIÓN
 PS -> PROGRAMAS DEL SISTEMA

El Sistema Operativo se describe como un conjunto de módulos integrados por programas que se encargan de administrar los dispositivos que integran la computadora, cada módulo puede llamar a cualquiera de los otros módulos siempre que lo necesite.

Este conjunto de módulos se encarga del control de los periféricos, el cargado de programas, chequeo de errores, administración de la memoria, etc. de la computadora.

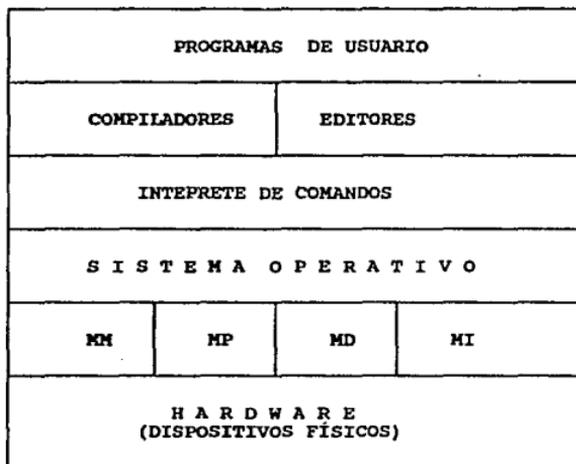


Figura -1.5-

Donde: MM -> MANEJO DE MEMORIA
MP -> MANEJO DEL MICROPROCESADOR
MD -> MANEJADOR DE DISPOSITIVOS
MI -> MANEJADOR DE INFORMACIÓN

Como se ve, el Sistema Operativo actúa como interfase entre el usuario y el hardware de la computadora, pero a su vez actúa como un administrador de los dispositivos de la computadora.

De manera general los módulos que componen el Sistema Operativo son:

- Administrador de memoria (Memory manager)
- Administrador del microprocesador (Procesor manager)
- Administrador de dispositivos (Device manager)
- Administrador de información (Information manager)

**DESCRIPCIÓN DE LAS COMPONENTES
DEL SISTEMA OPERATIVO.**

ADMINISTRADOR DE MEMORIA (memory manager).

Es la parte del Sistema Operativo que se encarga de la manipulación de la memoria primaria (memoria que el microprocesador accesa para instrucciones y datos).

Su labor es la de llevar el control de que partes de la memoria están en uso y no lo están, asignar memoria a procesos cuando la necesitan y liberarla cuando terminan su trabajo.

Algunas de las funciones que realiza el administrador de memoria son las siguientes:

- Conoce el estado de cada segmento de memoria, si el segmento esta asignado o no esta asignado.

- Determina la asignación de memoria, decide a quien debe asignarsele memoria, cuando y donde se debe asignar la memoria.
- Una vez que se asignó la memoria, se debe actualizar la información acerca del segmento de memoria que se asignó.
- Recupera el control de la memoria asignada cuando se ha concluido con el proceso.

ADMINISTRADOR DEL PROCESADOR (procesor manager).

El administrador del procesador se relaciona con la manipulación del microprocesador, específicamente, con la asignación del microprocesador a los procesos.

Este módulo se divide en las siguientes partes:

1.-PROGRAMADOR DE TAREAS (Job scheduler).

Conoce el estado de todos los trabajos que se realizan dentro de la computadora. También conoce que trabajo esta intentando conseguir el servicio del Microprocesador (estado de espera -hold state-) y que trabajos han sido atendidos por el microprocesador (Listo - ready-, Corriendo -running- o Estado bloqueado -blocked state-).

- Escoge que trabajo estará en el sistema, de estado de espera a estado listo.

Esta decisión se da en base a las siguientes características:

- Fuente y Prioridad que lo solicita.
- Asigna los dispositivos necesarios para que el trabajo pueda usar la memoria, administrador procesos, etc.
- Libera los dispositivos cuando el trabajo se ha realizado.

2- PROGRAMADOR DE PROCESOS (Process scheduler).

Una vez que el Programador de tareas ha cambiado de estado de espera a estado listo, este crea uno o más procesos para este trabajo.

El Programador de procesos decide que proceso consigue la atención del microprocesador, cuando y por cuanto tiempo.

Algunas funciones que realiza son:

- Conoce el estado de cada proceso (corriendo, listo o bloqueado). El módulo que realiza esta función, ha sido llamado controlador de tráfico (traffic controller).

- Decide que procesos consigue la atención del microprocesador y por cuanto tiempo. Esta tarea la realiza el Programador de procesos.

- Asigna el microprocesador a un proceso. Si se requiere una restauración (reset) de los registros del microprocesador para el estado correcto del proceso. Esta tarea la realiza el Controlador de tráfico.

- Libera al microprocesador, cuando el proceso excede su tiempo de trabajo asignado.
Esto requiere que todos los registros del microprocesador sean salvados para una futura reasignación. Esta tarea la realiza el controlador de tráfico.

- Conoce el estado de cada proceso (corriendo, listo o bloqueado). El módulo que realiza esta función, ha sido llamado controlador de tráfico (traffic controller).

- Decide que procesos consigua la atención del microprocesador y por cuanto tiempo. Esta tarea la realiza el Programador de procesos.

- Asigna el microprocesador a un proceso. Si se requiere una restauración (reset) de los registros del microprocesador para el estado correcto del proceso. Esta tarea la realiza el Controlador de tráfico.

- Libera al microprocesador, cuando el proceso excede su tiempo de trabajo asignado.
Esto requiere que todos los registros del microprocesador sean salvados para una futura reasignación. Esta tarea la realiza el controlador de tráfico.

3- CONTROLADOR DE TRÁFICO.

Conoce el estado del proceso.

Realiza las siguientes funciones:

- Conoce el estado de todos los procesos.
- Suministra el mecanismo para el cambio de estado de los procesos.
- Coordina la comunicación entre los procesos.

4- ADMINISTRADOR DE DISPOSITIVOS (Device manager)

Una de las funciones principales del administrador de dispositivos consiste en controlar todos los dispositivos de entrada / salida de la computadora.

Este debe emitir comandos a los dispositivos, capturar interrupciones y manejar errores, también debe proporcionar una interfase entre los dispositivos y el resto del sistema que sea simple y fácil de utilizar.

Las unidades de entrada / salida por lo general constan de una componente mecánica y una electrónica.

La componente electrónica se denomina controlador del dispositivo o adaptador, con frecuencia, es una tarjeta de circuitos impresos que se puede insertar en la computadora.

La componente mecánica es el dispositivo mismo.

Mencionamos esta distinción entre el controlador y el dispositivo porque el manejador de dispositivos trata siempre con el controlador, no con el dispositivo.

Las funciones básicas del administrador de dispositivos son:

- Conoce el estado de todos los dispositivos.

Para conocer el estado de los dispositivos se requiere de un mecanismo especial.

Uno comúnmente usado es tener una base de datos (data base) como un bloque de control de la unidad (Unit Control Block -UCB-) asociado con cada dispositivo.

- Decide quien debe ser atendido por el dispositivo, por cuanto tiempo y cuando.

Las técnicas que utiliza el administrador de dispositivos, son:

- a) Dispositivos dedicados (Dedicated device), estos son dispositivos que se dedican a un trabajo en particular.

- b) Dispositivos compartidos (Shared device), son los dispositivos que pueden ser compartidos por varios procesos.

 - c) Dispositivos virtuales (Virtual device), algunos dispositivos que normalmente deberían ser dedicados.
- Asigna a los dispositivos e inicia la operación de entrada / salida.

 - Recupera el control del dispositivo.

5- ADMINISTRADOR DE INFORMACIÓN (Information Management) .

El administrador de información se relaciona con el cargado de información y la devolución de la misma.

Las funciones del administrador de información son:

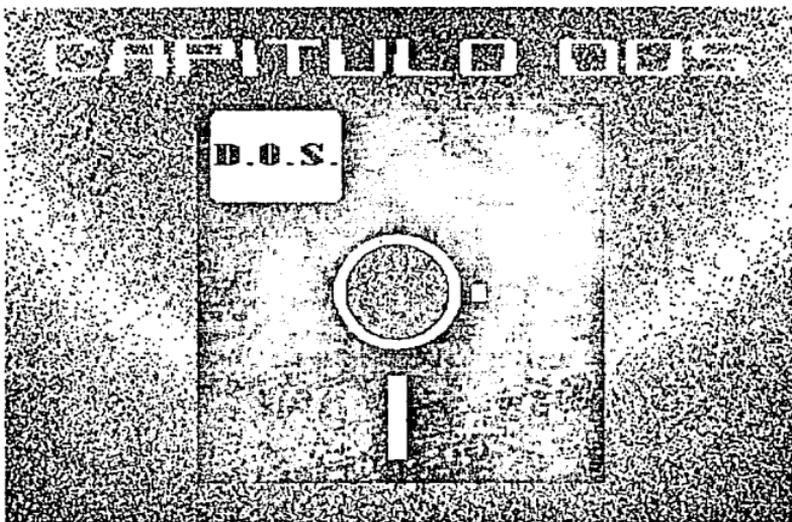
- Conoce el estado de toda la información del sistema por medio de varias tablas, la mayoría de ellas en un directorio.

- Determina donde y como la información es adquirida y quien tiene acceso a ella.

- Asigna la información.
Una vez que se ha decidido que proceso va a tener acceso a la información de una manera adecuada.

- Libera las fuentes de información.
Una vez que la información ya no es necesaria las fuentes de información son liberadas.

Podemos concluir que un Sistema Operativo es la parte intermediaria entre el hardware y el usuario, servirá para facilitar el uso de los dispositivos periféricos, tales como; el almacenamiento de información en discos, cintas, cassettes, etc. u otro medio de almacenamiento, también para ver lo que se esta tecleando como, gráficas, documentos, etc., tener una salida al exterior de la PC por medio de dispositivos de salida llamados puertos, ya sean paralelos o seriales para comunicación con otros dispositivos o interfases para controlar sistemas analógicos, la adquisición de datos para manejos estadísticos y la impresión de resultados.



COMPONENTES DEL MS-DOS

El MS-DOS (MicroSoft Disk Operating System) es un Sistema Operativo cargado desde disco, incluye programas de soporte y facilidades para el manejo del Sistema, ha sido liberado en diferentes versiones desde 1981.

En seguida se dará una lista de estas versiones.

VERSIÓN	FECHA	CARACTERÍSTICAS
PC-DOS 1.0	1981	Primer Sistema Operativo. Archivos orientados por pista (record-oriented files).
PC-DOS 1.1	1982	Soporta discos de doble cara.
MS-DOS 1.25	1982	Primer OEM liberado de MS-DOS
MS-DOS / PC-DOS 2.0	1983	Sistema Operativo para IBM PC/XT UNIX/XENIX sistema por archivos manejadores de dispositivos instalados Archivos orientados por bytes (Byte-oriented file) Soporta discos fijos.
PC-DOS 2.1	1983	Sistema Operativo para PCjr.
MS-DOS 2.11	1983	Soporte de Internacionalización

VERSIÓN	FECHA	CARACTERÍSTICAS
MS-DOS / PC-DOS 3.0	1984	Sistema Operativo para IBM PC/AT soporta discos flexibles de 1.2MB, también soporta discos fijos de grande capacidad. Soporte para archivos (record locking) aplicaciones de control de impresora
MS-DOS / PC-DOS 3.1	1984	Soporta redes
MS-DOS / PC-DOS 3.2	1986	Soporta disco flexible de 3.5 pulgadas discos formateados por pistas, adiciona manejadores de dispositivos.
MS-DOS / PC-DOS 3.3	1987	soporta IBM PS2 soporta particiones de disco con una capacidad de hasta 32MB.
MS-DOS / PC-DOS 4.0	1990	Soporta disco de alta capacidad mayores de 40MB. (versión no muy exitosa)
MS-DOS / PC-DOS 4.01	1990	soporte para manejo de la memoria extendida

VERSIÓN	FECHA	CARACTERÍSTICAS
MS-DOS / PC-DOS 5.0	1991	Chequeo completo de la memoria extendida, comandos del sistema con ayuda, mayor versatilidad para el manejo del Sistema Operativo

Figura -2.1-

El MS-DOS es un Sistema Operativo modular que consiste en múltiples componentes con funciones especiales.

Durante la inicialización del Sistema, estos módulos son cargados en memoria, en la parte baja de esta y estarán ahí hasta que se apague la computadora.

Estos módulos son un conjunto de programas supervisores que manipulan y controlan la computadora, en general, el Sistema Operativo MS-DOS suministra los módulos de los programas siguientes:

- A) El cargador del Sistema Operativo (The operating-system loader).
- B) Procesador de comandos (SHELL).

- C) MS-DOS kernel.
- Control de procesos (Process control).
 - Manejador de archivos (File manager).
 - Manejador de memoria (Memory management).
 - Soporte de periférico (Peripheral support).
- D) MS-DOS BIOS (Sistema básico de entrada/salida -Basic Input/Output System).

En seguida se explicará cada uno de los módulos del Sistema Operativo.

**CARGADOR DEL SISTEMA OPERATIVO
(Operating system loader).**

El cargador del Sistema Operativo se encarga de colocar al Sistema Operativo (desde un disco que contenga el Sistema), en memoria RAM. El proceso completo del cargado del Sistema Operativo es llamado **BOOTSTRAPING**, a menudo es muy complejo e involucra varios cargadores.

El MS-DOS es cargado como el resultado de una carga en frío (cold boot, encendido) o de una recarga (warm boot). Una carga en frío se realiza cuando la computadora es encendida por primera vez o cuando se realiza un apagado momentaneo (reset).

En el encendido por lo regular se realiza una auto prueba que (power-on self test -POST-), determina la cantidad de memoria disponible, determina los adaptadores de periféricos instalados, etc. El POST se reserva para el encendido ya que este toma cierta cantidad de tiempo (aproximadamente 10 segundos).

El reset, iniciado por la presión simultanea de las teclas Ctrl, Alt y Del, no prueba el hardware e inicia verificando si existe algún disco con el Sistema Operativo (bootable disk).

Todos los microprocesadores 8086/8088 y compatibles comienzan a trabajar ejecutando la instrucción que se encuentra en el CS:IP (FFFF:0000), que contiene una instrucción de salto que apunta al ROM-BIOS.

El ROM-BIOS contiene el código de inicialización de la computadora (esto nada tiene que ver con el MS-DOS, esta es una característica de los microprocesadores Intel).

Todas las PC's tienen memoria ROM que contiene los datos y programas necesarios para el arranque y operación de la computadora y sus periféricos. Una ventaja de tener ciertos programas cargados en memoria ROM es que estos están ahí y no necesitan ser cargados en memoria desde el disco en la manera que DOS debe ser cargado. En las computadoras compatibles con IBM, EL ROM-BIOS ocupa cierta cantidad de memoria ROM.

ROM_BIOS	FFFF:000Fh (1MB)
	FFFF:0000h
	F000:0000h
otra ROM y RAM	
RAM libre	A000:0000h (parte superior de RAM. IBM-PC)
	0000:0000h

Figura -2.2-

Las rutinas de inicialización del ROM-BIOS -POST- determinan que tipo de dispositivos están instalados y verifica la memoria convencional (conventional memory) para el primer Mega Byte.

Si existe algún problema en los dispositivos, este será reportado por medio de una serie de beeps o un mensaje en el monitor.

Hay cuatro clases diferentes de programas que están almacenados en la memoria ROM de la PC:

- Los programas de arranque (start-up).
Estos realizan el trabajo para el arranque de la computadora.

- Los programas de ROM-BIOS.
Este es un conjunto de rutinas escritas en lenguaje de máquina para la continua operación de la computadora.

- Los programas de ROM-BASIC.
Estos suministran la base para la programación en lenguaje BASIC.

- Los programas de ROM-extensión.
Son los programas adicionales que se suman a la memoria ROM cuando ciertos equipos opcionales son agregados a la computadora.

Antes de que el Sistema Operativo de la computadora pueda ser usado; el primer trabajo de los programas que residen en ROM es supervisar el arranque de la computadora.

Todos los dispositivos de la computadora deben ser probados e inicializados y el Sistema Operativo debe ser cargado en memoria RAM (bootstrap) desde algún drive. Los programas de prueba e inicialización de los dispositivos de la computadora están bajo el control de los programas del ROM start-up.

Y a la prueba se le conoce con el nombre de POST.

Entre las tareas que realizan las rutinas de arranque e inicialización (start-up) podemos mencionar algunas como:

- Que dispositivos están instalados.

- Inicializa los dispositivos de la computadora.

- Inicializa el vector de interrupciones.

Y por lo regular termina cargando el Sistema Operativo desde el disco.

Los programas de auto prueba e inicialización del POST pueden dividirse de la siguiente manera:

- Programas relacionados con el hardware central del sistema.
- Programas relacionados con la configuración y el hardware secundario o (non system board hardware) de la computadora.

Quando se presenta alguna falla en el hardware central el POST genera un código de error por medio de beeps. El POST prueba e inicializa primero el hardware central en siguiente orden:

- 1.- CPU.
 - 2.- ROM BIOS (suma de Bytes -checksum-).
 - 3.- CMOS RAM (solo el AT).
 - 4.- El controlador DMA (Direct Memory Acces) Intel 8237.
-

- 5.- El controlador del teclado.
- 6.- Los 64K de memoria RAM del sistema básico.
- 7.- Controlador de interrupciones programable (Programmable Interrupt Controller -PIC-).
- 8.- Temporizador de interrupciones programable (programmable interrupt timer -PIT-).

Una vez que el hardware central ha sido probado e inicializado, el POST verifica los datos de la configuración de sistema cargado en la memoria CMOS RAM.

Por último el POST prueba e inicializa el hardware secundario (nonsystem hardware) en el siguiente orden:

- Los datos de la configuración en CMOS RAM.
- Los controladores CTR.
- La memoria RAM después de los 64k.
- El teclado.
- La disponibilidad del drive A.
- Los circuitos de la interfase serie.
- El controlador del diskette.
- El controlador del disco duro.
- El hardware adicional.

Una vez que la computadora se encuentra lista para trabajar, el ROM-BIOS se prepara para trabajar en forma normal. Primero inicializa el vector de interrupciones al comienzo de la memoria y cualquier controlador que trabaje con el vector de interrupciones. El vector de interrupciones se encuentra localizado en las direcciones 0000:0000h a 0000:03FFh.

ROM-BIOS	FFFF:000Fh (1 MB) FFFF:0000h F000:0000h
otra ROM y RAM	
RAM libre	A000:0000h (Parte superior IBM-PC) 0000:0600h
Tablas del ROM BIOS	0000:0400h
Vector de interrupciones.	0000:0000h

Figura -2.3-

En computadoras IBM y compatibles, la memoria de 0000:0400h a 0000:0600h lo ocupan las tablas y buffers para las rutinas del ROM BIOS, luego el ROM-BIOS prepara cualquier interfase como, el controlador de acceso directo a memoria (DMA), los puertos serie, etc. y por último el ROM-BIOS inicia el cargado del Sistema Operativo (bootstrap) ejecutando el programa cargador contenido en la memoria ROM.

En la IBM PC, esta rutina verifica el primer drive del floppy "A" para ver si existe algún disco con el Sistema Operativo, si no lo hay, esta misma rutina verifica el drive "C" si tiene la máquina un disco fijo.

Este proceso se repite hasta que se encuentre un disco con el Sistema Operativo o hasta que estos drives sean verificados sin éxito.

Esto causa que se despliegue un mensaje indicando que el disco no contiene el Sistema Operativo. El disco con el Sistema Operativo puede ser detectado por cierto número de propiedades.

El cargador del ROM el primer sector del disco en memoria RAM y verifica si existe un salto corto o largo al principio del sector, ejecutando este salto llega a una rutina que correrá y esta le indicará si tiene el Sistema Operativo el diskette y buscará la última palabra del sector que debe contener el valor AA55H. Esta característica indica que el diskette contiene el cargador del Sistema Operativo.

Si estos datos no existen, el cargador del ROM despliega un mensaje indicando que el disco no contiene el Sistema Operativo.

Cuando un disco contiene el Sistema Operativo, el cargador del ROM carga en memoria RAM el primer sector del disco que contiene otro cargador y transfiere el control a este.

ROM BIOS	FFFF:000Fh FFFF:0000h F000:0000h
otra ROM y RAM	
posible RAM libre	A000:0000H (parte superior IBM-PC)
sector BOOT	localización arbitraria
RAM libre	0000:0600h
Tablas del ROM BIOS	0000:0400h
Vector de interrupciones	0000:0000h

Figura -2.4-

El cargador, contenido en memoria RAM, carga la siguiente parte del Sistema Operativo en la parte baja de la memoria.

Pero antes de que éste pueda proceder a cargar la siguiente parte del Sistema Operativo debe saber algunas cosas sobre el disco que contiene el Sistema Operativo. Esta información se encuentra dentro de una estructura de datos llamada BPB (BIOS parameter block) que se localiza, también, dentro del primer sector que contiene el Sistema Operativo.

Por medio de la información contenida dentro del BPB el programa cargador calcula la localización del directorio raíz dentro del disco que contiene el Sistema Operativo, entonces puede verificar que los dos primeros archivos, dentro del directorio raíz, sean IO.SYS y MSDOS.SYS.

El siguiente paso que realiza el cargador es leer los sectores que contienen los archivos IO.SYS y MSDOS.SYS en memoria RAM arriba del vector de interrupciones.

ROM BIOS	FFFF:000Fh FFFF:0000h F000:0000h
otras ROM y RAM	
Posible RAM libre	A000:0000h Parte superior de la RAM IBM-PC
Sector BOOT	Localización arbitraria
RAM libre	

MS-DOS.SYS	
IO.SYS	SYSINIT MS-DOS Residentes 0000:0600h
Tablas del ROM BIOS	 0000:0400h
Vector de Interrupciones	 0000:0000h

Figura -2.5-

Finalmente, suponiendo que el archivo fue cargado sin errores, el cargador del Sistema Operativo transfiere el control al IO.SYS. Como ya no se necesita el cargador, la memoria RAM que ocupaba se libera.

**SISTEMA DE INICIALIZACION DEL
MS-DOS (SYSINIT)**

El sistema de inicialización de MS-DOS comienza después que ha sido cargado el archivo IO.SYS y MSDOS.SYS y se ha transferido el control a IO.SYS.

El archivo IO.SYS se divide en tres módulos diferentes:

- Los manejadores de dispositivos residentes.
- El módulo básico de inicialización MS-DOS BIOS.
- El módulo de inicialización de MS-DOS.

Los dos módulos de inicialización son desechados tan pronto como MS-DOS esta completamente inicializado y el interprete de comandos esta corriendo; los manejadores de dispositivos residentes permanecen en memoria mientras MS-DOS este corriendo es por eso que los manejadores de dispositivos están localizados al principio del archivo IO.SYS, antes de los módulos de inicialización.

El módulo de inicialización BIOS por lo regular despliega un mensaje con el nombre de la compañía que fabrico la computadora (Original Equipment Manufacturer -OEM-).

Después examina las entradas en la tabla de interrupciones (interrupt table) para determinar que dispositivos se encontraron al correr el POST y ajusta la lista de los manejadores de dispositivos residentes. Este ajuste por lo regular remueve los drivers que no corresponden al hardware instalado.

Hasta aquí la rutina de inicialización ,MS-DOS BIOS, ha terminado su trabajo y el control es transferido al módulo SYSINIT.

El SYSINIT localiza el extremo superior de la memoria RAM y el mismo se copia ahí, transfiriéndole el control y procede con la inicialización del sistema. El primer paso es mover el archivo MSDOS.SYS, que contiene el MS-DOS Kernel, inmediatamente arriba de la parte residente del archivo IO.SYS, que contiene los manejadores de dispositivos residentes. Este movimiento escribe sobre la copia original de SYSINIT y en la rutina de inicialización MS-DOS BIOS, que ya no se necesitan más.

ROM BIOS	FFFF:000Fh F000:000h
otras ROM y RAM	
SYSINIT	A0000:0000h Parte superior de la RAM
RAM libre	
MS-DOS Kernell (MSDOS.SYS)	
MS-BIOS (IO.SYS)	Manejadores de dispositivos 0000:0600h

Tablas del ROM BIOS	0000:0400h
Vector de interrupciones	0000:0000h

Figura -2.6-

SYSINIT llama a la rutina de inicialización del MS-DOS Kernel, ahora relocalizada. Esta rutina inicializa el Kernel.

Después la rutina de inicialización de MS-DOS Kernel inicializa las funciones de los manejadores de dispositivos residentes y prepara los vectores para cualquier interrupción externa de hardware usada por los dispositivos. Cada bloque manejador de dispositivos regresa un apuntador al BPB por cada drive que este soporta; estos BPBs son inspeccionados por SYSINIT para encontrar el mayor sector usado por cualquier driver.

La rutina de inicialización del Kernel asigna un buffer de sectores de mayor tamaño encontrado y el manejador de dispositivos NUL al frente (head) de la lista de los manejadores de dispositivos.

La última operación de la rutina de inicialización del MS-DOS Kernel es desplegar un mensaje de copyright MS-DOS. Por último se regresa el control al SYSINIT que ya puede utilizar cualquier función de MS-DOS en conjunto con los manejadores de dispositivos residentes.

El siguiente paso de SYSINIT es intentar abrir el archivo CONFIG.SYS que se encuentra en el directorio raíz dentro del disco que contiene el Sistema Operativo (boot drive). Si el archivo no existe, SYSINIT usa los parámetros del sistema por default; si el archivo existe, SYSINIT lee el archivo en la parte alta de la memoria y convierte todos los caracteres a mayúsculas.

El contenido del archivo se procesa para determinar el número de buffers del disco, el número de entradas dentro de la tabla de archivos (file tables) y el número de entradas dentro de la tabla de traducción del drive (drive translation table); esta estructura es asignada enseguida del MS-DOS Kernel.

ROM BIOS	FFFF:000Fh (1 MB) F000:0000h
otra ROM y RAM	
SYSINIT	A000:0000h Parte superior de la RAM. IBM-PC
RAM libre	
Manejadores de dispositivos	
Bloque de control de archivos	
Buffer de disco	
Tablas del MS-DOS	
MS-DOS Kernel (MSDOS.SYS)	

MS-DOS BIOS (IO.SYS)	0000:0600h
Tablas del ROM BIOS	0000:0400h
Vector de interrupciones	0000:0000h

Figura -2.7-

SYSINIT procesa de una manera secuencial el texto del archivo CONFIG.SYS para determinar que manejadores de dispositivos se van a instalar y cargar el archivo que contiene el manejador de dispositivos instalados en memoria, después de los buffers del sistema de disco (system disk buffers).

La función de inicialización de cada manejador de dispositivos instalados en memoria es llamada después que el archivo del manejador de dispositivos es cargado en memoria. La rutina de inicialización es la misma que para los manejadores de dispositivos residentes, excepto en que el SYSINIT usa una dirección que el manejador de dispositivos envió para determinar en donde se encuentra el siguiente manejador de dispositivos.

Como en los manejadores de dispositivos residentes, los manejadores de dispositivos instalados en memoria pueden ser descartados por SYSINIT si la rutina de inicialización del manejador de dispositivos determina que el dispositivo no existe o es inoperable.

Un manejador de dispositivos descartado no se incluye en la lista de los manejadores de dispositivos.

SYSINIT ahora cierra todos los archivos y luego abre los tres dispositivos de caracteres CON, PRN y AUX.

**INICIALIZANDO EL PROCESADOR
DE COMANDOS (SHELL)**

La última función de SYSINIT es cargar y ejecutar el programa del SHELL usando la función EXEC de MS-DOS. Dentro del archivo CONFIG.SYS se encuentra el nombre del programa SHELL y sus parámetros de inicialización, el SHELL por default de MS-DOS es COMMAND.COM.

El programa del SHELL es cargado después de los manejadores de dispositivos instalados en memoria o después del bloque de control de archivos (FCB -File Control Block-) internos del MS-DOS.

ROM BIOS	FFFF:000Fh (1MB)
otra ROM Y RAM	
SYSINIT	A000:0000h Parte superior de la RAM
RAM libre	
COMMAND.COM (Parte transitoria)	
COMMAND.COM (inicialización)	
COMMAND.COM (residente)	

manejadores de dispositivos	
Bloque de control de archivos	
Buffer de disco	
Tablas de MS-DOS	
MS-DOS Kernel (MSDOS.SYS)	
MS-DOS BIOS (IO.SYS)	0000:0600h
Tablas del ROM BIOS	
Vector de interrupciones	0000:0000h

Figura -2.8-

EL PROCESADOR DE COMANDOS (SHELL)

El COMMAND.COM se divide en tres partes:

- Una parte residente.
- Un módulo de inicialización.
- Una parte transitoria.

La parte residente contiene el soporte para la terminación de programas iniciados por el COMMAND.COM y manda los mensajes de error (critical-error). Es también responsable del cargado de la parte transitoria cuando es necesario.

El módulo de inicialización es llamado una sola vez por la parte residente.

El módulo de inicialización primero mueve a la parte transitoria a la parte de la memoria más alta.

ROM BIOS	FFFF:000Fh (1MB)
otra ROM Y RAM	
COMMAND.COM (Transitoria)	A000:00000h Parte superior de la RAM
RAM libre	
COMMAND.COM (Residente)	
Manejadores de dispositivos	
Bloque de control de archivos	
Buffer de disco	
Tablas de MS-DOS	
MS-DOS Kernel (MSDOS.SYS)	
MS-DOS BIOS (IO.SYS)	0000:0600h
Tablas del ROM BIOS	
Vector de interrupciones	0000:0000h

Figura -2.9-

La parte transitoria procesa los parámetros para el SHELL que se encuentran dentro del archivo CONFIG.SYS. Luego procesa el archivo AUTOEXEC.BAT y por último regresa el control a la parte residente, que libera la memoria utilizada por el módulo de inicialización y la parte transitoria. La nueva parte transitoria, queda en espera de algún comando.

El procesador de comandos o SHELL, es la interfase del usuario con el Sistema Operativo y es responsable de la transferencia de comandos al Sistema Operativo, del cargado y la ejecución de programas. El procesador de comandos del Sistema Operativo esta por default en el archivo COMMAND.COM.

La parte transitoria acepta los comandos del teclado o de los archivos BATCH y los ejecuta. Los comandos se dividen en categorías:

- Comandos internos.
- Comandos externos.
- Archivos BATCH.

Los comandos internos son rutinas contenidas dentro del COMMAND.COM e incluyen operaciones como COPY o ERASE. La ejecución de un comando interno no sobrescribe la parte transitoria.

Los archivos BATCH son archivos de texto que contienen comandos internos, comandos externos y comentarios no ejecutables.

Los comandos externos son programas ejecutables y tienen la extensión .COM o .EXE.

Estos programas son llamados por el nombre del archivo sin extensión.

Los comandos externos son cargados por el COMMAND.COM por medio de la función EXEC.

Esta función carga el programa dentro del área transitoria de programas (Transient Program Area -TPA-), y luego pasa el control al programa. El control regresa al COMMAND.COM cuando el nuevo programa a terminado que libera la memoria inmediatamente.

Después que el programa ha terminado, la parte residente del COMMAND.COM verifica que la parte transitoria sea aún valida, por que si el programa fue demasiado grande, este pudo haber sobre escrito la parte transitoria del COMMAND.COM. Si la parte transitoria fue sobrescrita la parte residente carga una nueva copia de la parte transitoria.

El COMMAND.COM interpreta los comandos verificando si el programa especificado es algún comando interno, si no lo es, busca en los archivos con extensión COM, EXE, y BAT (en ese orden) y lo ejecuta.

MS-DOS: KERNEL

Es el corazón del Sistema Operativo MS-DOS y suministra las funciones básicas encontradas en un Sistema Operativo tradicional. Se encuentra contenido en el archivo MSDOS.SYS, este archivo tiene los atributos de oculto, del sistema y solo lectura.

Suministra una colección de funciones de soporte a los programas de aplicación escritos de una manera independiente del hardware; a estas funciones de soporte se les llama funciones del sistema (system function).

El MS-DOS kernel proporciona los siguientes servicios :

- Manejador de memoria (Memory manager).
- Manejador de archivo (File manager).
- Control de procesos (Process control).
- Soporte de periféricos (Peripheral support).

**MANEJADOR DE MEMORIA
(MEMORY MANAGER)**

Uno de los trabajos del Sistema Operativo consiste en manipular la memoria de la computadora y debe ser capaz de conocer el estado de esta. Existen tres posibles solicitudes que un Sistema Operativo debe de realizar, al manipular la memoria:

- Solicitud de asignación de un bloque de memoria.
- Solicitud para modificar el tamaño del bloque de memoria asignada.
- Solicitud para liberar el bloque de memoria asignado.

MS-DOS realiza estas operaciones sobre el TPA por medio de la interrupción 21h con las siguientes funciones:

- Función 48H (asignación de bloque de memoria).
- Función 49H (liberación del bloque de memoria).
- Función 4AH (modificación del bloque de memoria).

Las computadoras personales compatibles con MS-DOS tienen tres tipos de memoria RAM:

- Memoria convencional.
- Memoria de expansión.
- Memoria de extensión.

MEMORIA CONVENCIONAL

En la PC IBM y compatibles, MS-DOS y los programas que corren bajo su control ocupan la parte de la memoria convencional (de 640k hacia abajo) y se le conoce con el nombre de memoria convencional. La parte superior de la memoria (de 640k hacia arriba) lo ocupa la memoria ROM del sistema que contiene los programas de prueba, manejadores de dispositivos y la memoria de video. La memoria convencional suministrada por MS-DOS (de 640k hacia abajo) se divide en tres partes:

- La tabla del vector de interrupciones.
- El área del Sistema Operativo.
- El área de los programas transitorios o TPA.

La tabla de los vectores de interrupciones ocupa la parte más baja de la memoria, 1024 bytes de memoria, de la localidad 00000-003FFh; estas direcciones son específicas del microprocesador y no pueden ser cambiadas. Cada palabra doble es una posición en la tabla que llama a un vector de interrupción, y nos proporciona el segmento y el corrimiento (offset) de un manejador de interrupciones para el hardware asociado y el software .

Los manejadores de interrupciones se construyen dentro de el Sistema Operativo.

El área del Sistema Operativo empieza arriba de la tabla de vectores de interrupciones y almacena el Sistema Operativo, sus tablas, buffers, cualquier dispositivo adicional especificado en el CONFIG.SYS y la parte residente del COMMAND.COM.

La cantidad de memoria ocupada varía con la versión del Sistema Operativo MS-DOS.

El control de la memoria por MS-DOS comienza cuando DOS es cargado.

El TPA es llamado también memoria ARENA. La memoria TPA es dinámica y asignada en bloques. Todos los bloques de memoria, asignados o no asignados, comienzan con un bloque de control de memoria, o MCB (memory control block). Estos bloques de control, figura 2.3, identifican el tipo, tamaño del bloque de memoria y el programa o proceso al que pertenece.

Si el primer byte es 4DH el MCB es parte de la cadena que liga a todos los MCB o si el primer byte es 5Ah, el MCB es el final de la cadena.

Dirección (address)	tipo (type)	propietario (owner)	tamaño (size)
0A00:0 -->	4D	0008	1600
0A00:1 -->	Bloque de memoria controlado por MS-DOS		
2001:0 -->	4D	2013	0010
2002:0 -->	Bloque asignado al proceso 2013		
2012:0 -->	4D	2013	0500
2013:0 -->	Bloque asignado al proceso 2013		
2513:0 -->	5A	0000	7AEC
2514:0 -->	Bloque de memoria libre perteneciente a MS-DOS. Contiene el resto de memoria hasta la dirección más alta.		
9FFF:F -->			

Figura -2.10-

El tipo de proceso al que pertenece es cargado en el primer byte del MCB.

Los siguientes dos bytes en el MCB identifican al programa al que pertenece el identificador del proceso (Process Identifier -PID-) del bloque de memoria, que es idéntico a la dirección del segmento en donde se encuentra el prefijo del segmento del programa (Program Segment Prefix -PSP-).

El valor cero indica que el bloque de memoria no está asignado, o libre. Si no es cero, indica que el bloque está asignado, luego esta palabra contiene PID. El cuarto y quinto byte en el MCB es una palabra que contiene el tamaño del bloque de memoria siguiente. Este tamaño está expresado en párrafos (paragraphs) y no incluye el tamaño del MCB.

Los siguientes once bytes de el MCB son indefinidos y cada MCB es seguido por el bloque de memoria que controla.

Aunque la estructura de los MCB descritos arriba no está oficialmente documentada, hay una manera (no documentada) de acceder los MCB's, por medio de la interrupción 21h función 52H. Esta interrupción nos regresa un apuntador en los registros ES:BX; en la palabra apuntada por ES:[BX-2], es la dirección del segmento del primer MCB asignado a la cadena. Una vez que se ha encontrado el primer MCB de la cadena, es posible seguir todos los MCB's de la cadena.

Teniendo el primer MCB, la dirección del segmento del siguiente MCB en la cadena se localiza por la suma de la dirección del segmento del MCB corriente más el tamaño (en párrafos) del bloque de memoria más uno.

La cantidad de memoria que un programa va a utilizar depende de dos factores:

- El tipo de archivo desde el cual el programa es cargado.
- La cantidad de memoria disponible en el TPA.

Cuando el manejador de memoria del MS-DOS busca en los MCB de la cadena de bloques para satisfacer la solicitud de memoria, puede usar una de las siguientes estrategias:

- Usa el MCB con la dirección más baja que es lo suficientemente grande para satisfacer la solicitud de memoria.
- Usa el MCB con la menor cantidad de memoria sin importar su ubicación.
- Usa el MCB con la dirección más alta que es lo suficientemente grande para satisfacer la solicitud de memoria.

Se crea una nueva cabecera para la entrada a la memoria TPA que sobro del bloque de memoria asignado y puede ser usado para otras asignaciones.

MS-DOS asigna el bloque de memoria más grande del TPA a los programas cargados desde un archivo .COM.

Ya que los programas con extensión COM no contienen encabezado de archivo (file header) que pueda pasar información acerca de los segmentos o uso de la memoria a MS-DOS.

MS-DOS usa reglas más complicadas para asignar la memoria a los programas cargados desde un archivo con extensión EXE.

En este tipo de archivos existe un encabezado de archivo (file header) que contiene la información acerca de los requerimientos de la memoria del programa los cuales son pasados a MS-DOS.

MS-DOS inspecciona los MCB de la memoria TPA cuando cualquier función solicita un bloque de memoria o cuando es liberado el bloque de memoria.

Cuando el MS-DOS kernel recibe una solicitud para la asignación de un bloque de memoria, este inspecciona la cadena de cabeceras de memoria TPA para encontrar un bloque de memoria libre y poder satisfacer la solicitud.

**MANEJADOR DE ARCHIVO
(FILE MANAGER)**

El propósito de muchos programas de aplicación es leer, procesar y escribir datos sobre un disco magnético. Los cuales están organizados en archivos, que están identificados por nombres; los archivos pueden estar organizados en grupos y estos dentro de directorios.

El Sistema Operativo MS-DOS proporciona dos tipos diferentes de llamadas al Sistema Operativo para la manipulación de archivos sin desprestigiar las características hardware del dispositivo de disco (disk device). Así las aplicaciones se concentran en la forma y el contenido de los datos.

El servicio de almacenamiento en el disco (disk storage service) proporcionado por el Sistema Operativo se puede dividir en funciones de archivo y funciones de almacenamiento (file functions y record functions).

Las versiones de MS-DOS proporcionan dos conjuntos para manipular los archivos y el grabado de archivos:

- Bloque de control de archivos.
(File control block functions -FCB-).

- Funciones manejadoras de archivos.
(File handle functions).

EL FCB

Reposan sobre una estructura de datos llamada FCB para mantener cierta información acerca de los archivos que se abren. Esta estructura reside dentro de la memoria asignada a los programas de aplicación.

Las funciones FCB permiten al programador crear, abrir, cerrar, borrar, leer y escribir en cualquier posición dentro del archivo. Estas funciones no soportan una estructura jerárquica de archivo (estructura de árbol) que fue introducida en la versión de MS-DOS 2.0, entonces estas funciones solo pueden ser usadas para acceder archivos dentro de un subdirectorio común para un drive dado. Estas funciones ya no se usan más, debido a que se usaron en las primeras versiones de MS-DOS, por compatibilidad con el Sistema Operativo CP/M.

FUNCIONES MANEJADORAS DE ARCHIVOS.

Soportan una estructura jerárquica del archivo, permitiendo al programador crear, abrir, cerrar, borrar archivos dentro de cualquier subdirectorio sobre cualquier drive y leer o escribir dentro del archivo en cualquier posición.

La relación inicial entre el programa de aplicación y los datos cargados en el disco es el nombre del archivo que se designa por una cadena ASCII (una cadena -string- de caracteres ASCII terminados por un byte nulo o cero) que contiene el drive, trayectoria, nombre del archivo y extensión. Por ejemplo:

DRIVE:PATH\FILENAME.EXT

C:\SYSTEM\COMMAND.COM

Aparecerá dentro de la memoria como una secuencia de bytes:

```
43 3A 5C 53 59 53 54 45 4D 5C 43 4F 4D 4D 41
4E 44 2E 43 4F 4D 00
```

En donde el drive designa el disco en el cual el archivo reside, PATH, especifica el directorio en el cual se encuentra el archivo y FILENAME.EXT identifica el archivo mismo.

Todas las funciones manejadoras de archivos requieren que el nombre este en forma de una cadena (string) ASCII.

Cuando un programa desea abrir o crear un archivo, este pasa la dirección de la cadena ASCII especificando el archivo a MS-DOS dentro de los registros DS:DX.

Si la operación es exitosa, MS-DOS regresa un manejador (16-Bits) al programa dentro del registro AX. El programa debe salvar este manejador para futuras referencias.

Cuando el programa solicita una subsiguiente operación sobre el archivo, este pone al manejador dentro del registro BX antes de llamar a MS-DOS. Si la operación fue exitosa, todas las funciones manejadoras (handle funtion), limpian la bandera de acarreo (carry flag) dentro del CPU o si la operación fallo el registro AX contiene el código que describe la falla.

Las funciones manejadoras de archivos permiten al programador abrir o crear archivos pasando a MS-DOS una cadena de caracteres llamada nulo, terminado en cero que describe la localización del archivo dentro de la estructura jerárquica del archivo (el drive y la trayectoria), el nombre del archivo y su extensión.

MS-DOS suministra tres funciones manejadoras para crear archivos por medio de la interrupción 21h:

FUNCIÓN	NOMBRE
3CH	Crea un archivo con manejador (de la versión 2.0 en adelante).
5AH	Crea temporalmente un archivo (versión 3.0 en adelante)
5BH	Crea un nuevo archivo (versión 3.0 en adelante)

Figura -2.11-

Cada función es llamada con el segmento y el offset de ASCII en los registros DS:DX y el atributo asignado al archivo dentro de el registro CX, Los posibles atributos pueden ser:

CÓDIGO	ATRIBUTO
00H	Archivo normal
01H	Archivo de solo lectura
02H	Archivo oculto
04H	Archivo del sistema

Figura -2.12-

Para abrir un archivo ya existente se utiliza la interrupción 21h función 3Dh (abrir un archivo -open file with handle-) que abre un archivo normal, oculto o del sistema dentro del directorio especificado.

Cuando se llama esta función 3dh, el programa de aplicación suministra el apuntador del ASCII dentro de los registros DS:DX y 1-Byte de código de acceso dentro del registro al. Los bits del código de acceso son:

BITS	DESCRIPCIÓN
0-2	Permiso para leer / escribir (versiones 2.0 en adelante)
3	reservado
4-6	Modo (File-sharing) (versiones 3.0 en adelante)
7	Bandera de intervención (versión 3.0 en adelante)

Figura -2.13-

El código de permiso para leer/escribir toma los siguientes valores:

BITS 0-2	DESCRIPCIÓN
000	Se desea permiso para leer
001	Se desea permiso para escribir
010	Se desea permiso para leer y escribir

Figura -2.14-

En el modo de file-sharing, los valores siguientes son posibles:

BITS 4-6	DESCRIPCIÓN
000	Modo de compatibilidad. Otros programas de aplicación pueden abrir el archivo y realizar las operaciones de lectura y escritura.
001	Negado a todos. Otros programas de aplicación no pueden abrir el archivo.
010	Escritura negada. Otros programas de aplicación no pueden escribir sobre el archivo en modo de compatibilidad o con permiso de escritura
011	Lectura negada. Otros programas de aplicación no pueden leer el archivo en modo de compatibilidad o con permiso de lectura.

BITS 4-6	DESCRIPCIÓN
100	Nada negado. Otros programas de aplicación pueden abrir el archivo y realizar las operaciones de lectura y escritura pero no pueden abrir el archivo en modo de compatibilidad

Figura -2.15-

Para cerrar un archivo creado o abierto MS-DOS utiliza la interrupción 21h función 3Eh (close file). El programa de aplicación debe poner el manejador del archivo a cerrar en el registro BX. Si se realizó una operación de escritura sobre el archivo, MS-DOS actualiza la fecha, hora y tamaño del archivo dentro del directorio.

Otras operaciones que se pueden realizar con las funciones file handle por medio de la interrupción 21h son:

FUNCIÓN	DESCRIPCIÓN
42H	Mover el apuntador del archivo (move file pointer)
56H	Renombrar un archivo (rename file)
41H	Borrar el archivo
43H	Obtener / fijar el atributo del archivo (get / set file attribute).

FUNCIÓN	DESCRIPCIÓN
57H	Obtener / fijar la fecha y la hora del archivo (get / set date/time of file)

Figura -2.16-

Si la acción de crear o abrir es exitosa, MS-DOS regresa los manejadores, que son salvados por el programa de aplicación y usados para especificar el archivo en subsiguientes operaciones.

Si no es exitoso, MS-DOS activa la bandera de acarreo (carry flag) y regresa un código de error en el registro AX. El programa de aplicación debe verificar la bandera de acarreo (carry flag) después de cada operación y tomar las acciones apropiadas para el error. La tabla siguiente es una lista de los errores más frecuentes:

CÓDIGO	ERROR
02	Archivo no encontrado
03	Trayectoria no encontrada
04	Demasiados archivos abiertos
05	Acceso denegado
06	Manejador no valido
11	Formato no valido
12	Código de acceso no valido
13	Dato no valido

CÓDIGO	ERROR
15	Letra de la asignación del disco no valida
17	No es el mismo dispositivo
18	No más archivos

Figura -2.17-

Bajo la versión 3.X, el programa de aplicación puede usar la interrupción 21h función 59h (get extended error information) para obtener más detalles acerca de la causa del error. La información que regresa la interrupción 21h función 59h incluye el tipo de dispositivo que causo el error y recomienda la acción a seguir.

Cuando se usa la función manejadora de archivo (File Handle Functions), el Sistema Operativo mantiene una estructura de datos que contiene información acerca del archivo dentro de su propio espacio de memoria y esta estructura no la puede acceder el programa de aplicación.

El MS-DOS restringe el número de manejadores que pueden estar activos en cualquier momento, esto es, el número de archivos y dispositivos que pueden estar abiertos cuando se usa la función manejadora y son, de dos maneras:

- El número máximo de archivos abiertos dentro del sistema, para todos los procesos combinados activos, se especifica en:

FILES=nn

Dentro del archivo CONFIG.SYS. Esta entrada determina el número de entradas para ser asignada dentro de la tabla de archivos del sistema (system file table); bajo la versión 3.0 de MS-DOS el valor por default es 8 y el máximo es de 255.

- El número máximo de archivos abiertos para un único proceso es 20, suponiendo que hay suficientes entradas disponibles dentro de la tabla de archivos del sistema. Cuando se carga un programa, MS-DOS preasigna 5 de sus 20 manejadores para un dispositivo standard (device standard).

Cada vez que un proceso abre o crea una llamada al sistema, el MS-DOS asigna un manejador de los que restan de los 20, hasta que se agotan los manejadores de la tabla de archivos del sistema. Los primeros cinco manejadores son asignados a dispositivos de caracteres, que son por default CON, AUX y PRN:

MANEJADOR	SERVICIO	DEFAULT
0	Entrada standard	Teclado (keyboard)
1	Salida standard	Desplegado de video
2	Error standard	Desplegado de video
3	Auxiliar standard	Puerto serial (COMM 1)
4	Impresora standard	Puerto paralelo

Figura -2.18-

El MS-DOS tiene una tabla interna con la información de todos los archivos y dispositivos abiertos para todos los procesos activos. La secuencia del programa típico para acceder un archivo usando la familia de funciones manejadoras de archivo es la siguiente:

- 1- Obtener el nombre del archivo por medio del servicio de entrada (buffer input service Int. 21h función 0Ah) o del comando proporcionado por MS-DOS dentro del PSP (Program Segment Prefix).

- 2- Poner un cero al final del archivo especificado para crear una cadena ASCII.

- 3- Abrir el archivo usando la Int. 21h función 3Dh y en el modo 2 (read/write access), o crear el archivo usando la Int 21h function 3Ch; salvar el manejador.

- 4- Fijar el apuntador del archivo usando la Int 21h function 42h. Se puede fijar el apuntador del archivo en alguna de tres posiciones diferentes:
 - Al principio del archivo.
 - Una posición cualquiera del archivo.
 - Al final del archivo.

- 5- Leer el archivo (Int 21h function 3Fh) o escribir sobre el archivo (Int 21h function 40h). Ambas funciones requieren que el registro BX contenga el file handle.

- 6- Si el programa no ha terminado de procesar el archivo, ir al paso 4, de otra manera cerrar el archivo (Int 21h function 3Eh). Cualquier método de salida del programa también cerrara todos los manejadores activos.

**CONTROL DE PROCESOS
(PROCESS CONTROL)**

El control de procesos o tareas incluye el cargado de programas, la ejecución de tareas, terminación de tareas, programación de tareas (scheduling task) y la comunicación entre tareas.

Aunque MS-DOS no es un Sistema Operativo de multitareas (multitasking), este puede tener en memoria varios programas al mismo tiempo. Un programa puede invocar a otro, que se convertirá en el programa activo (tarea activa). Cuando la tarea que fue invocada a terminado, el programa invocado vuelve a convertirse en el programa activo. Estos programas nunca se ejecutan simultáneamente, es por eso que MS-DOS se considera un Sistema Operativo de una sola tarea (single tasking).

El método tradicional de comunicación entre programas incluye semáforos, colas (colas), memoria compartida y uniones (pipes). De estos MS-DOS formalmente soporta solo pipes.

Un pipe es una secuencia lógica, unidireccional de datos escrita por un programa y leída por otro.

Los datos del pipe residen en memoria o disco dependiendo de la implementación.

Los programas que corren bajo MS-DOS son de dos tipos:

- Los programas con extensión COM que pueden ocupar, como máximo 64KB.

- Los programas con extensión EXE que pueden ser tan grandes como la memoria disponible.

Ambos programas con extensión COM y EXE son cargados en memoria, para ser ejecutados, por el mismo mecanismo, la función EXEC, que constituye el cargador del MS-DOS. Esta puede ser llamada con el nombre del programa o del archivo para ser cargado por el COMMAND.COM.

Si hay suficiente memoria disponible en el TPA, EXEC asigna un bloque de memoria para retener el programa, crea el PSP en la base. Después lee el programa en la memoria inmediatamente arriba del PSP. Finalmente EXEC fija todos los registros de los segmentos, el stack y transfiere el control al programa. Al terminar el programa, EXEC recobra el control.

Los programas con extensión COM y EXE a menudo se les conoce como programas transitorios (transient programs).

A los programas transitorios "pertenece" el bloque de memoria que le ha sido asignado y tiene casi un control total sobre las fuentes del sistema mientras esta corriendo. Cuando el programa ha terminado, ya sea por que ha sido abortado por el Sistema Operativo o por que ha terminado su trabajo, sistemáticamente realiza su salida a MS-DOS, libera la memoria que le fue asignada (de aquí el termino transitorio) y puede ser usado por el siguiente programa por ser cargado.

**SOORTE DE PERIFERICOS
(PERIPHERAL SUPPORT)**

Una computadora se compone de un CPU, memoria y dispositivos periféricos (DISK DRIVES, DISPLAYS, IMPRESORA, MODEM, ETC.) que el CPU utiliza para adquirir datos o comunicarse con el mundo exterior.

El Sistema Operativo debe ser capaz de comunicar a un programa de aplicación con los dispositivos periféricos; primero debe proporcionar una interfase entre los programas de aplicación y el Sistema Operativo que debe ser lo suficientemente flexible para permitir al programa de aplicación especificar que desea del dispositivo. segundo, el Sistema Operativo debe tener la capacidad de poder transferir datos al dispositivo y controlar las operaciones del dispositivo.

Esta interfase entre el Sistema Operativo y los dispositivos es proporcionada por una parte del Sistema Operativo llamada Manejadores de dispositivos.

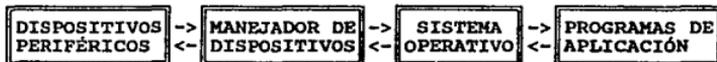


Figura -2.19-

Dentro de MS-DOS los dispositivos periféricos, esenciales, son:

- El teclado (para la entrada de información).
- El monitor (para la salida de información).
- Uno o más drives (disk drive) para el almacenamiento de información.

El MS-DOS reconoce dos tipos diferentes de periféricos, que son:

- Dispositivos de bloque (block device) que son, por lo regular discos flexibles y discos duros (para almacenar información).
- Dispositivos de carácter (character device) que son teclado (para la entrada), monitor (para la salida), impresora, puertos de comunicación, modems etc.

Un programa de aplicación puede usar dos técnicas diferentes para acceder un dispositivo de carácter :

- Por medio de las funciones manejadoras (handle-type function).

- Por el método tradicional, que mantiene como parte del sistema operativo por compatibilidad, puesto que las funciones handle son más poderosas y flexibles.

La interrupción 21h función 3fh lee un archivo o dispositivo (read file o device), permite a un programa de aplicación leer datos desde un archivo o un dispositivo que ha sido abierto con las funciones manejadoras.

El programa de aplicación es responsable de proporcionar el buffer de datos.

La interrupción 21h función 40h (write file or device) escribe desde un buffer a un archivo (o dispositivo) usando el manejador previamente obtenido al abrir o crear un archivo o dispositivo.

Durante la operación de crear o abrir, el MS-DOS busca en la cadena de dispositivos (device-driver chain) secuencialmente por el dispositivo de carácter con el nombre especificado antes de buscar en el directorio del disco.

MS-DOS soporta por lo menos los siguientes dispositivos de carácter, sin necesidad de instalarlo.

DISPOSITIVO	SIGNIFICADO
CON	Teclado y monitor
PRN	Impresora, usualmente puerto paralelo
AUX	Dispositivo auxiliar usualmente un puerto serie
CLOK	Fecha y hora
NUL	Dispositivo no existente

Figura -2.20-

TECLADO

Los programas de aplicación pueden enviar información por medio del teclado por diferentes métodos:

- Funciones manejadoras (MS-DOS handle-oriented functions).
- Funciones tradicionales de carácter.
- Las funciones IBM ROM BIOS de teclado.

Cuando un programa inicia su ejecución, MS-DOS suministra un manejador predefinido para ciertos dispositivos de carácter comúnmente usados, incluyendo el teclado.

Los manejadores que se muestran en la tabla pueden ser usados para leer o escribir sin más detalles.

MANEJADOR	NOMBRE DEL DISPOSITIVO	ABIERTO PARA
0	Entrada standard (stdin)	CON
1	Salida standard (stdout)	CON
2	Error standard (stderr)	CON
3	Auxiliar standard (stdaux)	AUX
4	Impresora standard (stdprn)	PRN

Figura -2.21-

Un programa puede obtener un manejador de un dispositivo de carácter explícitamente abriéndolo para entrada o salida usando su nombre lógico.

Las funciones manejadoras soportan la redirección de entrada/salida, permitiendo al programa tomar su entrada de algún otro dispositivo o archivo.

MONITOR

Los adaptadores para monitores, para las computadoras IBM y compatibles, tienen una interfase híbrida con el procesador central. Todas las características del monitor como la resolución horizontal y vertical, color son controlados por valores escritos en los puertos de entrada y salida cuya dirección esta unida físicamente al adaptador, mientras que la aparición del carácter en el monitor es controlado por localidades específicas dentro de la memoria llamada memoria de refresco (regen buffer o refresh buffer).

Ambos, CPU y controladores de video accesan esta memoria; el software actualiza el monitor simplemente escribiendo al código del carácter dentro del memoria de refresco.

El siguiente diagrama muestra la localización de las memorias de refresco para varios adaptadores dentro de la memoria.

DIRECCIÓN	ROM BIOS
FE000H	Sistema ROM, BASIC del sistema, etc.
F4000H	
C0000H	Reservado para extensiones de BIOS (discos duros, controladores, etc)
	Reservado

DIRECCIÓN	ROM BIOS
BC000H	16KB memoria de refresco para CGA, EGA, MCGA Y VGA en modo texto y 200 líneas modo gráfico
B8000H	
	Reservado
B1000H	4KB adaptador monocromatico memoria de refresco
B0000H	
	Memoria de refresco área para EGA, MCGA y VGA en 350 líneas o 480 líneas en modo gráfico
A0000H	Parte transitoria del COMMAND.COM
	Area transitoria de programas
00400H	MS-DOS, buffers, tablas y dispositivos manejadores
00000H	Vector de interrupciones

Figura -2.22-

Donde: CGA Adaptador gráfico de color (Color/Graphics Adapter)
 EGA Adaptador gráfico de alta resolución (Enhanced Graphic Adapter)
 MCGA Arreglo gráfico multi color (Multi-Color Graphics Array)
 VGA Arreglo gráfico de video (Video Graphics Array)

IMPREGORA

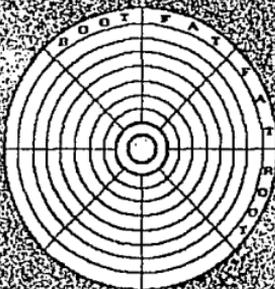
Se puede manipular la impresora y el puerto serie por diferentes métodos:

- Funciones orientadas al manejo (handle-oriented function).
- Funciones tradicionales de carácter.
- Las funciones IBM ROM BIOS de driver.

El método más usado para la manipulación de la impresora es el que utiliza la función manejadora (Int 21h función 40h) con su predefinido manejador.

Se puede escribir en algún otro dispositivo (por ejemplo LPT1 o LPT2) debemos abrir cada dispositivo (Int 21h función 3Dh) y usando el manejador que MS-DOS regresa para accesar el dispositivo individualmente (Int 21h función 40h).

CHAPTER 165



DESCRIPCIÓN FÍSICA DEL DISCO

Para almacenar información, las computadoras comúnmente usan discos flexibles (ya sean de 3.5 o de 5.25 pulgadas) y fijos (Con el término "fijo" nos referimos al hecho de que está instalado dentro de la computadora permanentemente y por lo tanto no puede ser movido de ahí).

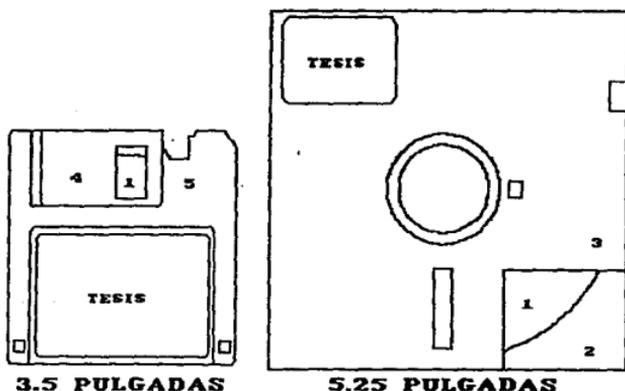


Figura -3.1-
Disco flexibles.

Los discos flexibles están compuestos de una pieza circular con capa magnética de plástico Mylar (1), que gira libremente en el interior de una cubierta muy ajustada, cuya finalidad es protegerlo (3).

Dentro de la cubierta hay un lubricante que ayuda al disco a girar libremente(2). En el caso de un disco de 3.5 pulgadas tiene una cubierta metálica(4) que sirve para proteger al disco y la cubierta es de plástico más resistente (5).

Un disco fijo también es referido como disco duro, ya que es de material metálico o de algún otro material rígido.

ORGANIZACIÓN DE DATOS EN UN DISCO

Cuando el sistema operativo MS-DOS escribe sobre un disco, lo hace de una manera muy ordenada. Escribe los datos en anillos circulares denominados tracks o pistas. Cada track esta dividido en secciones denominados sectores.

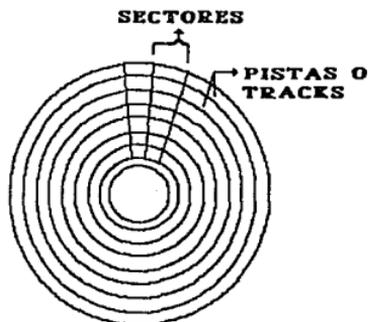


Figura -3.2-

Cada sector tiene un número único asignado. La estructura general que presenta un disco, es la misma ya sea para disco duro o flexible.

Para que MS-DOS pueda leer o escribir sobre un disco flexible se requiere que los tracks y los sectores estén definidos sobre el disco flexible.

MS-DOS proporciona el comando `FORMAT` para poder realizar esta tarea, cuando se aplica este comando pone una estructura a un disco flexible (floppy) o un disco duro, dividiendo el disco en tracks y sectores, poniendo en la cara cero, sector cero y pista cero un código para indicarle al sistema que ese disco está listo para cualquier acceso que desee hacer el sistema.

El mapeo del código de un disco es el siguiente:

No. DE SECTOR	DESCRIPCIÓN
0 A este sector se le llama BOOT	Identificación del fabricante - OEM- (Original equipment manufacturer) Conjunto de parámetros del BIOS -BPB- (Bios parameter block) Rutina cargadora del Sistema Operativo.
1... Depende del tamaño del disco	Tabla de asignación de archivos -FAT- (File allocation table)
...	Posible copia de la FAT
...	Directorio del disco (Root disk directory)
. Área de archivos . .

Figura -3.3-

SECTORES DE CONTROL Y DE DATOS

La información almacenada en un disco es accesada por medio de un sistema de soporte que es parte del MS-DOS kernel.

El MS-DOS kernel, por medio del manejador de dispositivos, ve un disco como un arreglo lógico de sectores de tamaño definido y asume que el arreglo contiene un sistema de archivo valido para MS-DOS. Cada unidad de disco flexible esta dividido en varias áreas de control y de archivo bien definidas, pero toda la información necesaria para interpretar la estructura de un disco se encuentra en el mismo disco en el sector de inicialización (BOOT).

El sector lógico 0, conocido como sector BOOT, contiene toda la información acerca de las características del disco.

EL BOOT

El boot incluye un pequeño programa ,que es usado cuando el sistema operativo es cargado desde el disco flexible, para indicar al ROM BIOS donde están los archivos del MS-DOS, y si están presentes, cargarlos y luego transferir el control a MS-DOS.

La descripción del sector boot es la siguiente:

DIRECCIÓN	DESCRIPCIÓN
00H	E9 XX XX o EB XX 90
03H	OEM, nombre y versión (8 Bytes)
0BH	Bytes por sector (2 Bytes)
0DH	Sectores por unidad de alojamiento (1 Byte)
0EH	Sectores reservados. Comenzando desde el cero (2 Bytes)
10H	Número de FATs (1 Byte)
11H	Número de entradas del directorio (2 Bytes)
13H	Sectores totales en el disco (2 Bytes)
15H	Byte que describe el tipo de disco (Media descriptor Byte)
16H	Número de sectores por FAT (2 Bytes)
18H	Sectores por track (2 Bytes)
1AH	Número de cabezas de lectura de disco (2 Bytes)
1CH	Número de sectores ocultos (2 Bytes)
1EH	Rutina Cargadora del sistema operativo

Figura -3.4-

La interpretación de la tabla es la siguiente:

- El primer byte en el sector es siempre el código de una instrucción de salto JMP (código de operación 0E9h) seguido por 16-bits de desplazamiento o un "short" JMP (código de operación 0EBh) seguido por 8-bits de desplazamiento y luego por un NOP (-no operation- código de operación 90h).
Si alguno de estos dos saltos no están presentes, el disco no ha sido formateado o no está formateado por MS-DOS.

- Después de la primer instrucción JMP están 8-bytes reservados por Microsoft para la identificación del OEM.

- El tercer componente del sector es el BPB.
Esta estructura de datos describe las características físicas del disco; y también contiene información usada por MS-DOS para calcular la dirección y el tamaño de cada área de control del disco (PAT y el directorio raíz).

- El elemento final del sector es la rutina de cargado del sistema operativo (disk bootstrap). Esta rutina es cargada en memoria RAM por la rutina ROM bootstrap, que se ejecuta automáticamente cuando la computadora es encendida.

El ROM bootstrap esta programado para mover la cabeza del manejador del disco al track 0, y leer el primer sector en una dirección predeterminada en la memoria RAM y saltar a esta. La rutina disk bootstrap es más sofisticada.

Esta calcula la dirección física de donde comienza el área de archivos, lee en memoria los archivos que contienen el sistema operativo, y transfiere el control al módulo BIOS en la localidad 0070:0000h.

LA FAT

La FAT suministra un mapa del disco, es llave para encontrar los clusters (Un cluster corresponde a un conjunto de sectores determinado por MS-DOS y se encuentra definido en el sector boot dentro del BPB) asignados a los archivos.

La FAT se hizo de tal manera que se dividió en campos, y cada campo corresponde a un específico cluster sobre el disco.

Esto significa que el espacio mínimo de memoria que puede ocupar un archivo, es un cluster.

Los archivos pueden ser divididos en partes para poder cargarlos en cualquier parte del disco.

El número de sectores por cluster varía de acuerdo a la versión del FORMAT. Entonces La FAT contiene campos que corresponden, en relación uno a uno, directamente a los clusters sobre el disco.

Vamos a suponer que un cluster contiene dos sectores (1024 bytes), si tenemos un archivo de 1025 bytes, este será cargado en dos cluster; aunque el segundo estará prácticamente vacío. Los clusters asignados a un archivo no están necesariamente en forma consecutiva.

Por ejemplo la asignación de clusters de varios archivos están como sigue:

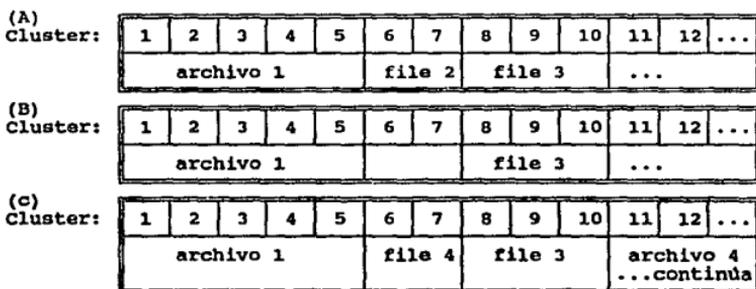


Figura -3.5-

El archivo 2 fue borrado (B) y si escribimos un archivo 4, un uso óptimo del espacio del disco usara el área de memoria que esta disponible. Aunque el archivo 4 sea demasiado grande, se podría dividir en partes como se muestra en la figura (C).

La FAT reconoce los cluster que pertenecen al archivo por medio del método de la cadena (chaining).

Como sabemos nosotros el primer cluster de la cadena?

Lo podemos obtener por medio del directorio. Después que hemos obtenido el primer cluster de la cadena, tenemos que encontrar el siguiente cluster de la cadena, para esto regresamos a la FAT.

El número que se encuentra en el campo de la FAT es el siguiente cluster que pertenece al archivo y así sucesivamente hasta llegar al final de la cadena.

CAMPOS DE LA FAT

La FAT se compone como ya se mencionó de campos. MS-DOS soporta dos tipos de campos (de 12 o 16 bits).

El campo de 12 bits lo podemos encontrar en la versión de MS-DOS 1 y 2, el campo de 16 bits fue introducido con la versión 3.0 del MS-DOS que se ajusta para discos duros con más de 4087 clusters. Siempre verifica el primer byte de la FAT para saber que tipo de disco se esta usando, este primer byte contienen una copia del descriptor del disco, que se encuentra también en el BPB dentro del sector boot.

VERSIÓN DEL MS-DOS	TIPO DE DISCO	CAP. DEL DISCO	DESCRIP. DEL DISCO	TAM. DEL DIR.	S/C	TAM. DE LA FAT
1, 2, 3 (menos 1.0)	5.25" SS 8 SECT.	160K	FF	4 SECT.	1	1 SECT.
2, 3	5.25" SS 9 SECT.	180K	FC	4 SECT.	1	1 SECT.
1, 2, 3	5.25" DS 8 SECT.	320K	FE	7 SECT.	2	2 SECT.
2, 3	5, 25" DS 9 SECT.	360K	FD	7 SECT.	2	2 SECT.
3	5.25" HD 15 SECT.	1.2M	F9	14 SECT.	1	7 SECT.
2, 3	DISCO FIJO -XT-	10M	F8	32 MAX.	8	8 SECT.
2, 3	DISCO FIJO -AT-	20M	F8	32 MAX.	4	40 SECT.

Figura -3.6-

El segundo, tercero (para un campo de 12 bits) y cuarto byte (para un campo de 16 bits) que constituyen el resto del área reservada, siempre contienen el valor FFh.

FAT de 12 BITS	FAT de 16 BITS
F9 FF FF	F8 FF FF FF

Figura -3.7-

El contenido de los campos de la FAT se interpretan de la siguiente manera:

12 BITS	16 BITS	DESCRIPCIÓN
000H	0000H	Cluster libre
001h	0001h	Código no usado
FF0H-FF6H	FFF0H-FFF6H	Reservado
FF7H	FFF7H	Cluster dañado (no puede ser usado)
FF8H-FFFH	FFF8H-FFFFH	Último cluster del archivo
XXXH (otros valores)	XXXXH (otros valores)	siguientes clusters de los archivos

Figura -3.8-

Cada campo del directorio contiene el número del primer cluster asignado al archivo, que se usa como punto de entrada a la FAT.

Desde este punto de entrada, cada campo de la FAT contiene el número del siguiente cluster asignado al archivo, hasta que se encuentre el último cluster asignado.

Veamos ahora como DOS carga un archivo de 3 073 bytes, suponiendo que estamos trabajando con campos de 12 bits.

A)

ENTRADA :	2	3	4	5	6	7	8	9	10	11	12	13
	3	4	5	FFF	0	0	FF7	0	0	0	0	0

(Cada entrada es de 1.5 Bytes de longitud)

B)

ENTRADA :	2	3	4	5	6	7	8	9	10	11	12	13
	3	4	5	FFF	7	0	FF7	0	0	0	0	0

C)

ENTRADA :	2	3	4	5	6	7	8	9	10	11	12	13
	3	4	5	FFF	7	9	FF7	10	0	0	0	0

D)

ENTRADA :	2	3	4	5	6	7	8	9	10	11	12	13
	3	4	5	FFF	7	9	FF7	10	FFF	0	0	0

Figura -3.9-

Si nuestra FAT aparece como en la figura -3.9 (A)-; DOS verifica los campos de la FAT hasta que encuentra un 0 en el cluster 6, que significa que el cluster está disponible, continua checando la FAT hasta que encuentra otro campo con el valor 0 en el cluster 7. DOS asigna los primeros 1024 bytes del archivo al cluster 6, e indica en donde se encuentra el siguiente cluster asignado al archivo poniendo un 7 en el campo 6, como se muestra en la figura -3.9(B)-. Un 6 es también grabado en el campo del directorio como el primer cluster asignado al archivo.

Para leer nuestro archivo primero leemos la información que se encuentra en el cluster 6, además el campo del cluster 6 contiene el 7; que es el siguiente cluster en la cadena de clusters que contienen nuestro archivo.

DOS asigna los primeros 3Kb (3 072 bytes) del archivo encadenando los clusters como se muestra en la figura -3.9 (C)-.

Como son 3073 bytes, nos falta un byte por asignar, a este byte debe asignarse un cluster completo.

DOS reserva este cluster para el archivo marcándolo con el valor FFF, dentro de la FAT, de esta manera indica que la cadena ha terminado. Como se muestra en la figura -3.9 (D)-.

MS-DOS contiene dos copias idénticas de la FAT de cada volumen, actualiza las copias simultáneamente siempre que el archivo se extienda o el directorio sea modificado. Si el acceso a un sector falla debido a un error en la lectura, lo intenta en la copia de la FAT hasta que la lectura del disco sea exitosa o las copias de la FAT se agoten.

**EL DIRECTORIO RAÍZ
(THE ROOT DIRECTORY)**

Después de las dos copias de la FAT se tiene el directorio del disco. Este se divide en campos. Cada campo consta de 32 bytes.

DESPLAZAMIENTO	DESCRIPCIÓN
00H	Nombre del archivo (nota 1)
08H	Extensión (nota 2)
0BH	Atributo del archivo (nota 3)
0CH	Reservado (nota 4)
16H	Hora de creación u hora de última actualización (nota 4)
18H	Fecha de creación o de última actualización (nota 5)
1AH	Cluster de inicio del archivo, en la FAT (nota 6)
1CH	Tamaño del archivo (nota 7)

Figura -3.10-

DESCRIPCIÓN DE LAS NOTAS DE LA FIGURA -3.10-

- 1- El primer byte del nombre del archivo (filename) puede contener la siguiente información especial:

VALOR	SIGNIFICADO
00H	El campo nunca ha sido usado; fin de la porción ocupada por el directorio.
05H	El primer carácter de nombre del archivo es actualmente E5h.
2EH	El campo contiene el directorio padre. Si el siguiente byte es también 2Eh, el campo del cluster contiene el número del directorio padre (cero si el directorio padre es el directorio raíz).
E5H	El archivo ha sido borrado.

- 2- Mapa del Byte de atributo :

BIT	SIGNIFICADO
0	Solo lectura; si se intenta abrir el archivo para escribir o borrar la acción fallara.
1	Archivo escondido; excluido de búsqueda normal.
2	Archivo del sistema; excluido de búsqueda normal.
3	Etiqueta; puede existir solo en el directorio raíz.
4	Directorio; excluido de búsqueda normal.

5	Bit de archivo; se fija cuando el archivo es modificado.
6	Reservado
7	Reservado

- 3- El campo que contiene el tiempo se decodifica de la siguiente manera:

BITS	SIGNIFICADO
00H-04H	Número binario de incrementos de 2-segundos (0-29, que corresponden a 0-59 segundos).
05H-0AH	Número binario para los minutos (0-59).
0BH-0FH	Número binario para las horas (0-23).

- 4- El campo de la fecha se decodifica de la siguiente manera:

BITS	SIGNIFICADO
00H-04H	Día (1-31).
05H-08H	Mes (1-12).
09H-0FH	Año (Relativo a 1980).

- 5- El tamaño del archivo es interpretado como un entero de cuatro bytes.

EL AREA DE DATOS

Después del directorio, se tiene el área para almacenar la información, que es el resto del disco.

MS-DOS ve los sectores en esta área como un conjunto de clusters, cada cluster contiene uno o más sectores, cada cluster corresponde a un campo de la FAT que describe su uso, y cada nombre del archivo es almacenado en el directorio.

EL DISCO DURO

El disco duro trabaja de la misma manera que el disco flexible. La diferencia principal es que el disco flexible consiste de un único disco (platter), el disco duro consiste de uno o más platters. Cada platter contiene dos superficies sobre las cuales los datos son almacenados. MS-DOS divide cada platter en tracks y cada track esta subdividido en sectores. Todos los tracks que tienen el mismo radio forman un cilindro. Así todos los tracks numerados con cero forman un cilindro, todos los tracks numerados con "uno" forman otro cilindro y así sucesivamente.

El disco duro se divide en particiones lógicas, cada partición permite al dispositivo alojar diferentes sistemas operativos.

Cada partición tiene una tabla localizada en un sector al principio del disco.

Esta tabla indica donde se localiza el bloque del dispositivo lógico (logical block device).

Bajo MS-DOS en la partición standard, el primer sector físico del disco duro contiene la tabla de partición y el programa bootstrap capaz de checar la tabla de partición para una partición que contenga el sistema operativo, cargar el boot sector y transferir el control a este.

La tabla de partición, localizada al final del primer sector físico del disco, puede contener como máximo cuatro campos:

Desplazamiento desde el principio del sector

BYTES	CONTENIDO
000-1BDH	Reservado
1BE-1CDH	Partición #1
1CE-1DDH	Partición #2
1DE-1EDH	Partición #3
1EE-1FDH	Partición #4
1FE-1FFH	Distintivo AA55H

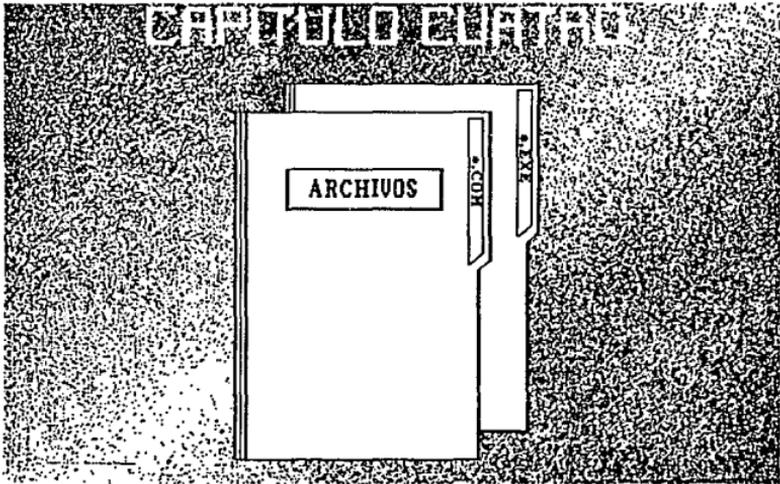
Figura -3.11-

Cada campo de 16 Bytes, contiene la siguiente información:

BYTE(S)	CONTENIDO
00H	Indicador del Boot. 0 = no bootable* 80H = bootable*
01H	Cabeza inicial.
02H-03H	Principio del cilindro / sector
04H	tipo de partición 00H no usada 01H FAT de 12 Bits 04H FAT de 16 Bits 05H partición extendida 06H partición grande (versión >= 4.0)
05H	Cabeza final.
06H-07H	Fin del cilindro / sector.
08H-0BH	Comienzo de la partición (comienzo relativo del disco).
0CH-0FH	Longitud de la partición en sectores.

Figura -3.12-

*Se le llama a un disco BOOTABLE, cuando este tiene los archivos del Sistema Operativo.



**PREFIJO DEL SEGMENTO DEL PROGRAMA
(PROGRAM SEGMENT PREFIX - PSP-)**

El PSP es un área reservada, de 256 bytes, que MS-DOS fija en la base del bloque de memoria asignado a los programas transitorios .COM y .EXE cuando son cargados en memoria. Contiene algunos enlaces con MS-DOS que pueden ser usados por los programas transitorios, información que MS-DOS pasa a los programas transitorios para ser usada o no según el programa los requiera.

0000H	Interrupción 20H.
0002H	Segmento y final del bloque.
0004H	Reservado.
0005H	Llamada a la función despachadora del MS-DOS.
000AH	Contenido previo del manejador de vectores de interrupciones.
000EH	Contenido previo del vector de interrupción CTRL-C
0012H	Contenido previo del vector de interrupción CRITICAL-ERROR.
0016H	Reservado.
002CH	Dirección del segmento del bloque del medio ambiente.
002EH	Reservado.
005CH	Bloque del control de archivos por omisión No. 1.
006CH	Bloque del control de archivos por omisión No. 2.
0080H	Buffer para lista de comandos.

Figura -4.1-

PSP:0000h (Terminate vector).

Esta dirección contiene la instrucción Int 20h, que el programa usa para transferir el control a MS-DOS. Este método de terminación no debe ser usado más en nuevos programas.

PSP:0002h (Address of last segment allocated to program).

Contiene la dirección del segmento del siguiente párrafo (paragraph) del bloque de memoria asignado al programa. Esta deberá ser usada solo para determinar el tamaño o el final del bloque de memoria asignada al programa; no debe ser considerada como un apuntador a memoria disponible que el programa puede apropiarse.

PSP:0005h (MS-DOS function call vector).

MS-DOS suministra este vector solo como soporte a los programas escritos bajo el estilo de CP/M. Este vector no deberá ser usado para llamar a MS-DOS en programas nuevos.

PSP:000Ah-0015h (Parents 22h,23h y 24h interrupt vector save).

MS-DOS salva estos vectores porque permite a algún programa (llamado programa padre) ejecutar otro programa (llamado programa hijo) y por medio de una llamada a MS-DOS regresa el control al programa padre siempre que el programa hijo haya concluido su trabajo, MS-DOS debe volver a cargar estos tres vectores al programa padre en caso de que el programa hijo los haya cambiado.

Los tres vectores involucrados son:

Int 22h (program termination handler)

Int 23h (control-C/control-break handler vector)

Int 24h (critical error handler).MS-DOS salva los vectores originales en el PSP del programa hijo.

PSP:002Ch (segment address of environment).

Contiene la dirección del segmento del primer párrafo del bloque del medio ambiente (environment block) del MS-DOS. Este apuntador habilita al programa para buscar dentro del bloque del medio ambiente, alguna configuración o directorio alguna trayectoria definida por el usuario por medio del comando SET.

PSP:0050h (new MS-DOS call vector).

Muchos programadores desprecian el contenido de esta localidad porque solo contiene las instrucciones Int 21H y RETF.

Un programa .EXE puede llamar esta localidad por medio de una llamada larga (Far Call). Desde luego el programa puede hacer un simple llamado a la Int 21h directamente, que es mucho más pequeña y rápida que llamar a 50h.

PSP:005Ch (default file control block 1) y

PSP:006Ch (default file control block 2).

Muchos programadores no usan ya esta característica, porque el acceso a archivos usando el FCB no soporta directorios con trayectorias y otras nuevas características de MS-DOS.

PSP:0080h (command tail y default transfer área TPA).

MS-DOS usa el área comprendida entre 0080h y 00FFh como DTA (default transfer área). El DTA se usa como buffer de datos en caso de que se realice una operación de lectura o escritura sobre un archivo usando las funciones del FCB para accederlo. La misma área de memoria nos sirve como buffer para la lista de comandos (command tail).

**PROGRAMAS CON EXTENSIÓN
".COM"**

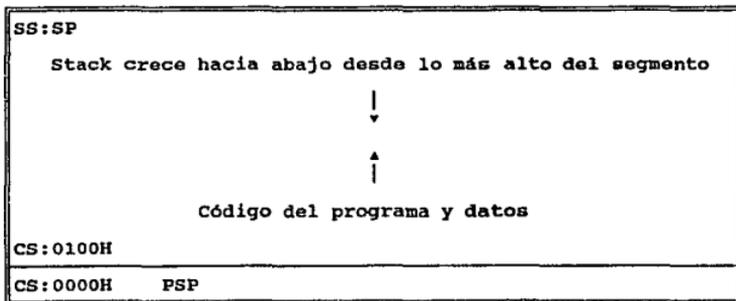
Los programas .COM son cargados arriba del PSP y no tiene ningún encabezado (header) que especifique otro punto de entrada, siempre tiene el origen 100h, que es la longitud del PSP.

Como los archivos .COM no tienen encabezado que informe a MS-DOS la memoria necesaria para asignar al programa; MS-DOS asigna el mayor bloque de memoria disponible a los programas .COM.

La longitud máxima de los programas .COM es de 65536 bytes, menos la longitud del PSP (256 bytes) y 2 bytes obligatorios del stack. Después que se ha asignado el mayor bloque de memoria a los programas .COM, MS-DOS construye el PSP en la parte baja del bloque de memoria asignado al programa. No existen diferencias entre el PSP construido para los programas .COM y .EXE.

Luego MS-DOS fija los registros DS, ES y SS apuntando el principio del PSP.

Finalmente, MS-DOS transfiere el control al programa fijando el registro CS con la dirección del segmento del PSP y el registro IP con 100h.



DS:0000H
ES:0000H
SS:0000H

Figura -4.2-

Mapa de la memoria, después de haber cargado un programa .COM. Cuando el programa .COM ha terminado su trabajo, este regresa el control a MS-DOS por medio de la interrupción 21h función 4Ch (terminate process with return code).

**PROGRAMAS CON EXTENSIÓN
.EXE**

Los programas .EXE pueden ser, prácticamente de un tamaño ilimitado. Ponen el código, datos y el stack en diferentes partes del archivo, y el programa ligador (LINKER) pone información extra al principio del archivo .EXE.

El cargador de MS-DOS trae los programas .EXE a memoria inmediatamente arriba del PSP, aunque el orden del código, datos y el stack puede variar.

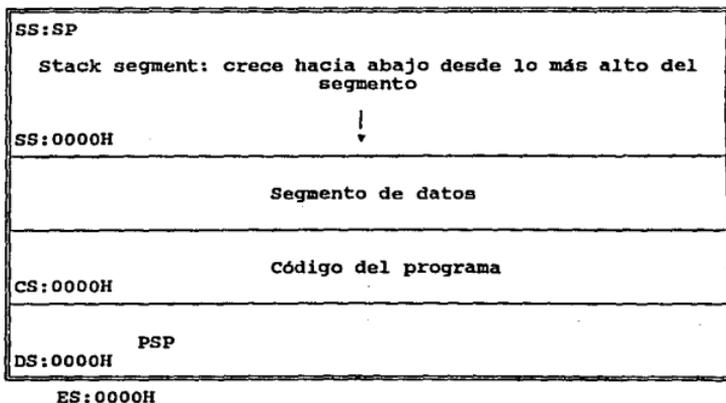


Figura -4.3-
Mapa de la memoria después de cargar un programa .EXE.

La información que el linker pone al principio del archivo se le llama encabezado del archivo (o bloque de control de información. El linker nunca crea un encabezado menor de 512 bytes, siempre mayores o iguales a 512 bytes y múltiplos de 512 bytes.

El encabezado del archivo .EXE contiene la siguiente información, que MS-DOS lee en un área de trabajo temporal dentro de la memoria para usarla mientras es cargado el programa .EXE.

0000H	Primera parte del distintivo del archivo EXE (4DH)
0001H	Segunda parte del distintivo del archivo EXE (5AH)
0002H	Longitud del archivo (MOD 512)
0004H	Tamaño del archivo en páginas de 512 Bytes incluyendo el encabezado
0006H	Número entradas de la tabla de relocalización
0008H	Número de párrafos en el encabezado (unidades de 16 Bytes)
000AH	Número mínimo de párrafos que necesita el programa
000CH	Número máximo de párrafos designados al programa
000EH	Desplazamiento del segmento del módulo Stack
0010H	Entrada que contiene el registro SP (Stack Pointer)
0012H	Suma de palabras (Checksum)
0014H	Entrada del contenido del IP (Instrucción Pointer)
0016H	Desplazamiento del segmento del módulo de código

0018H	Desplazamiento de la primera relocalización del archivo
001AH	Número overlay
001BH	Espacio reservado
	Tabla de relocalización
	Espacio reservado
	Segmento del programa y datos
	Segmento del Stack

Figura -4.4-

00-01h (.EXE signature).

MS-DOS reconoce que un archivo contiene un programa .EXE si los dos bytes en el encabezado contienen 4Dh 5Ah (que son los caracteres ASCII M y Z), si contiene cualquier otro valor MS-DOS asume que el archivo contiene un programa .COM.

02-03h (last page size).

La palabra en esta localidad da el número actual de bytes en la última página del archivo (página = 512 bytes).

04-05h (file pages).

Esta palabra contiene el número total de páginas requeridas par cargar el archivo.

Si el archivo contiene 1024 bytes, esta palabra contiene el valor 0002h; si el archivo contiene 1025 bytes, esta palabra contiene el valor 0003h.

06-07h (relation items).

Esta palabra da el número de entradas que existen en la tabla de relocalización de punteros (relocation pointer table).

08-09h (header paragraph).

Esta palabra da el tamaño del encabezado del archivo .EXE en párrafos (paragraphs).

0A-0Bh (MINALLOC).

Esta palabra indica el número mínimo de párrafos que el programa necesita para iniciar su ejecución y para retener el programa mismo.

0C-0Dh (MAXALLOC).

Esta palabra indica el número máximo de párrafos que podrían ser asignados al programa antes de su ejecución.

0E-0Fh (initial SS value).

Esta palabra contiene la dirección del párrafo del segmento stack relativo al inicio del programa a ser cargado.

Al momento de ser cargado, MS-DOS relocaliza esta dirección sumando la dirección del segmento en donde inicia el programa, y la dirección resultante es asignado al SS antes de dar el control al programa.

10-11h (initial SP value).

Esta palabra contiene el valor absoluto que MS-DOS carga en el registro SP antes de dar control al programa.

12-13h (complemented checksum).

Esta palabra contiene la suma de todas las palabras en el archivo .EXE. Las versiones actuales de MS-DOS ignoran esta palabra al cargado el programa .EXE.

14-15h (initial IP value).

Esta palabra contiene el valor absoluto que MS-DOS carga dentro del registro IP.

16-17h (pre-relocated initial CS value).

Esta palabra contiene el valor inicial, relativo al inicio del programa a ser cargado, MS-DOS ajusta este valor de la misma manera que lo hizo con el SS antes de ser cargado el registro CS.

18-19h (relocation table offset).

Esta palabra da el offset del inicio del archivo para localizar la tabla de relocalización de punteros (relocation pointer table).

1A-1Bh (overlay number).

Esta palabra tiene un valor, normalmente, de 0000h, indicando que el archivo .EXE es parte de un programa residente o primario.

La siguiente es una lista de los pasos que realiza MS-DOS para cargar un programa .EXE:

1. Lee el encabezado del archivo .EXE dentro de un área de trabajo.
2. Determina el tamaño del mayor bloque de memoria disponible.

3. Determina el tamaño del programa a cargar usando los datos contenidos en el tamaño de la última página (last page size), páginas del archivo (file pages) y párrafos del encabezado (header paragraphs).

4. Suma el valor del campo MINALLOC y el tamaño del PSP (100h bytes).
Si el resultado excede el tamaño del mayor bloque disponible, MS-DOS termina el proceso de carga y regresa un error.

5. Suma el valor del campo MAXALLOC y el tamaño del PSP. Si el tamaño del bloque encontrado excede el total calculado, asigna la memoria al programa; si el total calculado excede el bloque de memoria, MS-DOS asigna el bloque entero.

6. Si MINALLOC y MAXALLOC ambos contienen 0000h, MS-DOS usa el tamaño del programa para determinar el inicio del segmento. Si MINALLOC y MAXALLOC contienen valores diferentes de cero (el caso normal), MS-DOS establece el inicio del segmento enseguida del PSP.

7. Carga el programa al inicio del PSP.

8. Lee los apuntadores de relocalización en el área de trabajo (work área) y relocaliza el programa.
9. Construye el PSP en los primeros 100h bytes de la memoria asignada y determina los valores iniciales para los registros AL y AH.
10. Fija los valores de los registros SS y SP.
11. MS-DOS fija los valores de los registros DS y ES que apuntan el inicio del PSP.
12. Transfiere el control al programa .EXE fijando el valor de los registros CS y IP.

LA FUNCIÓN "EXEC" DEL MS-DOS

El sistema cargador de MS-DOS, que carga en memoria los archivos .COM o .EXE y corre los programas contenidos en estos archivos, puede ser llamado por cualquier sistema operativo que contenga la función EXEC del MS-DOS. La función EXEC (int 21h función 4Bh) permite a un programa (llamado padre) cargar otro programa (llamado hijo) desde un dispositivo de almacenamiento, ejecutarlo y recobrar el control cuando el programa hijo ha terminado.

La función EXEC (Int 21h función 4Bh) permite a un programa (llamado padre) cargar cualquier otro programa (llamado hijo) desde algún dispositivo de almacenamiento (floppy disk) ejecutarlo y luego recobrar el control cuando el programa hijo haya terminado. Cuando se solicita a la función EXEC que ejecute algún programa, lo primero que intenta es localizar y abrir el archivo especificado por el nombre del programa. Si el archivo no se encuentra, EXEC falla inmediatamente y regresa un código de error al que lo llamo (programa padre).

Si el archivo existe, EXEC abre el archivo, determina su tamaño, revisa el primer bloque (sector) del archivo. Si los dos primeros bytes del archivo son los caracteres ASCII MZ, el archivo contienen un programa .EXE, y el tamaño del código del programa, datos, stack segment se obtiene de la cabecera del archivo (file header) .EXE. De otra manera se toma el archivo como .COM que se asigna el bloque más grande de memoria (64kb de memoria).

Hasta este punto se conoce la cantidad de memoria necesaria para cargar el programa, entonces EXEC intenta asignar dos bloques de memoria: uno que contenga los nuevo ambientes(environment) del programa y el otro bloque contiene el código del programa, datos y el stack segment. Suponiendo que hay suficiente memoria para cargar el programa, la cantidad asignada de memoria varia de acuerdo al tipo de programa.

A los programas .COM por lo general se les asigna toda la memoria disponible (no más de 64kb), mientras la cantidad asignada a los programas .EXE es controlada por los campos que se encuentra en el encabezado del archivo, MINALLOC y MAXALLOC.

Luego EXEC copia el ambiente (environment) del padre a la memoria asignada al environment del programa hijo, construye el PSP (program segment prefix) en la base del bloque de memoria asignada al programa hijo y por último copia, del programa padre, la cola de comandos (command tail) , FCB (file control block) dentro del PSP del programa hijo.

El contenido previo del vector terminación (interrupción 22h), Control-C (Int 23h), y Error-critico (Int 24h) son salvadas dentro del nuevo PSP, y el vector de terminación es actualizado para regresar el control al programa padre cuando el programa hijo ha terminado o sea abortado.

El código y los datos del programa hijo son leídos desde el archivo del disco al bloque de memoria asignado al programa hijo sobre el nuevo PSP. Si el archivo del programa hijo tiene la extensión .EXE se utiliza la información contenida en el encabezado del archivo para fijar los segmentos de referencia dentro del programa para reflejar su actual dirección de cargado.

La función EXEC fija los registros del CPU de acuerdo al tipo de programa y transfiere el control al programa. El punto de entrada los programas .COM es siempre en el offset 100h dentro del bloque de memoria asignado al programa (el primer byte después del PSP). El punto de entrada para un programa .EXE se especifica en el encabezado del archivo y puede ser cualquiera dentro del programa. Cuando el programa hijo ha terminado, el programa padre recobra el control y puede usar alguna llamada al sistema de funciones para obtener el código de retorno del programa hijo para determinar si el programa hijo terminó normalmente, por que hay error crítico en el hardware o por que el usuario ha dado el comando Control-C. El archivo COMMAND.COM del MS-DOS usa la función EXEC para correr sus comandos internos o programas de aplicación.

Sumario de diferencias entre los archivos .COM y .EXE

	PROGRAMAS CON EXTENSIÓN .COM	PROGRAMAS CON EXTENSIÓN .EXE
Tamaño	máximo 65536, mínimo 256 bytes del PSP y 2 bytes para el stack	No tiene limite
Punto de entrada	PSP : 0100H	Definido por la declaración END
CS	PSP	Segmento conteniendo el punto de entrada del programa
IP	0100H	Desplazamiento del punto de entrada con el segmento
DS	PSP	PSP
ES	PSP	PSP
SS	PSP	Segmento con el atributo del STACK
SP	FFFEH o palabra superior disponible en memoria	Fin del segmento definido con el atributo del STACK
Stack	Palabra cero	Depende del fuente
Tamaño del stack	máximo 65536 bytes, mínimo 256 bytes para PSP y tamaño de código y datos ejecutables	Segmento definido con el atributo del STACK
Llamada a subrutinas	NEAR (Mismo segmento)	NEAR o FAR (en el mismo segmento o a otro segmento)
Método de salida	Interrupción 21H Función 4CH	Interrupción 21H función 4CH
Tamaño del archivo	Tamaño exacto del programa	El tamaño del archivo más el encabezado

Figura -4.5-

PROGRAMAS RESIDENTES (TSR) - 1
(Terminate and Stay Resident)

MS-DOS es capaz de soportar varios programas en memoria ,pero solo uno se esta ejecutando, estos programas múltiples se crean por medio del cargador de MS-DOS y regresando el control a MS-DOS sin remover el programa de memoria. Por que el programa no deja la memoria cuando regresa el control al sistema operativo, estos programas son llamados programas residentes.

El código y datos de los TSR puede ser dividido en dos partes:

- Parte residente.
- Parte transitoria.

Dirección alta . . .	Inicialización de código y datos (porción transitoria, cuando se ejecuta al corren el programa)
. . .	Aplicación de código y datos (porción residente en RAM)
. . .	Monitoreo de rutinas (Porción residente en RAM)
Dirección Baja	PSP

Figura -4.4-

La parte residente contiene el código y datos necesarios para realizar la función demandada. La parte transitoria instala el TSR en memoria.

Los TSR se pueden dividir en pasivos o activos dependiendo del método que utilice para transferir el control a la parte residente. Un TSR pasivo se ejecuta solo cuando un programa transfiera el control a este, por medio de una interrupción o por medio de un salto grande o llamada (long JMP o CALL).

El programa que ha llamado al TSR se interrumpe, el estado de MS-DOS, BIOS y el hardware están bien definidos cuando el TSR toma el control.

Los TSR activos son llamados cuando ocurre algún evento externo para el estado del programa que esta corriendo actualmente, como la presión de una tecla o cuando es llamada una interrupción hardware.

Cuando es llamado el TSR casi siempre suspende el programa que se esta ejecutando. Para evitar alguna degeneración del programa que estaba corriendo, monitorea el estado de MS-DOS, ROM BIOS y hardware; toma el control del sistema solo cuando ha salvado el estado de MS-DOS y el hardware es estable.

MS-DOS suministra dos mecanismos para terminar un programa y dejar su parte residente en memoria. Estos son, la interrupción 21h función 32h y la interrupción 27h.

El método más usado es ejecutar la interrupción 21h función 31h.

CAPITULO CINCO



El desarrollo del programa PROSOFT esta constituido por cinco módulos principales, que son:

- ▶ Manejador de módulos
- ▶ Teclado
- ▶ Discos flexibles
- ▶ Disco duro
- ▶ Ayuda

El manejador de módulos se encarga de llamar a las otras rutinas cuando sean invocadas estas por el usuario o regresar al Sistema Operativo. El módulo de ayuda es el único que regresa al manejador de módulos, los demás realizan su función y terminando esta, regresan al Sistema Operativo (módulos de teclado y discos flexibles) o prepara al sistema para apagar la máquina (disco fijo) y un módulo independiente ubicado en el disco llave.

A continuación se describirá cada módulo:

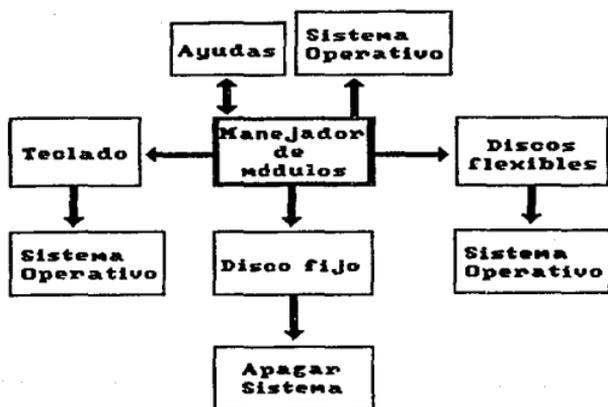


Figura -5.1-

▶ AYUDA.

Proporcionará dos tipos de ayudas, general y sensitiva al contexto. La ayuda general dará la información global del módulo que solicite el usuario y la ayuda sensitiva al contexto, desplegará una descripción específica del tópico, en donde se encuentre el cursor.

AYUDA

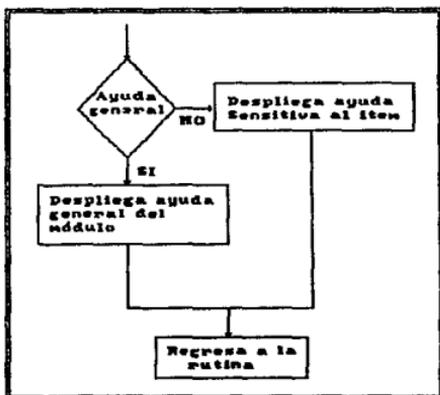


Figura -5.2-

TECLADO.

Tiene dos submodulos, desactivación de teclado y cambio de clave.

El primero ve si tiene una clave, si no la tiene se la pide al usuario y en seguida desactiva el teclado, entrando a un ciclo de verificación de la clave, saliéndose del ciclo cuando la clave que se le tecleo fue la correcta y activara el teclado, saliéndose al Sistema Operativo (Figura -5.3-) y el segundo pide la clave anterior, si es correcta pide la nueva clave y la guarda (Figura -5.4-).

TECLADO

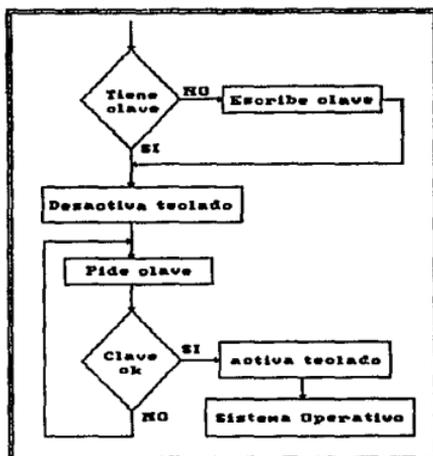


Figura -5.3-

CAMBIO DE CLAVE (MÓDULO DOS DE TECLADO)

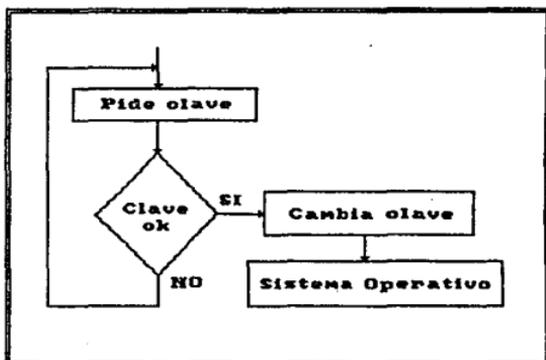


Figura -5.4-

DISCOS FLEXIBLES.

Este módulo es residente en memoria RAM y se compone de tres partes, la primera desactiva los drives "A:" y "B:" (Figura -5.5-), la segunda deja el acceso a estos pero sin liberar la memoria (Figura -5.6-) y la tercera, permite el acceso a los drives A: o B: y quita al programa liberando la memoria RAM (Figura -5.7-) la sección punteada indica que esta residente en memoria y por lo tanto el Sistema Operativo esta activo.

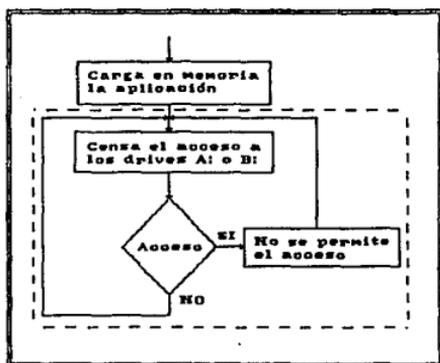


Figura -5.5-

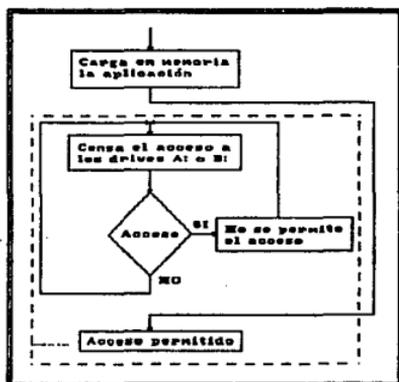


Figura -5.6-

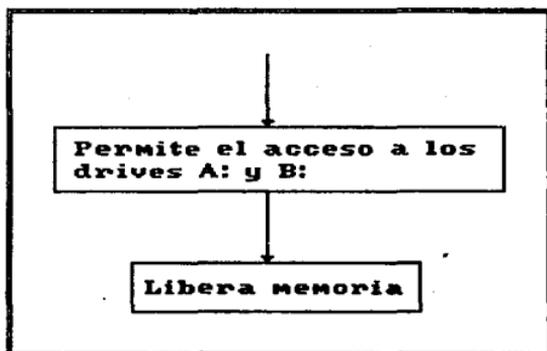


Figura -5.7-

► DISCO DURO.

En esta opción se pedirá la clave de acceso, si la clave es correcta se desactivará el disco duro.

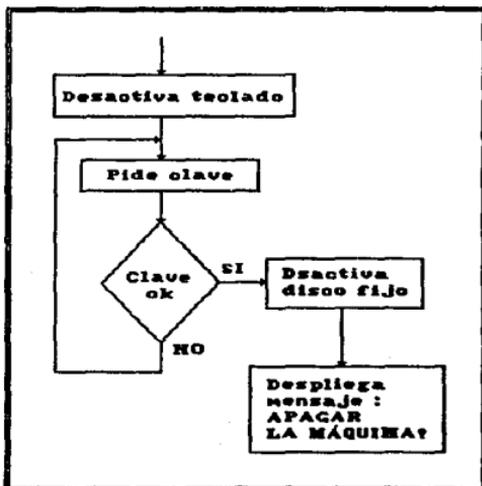


Figura -5.8-

EL MÓDULO INDEPENDIENTE.

Pedirá una clave de acceso, si esta es correcta activara el disco duro, desplegará el mensaje de quitar el disco llave y teclear <RETURN>, para generar una recarga al Sistema Operativo.

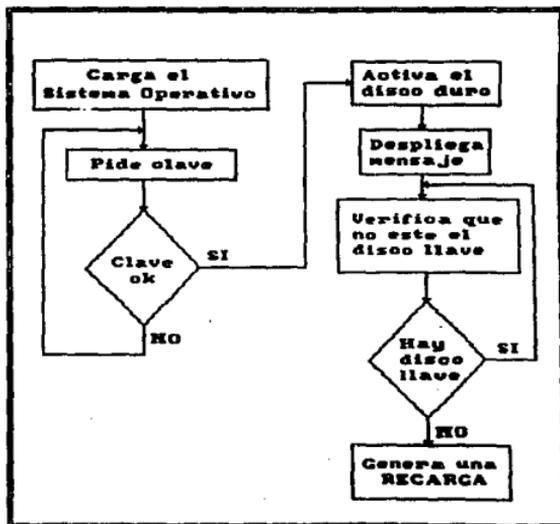
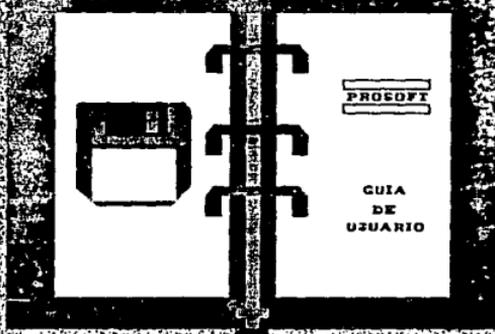


Figura -5.9-

CAPITULO SEIS



PROSOFT

GUIA
DE
USUARIO

1.- INTRODUCCIÓN

El paquete "PROSOFT" ayudará al usuario a evitar el acceso a la computadora a personas no autorizadas.

El usuario, por medio de claves de acceso podrá impedir el acceso al teclado, a la desactivación del disco duro y a la activación de este.

El paquete trabaja por medio de menús temporizados a cinco minutos, en cada menú aparecerá en la parte inferior de la pantalla, las teclas de movimiento, de selección, la tecla de salir del menú (ESC) (pasándose al menú anterior o si está en el menú principal termina de correr el paquete).

Con la tecla "F1" se obtiene una ayuda general del menú en el que se encuentra, y si se desea una ayuda más específica sobre cada opción, coloque el cursor donde se desee la ayuda y presione "Ctrl + F1" (las dos teclas juntas).

2.- AMBIENTE

El programa PROSOFT está desarrollado para correr en una computadora personal IBM o compatible que contenga disco duro y Sistema Operativo DOS versión 3.3. o mayor.

3.- INSTALACIÓN

Para instalar el programa PROSOFT realizar los siguientes pasos:

a.-Inserte el disco en el drive A, teclear A:[ENTER]

b.-Teclear INSTALA [ENTER]

Para correr el programa los pasos son los siguientes:

a.-Teclear PS [ENTER]

4. TECLAS UTILIZADAS

- ↑ Las teclas indicadas permiten posicionarnos en la opción de interés de un menú.
- ← Una vez posicionado en la opción deseada la selección se hace oprimiendo ésta tecla (Intro o Enter).
- Esc Esta tecla permite el regreso al menú anterior (o salir del programa si el usuario se encuentra en el menú principal).
- F1 Con esta tecla se desplegará la ayuda general del menú.
- Ctrl-F1 Desplegará la ayuda específica en donde se encuentre el cursor.

5. DESCRIPCIÓN DE MENÚS

5.1 MENÚ PRINCIPAL.

Se encuentran las siguientes opciones:

- password de teclado.
- password de floppys.
- disco fijo.

5.1.1 - PASSWORD DE TECLADO -

En este menú se tienen las opciones de desactivar el teclado y/o cambiar de clave.

5.1.2 - PASSWORD DE FLOPPYS -

Las opciones de este menú son las siguientes:

- Desactivar discos flexibles.
- Activar discos flexibles.
- Restaurar los manejadores de los discos.

No permitirá el acceso a los discos flexibles "A" o "B". Desplegando un mensaje de aviso y un sonido de alarma, tecleando la tecla "ESC" cesará el aviso y el sonido simultáneamente.

5.1.3 - DISCO FIJO -

Desactivará el disco fijo, esto es, una vez desactivado NO se podrá tener acceso al disco duro hasta que se haga el procedimiento correspondiente. Si está desactivado el disco y si no se tiene el método para activarlo no hay que tratarlo de activar con cualquier herramienta, porque se podría perder toda la información del disco duro.

5.1.4 - DESACTIVACIÓN DEL TECLADO -

En esta opción el teclado quedará bloqueado y en espera de la clave de acceso, la indicación de que el teclado quedo bloqueado será, la pantalla se limpiará y en la parte superior izquierda estará una llave en indicación de espera de la clave de acceso. Una vez que se le dio la clave de acceso, el programa terminará limpiando la pantalla y quedando en la ruta donde se le llamo.

En el caso de que se apagara la maquina cuando el sistema está esperando la palabra clave, se debe inicializar el sistema con el procedimiento de arranque.

5.1.5 - NUEVA CLAVE DE TECLADO -

Para colocar una nueva clave de acceso al teclado, primero se tiene que dar la clave anterior y en seguida se pedirá que se de la nueva clave, esta clave se tiene que ser de ocho dígitos (se recomienda teclear con cuidado ya que una vez dado "ENTER" o teclado más de ocho dígitos, no se podrá tener la clave).

5.1.6 - DESACTIVAR DISCOS FLEXIBLES -

Esta opción una vez activa, NO permitirá el acceso a los floppys "A" o "B".

5.1.7 - ACTIVAR DISCO FLEXIBLE -

En esta opción se permitirá el acceso a los floppys "A" o "B".

5.1.8 - RESTAURAR DISCO FLEXIBLE -

habilitará a los floppys "A" o "B", liberando parte de la memoria "RAM".

6. - LA ACTIVACIÓN POR DISCO FLEXIBLE -

Poner el disco "llave" en el drive "A" y reinicializar el sistema.

Se tiene que dar la palabra clave para la activación del disco duro.

CONCLUSIONES

Para que se tenga seguridad en una computadora personal, compatible con IBM, se podrá utilizar una gran variedad de mecanismos de protección, ya sea utilizando algunos componentes físicos o por medio de programas.

En la utilización de componentes físicos se tendrán de dos tipos:

- Los que son por medio de un switch que desactiva el teclado.

- Y los que por medio de circuitos lógicos puedan desactivar la fuente de poder, el procesador o cualquier otro circuito indispensable para el funcionamiento de la computadora.

Se vio que con estos sistemas se tiene que modificar parte de la estructura física de la computadora, lo cual no es muy práctico, o en su defecto que la computadora ya tenga de fábrica una protección de este tipo.

En la utilización de programas de protección, se observó que algunas computadoras ya traen este programa dentro del sistema operativo.

CONCLUSIONES

Y para la creación de un programa de protección fuera del sistema de la máquina, hay que tener mucho cuidado para no afectar el funcionamiento de la computadora ya que se puede perder toda la información.

Podemos concluir que la mayoría de las protecciones son buenas y cumplen adecuadamente con su función, pero si el usuario no tiene cuidado, de nada sirve que se posea la mejor protección.

REFERENCIA BIBLIOGRÁFICA

CAPITULO	LIBROS DE REFERENCIA
<p align="center">UNO</p>	<ul style="list-style-type: none"> ◀ OPERATING SYSTEMS CONCEPTS AND APPLICATIONS Donald R. Horner Scott, Foresman and Company 1989 Capitulos : 1, 2, 3. ◀ SISTEMAS OPERATIVOS DE LA COMPUTACIÓN William S. Davis Representaciones y Servicios de Ingeniería, S.A. 1980 Capitulos : 1, 2, 3, 4.
<p align="center">DOS</p>	<ul style="list-style-type: none"> ◀ ADVANCED MS-DOS PROGRAMMING Ray Duncan Microsoft Press. 1988 Capitulos : 1, 2. ◀ MANUAL DE CONSULTA ABREVIADO IBM Corp. y Microsoft Inc. ◀ MS-DOS ENCYCLOPEDIA Ray Duncan, Foreword by Bill Gates Microsoft Press. 1988 Sección I, Sección II, Parte A, Artículo 1,2. ◀ PROGRAMMER'S GUIDE TO THE IBM PC Peter Norton Microsoft Press. 1988

REFERENCIA BIBLIOGRÁFICA

CAPÍTULO	LIBROS DE REFERENCIA
TRES	<ul style="list-style-type: none"> ◀ ADVANCED ASSEMBLY LANGUAGE ON THE IBM PC Steven Holzner Brady. 1987 Capítulo 6. ◀ ADVANCED MS-DOS PROGRAMMING Ray Duncan Microsoft Press. 1988 Capítulo 10. ◀ MS-DOS ENCYCLOPEDIA Ray Duncan, Foreword by Bill Gates Microsoft Press. 1988 Sección II, Parte A, Artículo 3. ◀ WRITING MS-DOS DEVICE DRIVERS Roberts Lai / The Waite Group Addison Wesley. 1987 Capítulo 7.
CUATRO	<ul style="list-style-type: none"> ◀ ADVANCED MS-DOS PROGRAMMING Ray Duncan Microsoft Press. 1988 Capítulos : 3, 8. ◀ MS-DOS ENCYCLOPEDIA Ray Duncan, Foreword by Bill Gates Microsoft Press. 1988 Sección II, Parte B, Artículo 4 Sección II, Parte C, Artículo 11 ◀ PROGRAMMER'S GUIDE TO THE IBM PC Peter Norton Microsoft Press. 1988

REFERENCIA BIBLIOGRÁFICA

CAPITULO	LIBROS DE REFERENCIA
CINCO	<ul style="list-style-type: none"> ◀ POWER USER'S GUIDE Herbert Schildt Mc. Graw Hill. 1988 (Consulta de desarrollo) ◀ SYSTEM BIOS FOR IBM PC/XT/AT COMPUTERS AND COMPATIBLES (Technical Reference Series -Phoenix-) Addison-Wesley Publishing Company, Inc. 1989 (Consulta de desarrollo) ◀ TURBO C BIBLE Barkakaty Howard w Sams & Company. 1989 (Consulta de desarrollo) ◀ USING TURBO C Herbert Schildt Mc. Graw Hill. 1989 (Consulta de desarrollo) ◀ 8088 ASSEMBLER LANGUAGE PROGRAMMING: THE IBM PC David C. Willen / Jeffrey I. Kranz Howard W. Sams & Company. 1989 (Consulta de desarrollo) ◀ 8088 MICROPROCESSOR PROGRAMMING, INTERFACING, SOFTWARE, HARDWARE AND APLICATIONS. Avtar Singh / Walter A. Triebel Prentice Hall. 1989 (Consulta de desarrollo)
SEIS	NO TIENE REFERENCIA