

**Unidades de Disco**

**flexible fl-x y ml-x**

## CONTENIDO

Unidades X de disco flexible (Floppys) VORTEX. Tipos F1-X y M1-X para todos los ordenadores: CPC464+DDI-1, CPC664 y CPC6128 .....	1
Unidades de disco flexible VORTEX: resumen de características generales .....	5
Puesta en marcha del ordenador CPC 464 + DDI-1 con unidad X de disco .....	7
Puesta en marcha de los ordenadores CPC 664/CPC6128 con unidad X de disco .....	9
Discos flexibles, discos flexibles y siempre discos flexibles .....	10
¿Qué es un "DOS"? .....	12
Fundamentos .....	14
La orden <b>!FORMAT</b> del sistema operativo VDOS .....	16
Puesta en marcha y conservación de un programa BASIC .....	18
¿Qué es un fichero o grupo de datos? .....	21
Las órdenes <b>!A</b> y <b>!B</b> del sistema operativo VDOS .....	23
La orden <b>!X</b> del sistema operativo VDOS .....	25
Las órdenes <b>CAT</b> y <b>!DIR</b> del sistema operativo VDOS .....	27
La orden <b>!ATTRIBUT</b> del sistema operativo VDOS .....	31
La orden <b>!ERA</b> del sistema operativo VDOS .....	33
La orden <b>!REN</b> del sistema operativo VDOS .....	34
La orden <b>!SELECT</b> del sistema operativo VDOS .....	35
La orden <b>!RESET</b> del sistema operativo VDOS .....	36
La orden <b>!CODE</b> del sistema operativo VDOS .....	37
La orden <b>!AMSDOS</b> del sistema operativo VDOS .....	38
Reproductor de cintas en el ordenador CPC .....	39
La orden <b>!DISC</b> del sistema operativo VDOS .....	40
La orden <b>!CAS</b> del sistema operativo VDOS .....	40
La orden <b>!DISC.IN</b> del sistema operativo VDOS .....	40
La orden <b>!CAS.IN</b> del sistema operativo VDOS .....	40
La orden <b>!DIS.OUT</b> del sistema operativo VDOS .....	40
La orden <b>!CAS.OUT</b> del sistema operativo VDOS .....	41
La orden <b>!XMON</b> del sistema operativo VDOS: el monitor VORTEX "Z80" .....	41
La orden <b>!ROMOFF</b> del sistema operativo VDOS .....	51

La orden <b>!ROMS</b> del sistema operativo VDOS .....	52
Administración relativa y secuencial de ficheros o grupos de datos ..	52
La orden <b>!FILES</b> del sistema operativo VDOS .....	57
La orden <b>!OPEN</b> del sistema operativo VDOS .....	58
La orden <b>!FIELD</b> del sistema operativo VDOS .....	59
La orden <b>!GET</b> del sistema operativo VDOS .....	60
La orden <b>!PUT</b> del sistema operativo VDOS .....	61
La orden <b>!CLOSE</b> del sistema operativo VDOS .....	62
Posibilidades de detección de errores en los discos flexibles .....	66
La orden <b>!DERROR</b> del sistema operativo VDOS .....	66
Códigos de error y su significado .....	67
La función "giro de llave y en marcha" (turnkey) del sistema opera- tivo VDOS .....	73
Unidades X de disco flexible en los ordenadores CPC 464 con DDI-1, CPC 664 y CPC6128 .....	74
Producción de un disco flexible del sistema VORTEX para el siste- ma operativo CP/M 2.2 .....	75
Ordenadores modelo CPC 464 y CPC 664 con unidad de disco flexi- ble X y tarjeta de memoria RAM VORTEX .....	81
Confección de un disco VORTEX para el sistema operativo CP/M Plus (sólo en el ordenador tipo CPC 6128) .....	81
La carga del sistema operativo CP/M .....	85
La orden <b>!CPM</b> del sistema operativo VDOS .....	85
La orden <b>!CPM,1</b> del sistema operativo VDOS .....	86
La orden <b>!CPM,2</b> del sistema operativo VDOS .....	87
Sistema operativo CPM con la unidad de disco flexible X VORTEX .....	88
La orden <b>FILECOPY</b> del sistema operativo CP/M .....	88
La orden <b>XMON</b> del sistema operativo CP/M .....	98
La orden <b>SYSGEN</b> del sistema operativo CP/M .....	99
La orden <b>DISKTOOL</b> del sistema operativo CP/M .....	101
La orden <b>INSTALL</b> del sistema operativo CP/M .....	109
La orden <b>CASCOPY</b> del sistema operativo CP/M .....	124
La orden <b>VDOS</b> del sistema operativo CP/M .....	126
La orden <b>FAST</b> del sistema operativo CP/M .....	126
Avisos de error del sistema operativo CP/M y su eventual corrección	127
Los sistemas operativos VDOS y CP/M internamente .....	134
BIOS (Basic Input Output System, Sistema de Introducciones y Sali- das BASIC) .....	134

DOS (Disk Operating System, Sistema Operativo de Disco) .....	137
Variables de memoria RAM de los sistemas BIOS y DOS .....	148
Examen detallado del CP/M: el sistema operativo CP/M .....	164
La memoria RAM en el sistema operativo CP/M .....	168
Estructura del bloque de parámetros de discos (DPB) .....	168
Estructura de la cabecera de parámetros de disco, Disk Parameter Headers (DPH) .....	170
Disposición de un bloque de control de ficheros, File Control Block (FCB) .....	171
El vector de ubicación .....	171
El vector de comprobación de tamaño, Check Size Vektor .....	172
El vector de salto (Jump) del sistema operativo BIOS .....	173
Entradas del sistema operativo BIOS .....	174
Funciones del sistema operativo BDOS .....	175
Para finalizar comentaremos algo sobre el Hardware .....	177
Ocupación de las clavijas de los conectores .....	178
Clavija hembra de enchufe DIN en la unidad X de disco flexible ..	179
Alimentación de corriente al controlador de floppys VORTEX (no a la unidad X de disco flexible) .....	180
Unidades de disco flexible .....	181
<b>Apéndice 1: El problema EOF .....</b>	<b>183</b>

Unidades de disco flexible (Floppys) VORTEX  
Tipos F1-X y M1-X  
para todos los ordenadores:  
CPC464+DDI-1,  
CPC664 y CPC6128

Estimado poseedor de un ordenador CPC:

Hace aproximadamente un año (en Abril de 1985) vió la luz del mundo el primer "bebé de VORTEX":

la estación VORTEX de discos flexibles Floppy F1-D,

concebida para el ordenador CPC 464, entonces recién fabricado. A este nuestro primer retoño se le ha llamado "una pequeña revolución". Con razón, puesto que los datos de rendimiento de este aparato son efectivamente revolucionarios: alta capacidad de memoria (1.4 MegaByte), formato usual de disquetes (5.25"), disquetes baratos y fácilmente adquiribles, unidades de disco modernísimas, electrónica perfecta del controlador y software inteligente y rápido del sistema operativo.

También el servicio al cliente es de primera calidad. Lo mismo si se trata de explicar cómo está constituido el formato de un disquete, o si se necesita ayuda eficaz para la adaptación de un programa, en nuestro equipo de colaboradores encontrará usted siempre un técnico competente, que le ayudará gustosamente.

Nadie es perfecto . . . No existe ningún programa, que no tenga por lo menos un punto débil. Generalmente no es sólo uno. Si alguna otra marca le ha dejado colgado, nosotros no le abandonaremos. Estamos constantemente perfeccionando y puliendo nuestros equipos y, lo que es más importante, usted forma también parte de nuestras preocupaciones. Cuando existen novedades o se han corregido eventuales fallos, podrá equipar siempre su Floppy con este nuevo software, aunque haya comprado su aparato hace más de un año.

Reduciendo todo esto a un común denominador, sabrá por qué tenemos tanto éxito.

La rápida introducción en el mercado del ordenador CPC 664 y luego también del CPC 6128 nos ha inducido a darles descendencia. Hemos desarrollado con toda rapidez toda una serie de nuevas unidades de disco Floppy y, como coronación de este trabajo, una placa fija.

Una parte notable de nuestro trabajo ha estado ocupada por un proyecto, que hemos denominado internamente con el nombre de "proyecto X". En este nombre se esconde todo un grupo de unidades de disco, pensadas exclusivamente para trabajar junto con una unidad de disco de 3" y un controlador de Floppys de 3" ya disponibles. (CPC 464 + DDI-1 ... CPC 664 ... CPC 6128). En este trabajo habíamos excluído, naturalmente, el desarrollo sencillo y ya terminado de una segunda unidad de disco con menor capacidad de memoria (las unidades de disco VORTEX F1-Z y M1-Z). Pues, no; tenía que ser algo que hiciera totalmente innecesaria la compra de una "segunda unidad de disco".

La demanda nos decía:

"En primer lugar necesitamos una gran capacidad de memoria, a fin de terminar de una vez con nuestros problemas: nos bastaría con 700K.

Y, por favor, por fin un formato usual de disquetes que se puedan adquirir siempre y en todas partes a precios bajos: 5.25" y 3.5".

Además un sistema operativo profesional: VDOS 2.0 y naturalmente también CP/M 2.2 y CP/M Plus.

Y además también un estuche "que se balancee en su entorno".

Pero, sobre todo, que no haya que enviarlo a ningún sitio, ni soldarlo, etc. Y además está también la cuestión de los "derechos de reproducción" ...

El rendimiento y el precio deben ser sensacionales.

El resultado de todo esto han sido las unidades X de disco flexible VORTEX F1-X y M1-X:

- Alta capacidad de memoria: 708K, ajustados a formato.
- Formatos usuales de disquetes: 5.25" o bien 3.5".  
Disquetes baratos, sin problemas de suministro.

- Sistema operativo profesional VDOS 2.0.  
Módulo VDOS, en el que se encaja el controlador DDI-1.
- Administración relativa de los grupos de datos o ficheros bajo el programa BASIC.
- Monitor (programa de comprobación) del microordenador Z80 en el lenguaje de la máquina.
- Totalmente compatible con el software ya existente de los sistemas operativos VDOS 2.0 y AMSDOS.
- CP/M 2.2 y CP/M Plus.
- Preparados para el punto de interconexión en serie RS-232-C.
- 128 posibles registros de directorio.
- El precio ... de ensueño.

Usted se ha decidido también por este aparato. En realidad es lógico, puesto que no hay ninguna alternativa.

El presente manual describe todas las órdenes y detalles de las unidades VORTEX X de disco flexible.

Las informaciones detalladas concretas respecto a los sistemas operativos CP/M 2.2 y CP/M Plus las encontrará en los respectivos manuales, que le habrán entregado al comprar el ordenador CPC.

El presente manual consta de tres grandes partes:

- Descripción de todas las órdenes del sistema operativo VDOS.
- Descripción de todas las órdenes VORTEX del sistema CP/M. El capítulo "unidades X de disco flexible en el ordenador CPC 464 con controlador DDI-1, y en los ordenadores CPC 664 y CPC 6128" describe especialmente la forma en que puede fabricar un disquete del sistema CP/M.
- Informaciones técnicas sobre los sistemas operativos VDOS y CP/M en relación con las unidades X de disco flexible.

Tómese el tiempo necesario para leer este manual con toda tranquilidad. Esto le ayudará a usted y a nosotros.

Ahora, le deseamos mucho éxito con su nuevo Floppy VORTEX.

Sus amigos del equipo VORTEX.

## **MUY IMPORTANTE**

No está permitido sacar fotocopias de este manual, ni siquiera parcialmente, ni tampoco distribuir las, sin el permiso previo por escrito de la empresa VORTEX.

Nos reservamos el derecho a realizar cualquier modificación.

Todos los trabajos de mantenimiento y garantía deben ser realizados por VORTEX o por sus concesionarios autorizados. En cualquier otro caso, quedará anulado el derecho a reclamar la garantía.

Si no se empaquetan adecuadamente los aparatos devueltos a VORTEX para trabajos de mantenimiento y garantía (es frecuente que falte, sobre todo, el disquete de cartón de protección para el transporte en la unidad de disco), se perderá automáticamente cualquier derecho a la garantía.

**Envíense los aparatos siempre en sus embalajes originales.**

VORTEX no se responsabiliza en lo referente a la adecuación de los productos para determinadas aplicaciones.

Z80 es una marca registrada de la Firma Zilog Inc.

VORTEX y VORTEX "VDOS" son marcas registradas de la Firma VORTEX GmbH.

CP/M 2.2 y CP/M Plus son marcas registradas de la Firma Digital Research Inc.

WordStar es una marca registrada de la Firma MicroPro Inc.

dBase II es una marca registrada de la Firma Ashton Tate.

Multiplan es una marca registrada de la Firma Microsoft.



## Unidades X de disco flexible VORTEX: resumen de características generales

Nuestras unidades X de disco se pueden suministrar en dos versiones diferentes:

- con una unidad de disco de 5.25"
- o bien
- con una unidad de disco de 3.5",

ambas con una capacidad de memoria de 708K formateada.

Las unidades X de disco, equipadas con mecanismo de 5.25", tienen la denominación F1-X.

Las unidades X de disco, equipadas con un mecanismo de 3.5", tienen la denominación M1-X.

Son posibles las siguientes combinaciones de aparatos (juegos de equipos):

CPC464+DDI-1	+ F1-X 5.25"	, 708K, VDOS, AMSDOS, CP/M 2.2
CPC464+DDI-1	+ M1-X 3.5"	, 708K, VDOS, AMSDOS, CP/M 2.2
CPC664	+ F1-X 5.25"	, 708K, VDOS, AMSDOS, CP/M 2.2
CPC664	+ M1-X 3.5"	, 708K, VDOS, AMSDOS, CP/M 2.2
CPC6128	+ F1-X 5.25"	, 708K, VDOS, AMSDOS, CP/M 2.2 Plus
CPC6128	+ M1-X 3.5"	, 708K, VDOS, AMSDOS, CP/M 2.2 Plus

En el presente manual hablaremos siempre de unidades X de disco puesto que todas las órdenes en los aparatos M1-X y F1-X son absolutamente idénticas.

Todas las unidades X de disco VORTEX necesitan tensión de servicio de 220 V a 50 Hz.

En todas las unidades X de disco está incluido el denominado módulo VDOS (pequeño interfaz en una atractiva caja de plástico, que lleva básicamente el VDOS EPROM). Este módulo está preparado también para el alojamiento del punto de interconexión VORTEX RS-232-C. Si posee un ordenador CPC 464 + DDI-1, su controlador de Floppys DDI-1 "desaparece" en este módulo.

La unidad de disco y el grupo de red están alojadas en una caja metálica robusta.

Para conectar y desconectar la unidad de discos se dispone de un conmutador en el lado posterior de la caja metálica. Cuando este conmutador sobresale sólo un poco por fuera de la pared posterior, está conectada la unidad.

La clavija de enchufe a la red puede quedar conectada, aunque no se use el aparato durante mucho tiempo.

En todas las unidades X de disco se puede realizar opcionalmente la conmutación entre los sistemas operativos VDOS y AMSDOS.

**Al conectar el equipo, está siempre disponible inmediatamente el sistema operativo VDOS.**

En todos los equipos que comprenden un ordenador CPC 6128 tiene a su disposición alternativamente el sistema operativo CP/M 2.2 y también el CP/M Plus.

Si la unidad X de disco se utiliza con los sistemas operativos VDOS o bien CP/M (con el BIOS de VORTEX), se dispone de 128 registros de directorio (repertorio de direcciones).

La unidad de disco de 3" puede administrar un máximo de 64 registros de directorio en cualquier estado operativo (AMSDOS, VDOS, CP/M).

Si dispone de un equipo compuesto de CPC 464 + DDI-1 con unidad X de disco, deberá atenerse al siguiente orden sucesivo de conexión de los aparatos:

- 1.º Conectar el controlador de Floppys DDI-1.
- 2.º Conectar la unidad X de disco VORTEX.
- 3.º Conectar el ordenador CPC 464.

En los equipos con los ordenadores CPC 664 o CPC 6128 no es necesario atenerse a ningún orden sucesivo de conexión.

El equipo CPC 464 + DDI-1 con unidad X de disco dispone de la "función turnkey" (giro de llave y en marcha). Después de cada reconexión Reset (conexión/desconexión del aparato o pulsado simultáneo de las teclas **[CTRL]**, **[SHIFT]** y **[ESC]**), se busca un fichero o grupo

de datos "HELLO.BAS" en el disquete de 3" de la unidad de disco de 3". Cuando se ha encontrado, se pone en marcha. La búsqueda se puede interrumpir pulsando inmediatamente (después de la reconexión) por ejemplo, la tecla **ESC**.

Con ayuda de esta función, el sistema se puede avanzar directamente desde la puesta en marcha hasta un determinado estado operativo (por ejemplo, CP/M).

Todos los equipos con ordenadores CPC 664 y CPC 6128 carecen de esta función.

## **Puesta en marcha del ordenador CPC 464 + DDI-1 con unidad X de disco**

Desconectar de la red todos los enchufes.

La puesta en marcha de este equipo se efectúa en dos etapas:

- Montaje del controlador de Floppys DDI-1 en la caja del módulo VDOS (caja gris de plástico con forma de tronco de pirámide).
- Enchufar la unidad X de disco al cable de cinta plana del controlador DDI-1.

La única herramienta necesaria ahora es un pequeño destornillador de estrella. Eso es todo. Además, necesita también un poco de tranquilidad, para conseguir la puesta en marcha de su aparato.

- a) Desembale el módulo VDOS. En la parte inferior de la caja verá cuatro pequeños tornillos con cabeza de estrella. Suéltelos y guárdelos en un sitio seguro. Luego puede quitar la tapa.
- b) Obsevando la caja, verá una pequeña chapa (del lado de los componentes). Mirando perpendicularmente hacia dicha chapa, verá otra, que no lleva ningún componente y está provista de 50 "polos de contacto". Esta pequeña chapa se llama también "enchufe Card-Edge" (de bordes de tarjeta).
- c) A continuación ponga su controlador de Floppys DDI-1 (que habrá desenchufado previamente de su ordenador CPC; así como la unidad de disco de 3").

- d) Mirando al controlador, encontrará la pieza opuesta al “enchufe Card-Edge”, es decir, la “base de enchufe Card-Edge” (con la que se conecta también el controlador al ordenador CPC, sin la unidad X de disco).
- e) Observando detenidamente el módulo VDOS, verá a un lado, a través de una hendidura, un cable de cinta plana de 50 polos, que sobresale, con una base de enchufe Card-Edge.
- f) Entonces coja en su mano izquierda el módulo VDOS, de modo que pueda mirar en su interior desde abajo, quedando en la dirección opuesta a usted el cable de cinta plana de 50 polos mencionado en el párrafo anterior.
- g) Tome ahora en su mano derecha el controlador de Floppys DDI-1, de modo que la base de enchufe Card-Edge de 50 polos quede mirando hacia el módulo abierto y el cable de cinta plana de 34 polos quede orientado hacia la derecha.
- h) Enchufe el controlador DDI-1 en el módulo VDOS, de modo que la chapa pequeña vertical situada en él se encaje en la base de enchufe Card-Edge de 50 polos. Méntalo bien, apretando fuerte.
- i) Tome ahora otra vez la tapa de la base del módulo y colóquela en su sitio con los correspondientes tornillos, cerrando dicho módulo (con los cuatro tornillos de estrella antes mencionados, que usted había guardado).
- j) Verá, que VORTEX lo ha pensado bien todo: no hay ninguna segunda cajita bailando alrededor, sino una unidad compacta y libre de aditamentos que proporciona además espacio suficiente para el punto de interconexión VORTEX RS-232-C.
- k) El módulo VDOS se ha convertido ahora en un controlador completo de Floppys, con AMSDOS y VDOS. Enchúfelo en la parte posterior del ordenador CPC 464 en el “bus de expansión” de 50 polos (en el mismo sitio en el que estaba enchufado el controlador DDI-1). El módulo VDOS debe quedar situado, de modo que el cable de cinta plana de 34 polos quede mirando hacia la derecha (hacia el reproductor de cassettes).
- l) Coloque ahora a la derecha, junto a su ordenador CPC (en el lado con el reproductor de cassettes) la unidad X de disco (con la hendi-

dura para los disquetes mirando hacia adelante) y encima de ella la unidad de disco de 3".

- m) Ahora debe estar ya todo en orden. La regleta de bornes de 34 polos, situada en el extremo del cable de cinta plana se enchufa arriba, a la unidad de disco de 3", mientras que la clavija que sale de la unidad X se conecta a la segunda base del cable de cinta plana.
- n) Enchufar ahora todas las clavijas en sus bases de enchufe; luego a la unidad de disco de 3"; después la unidad X de disco (véase el pulsador situado en el lado posterior de la misma) y finalmente conectar el ordenador CPC. Todo terminado.

Con ello, su "nuevo" sistema de ordenador queda preparado para su funcionamiento.

Tenga en cuenta otra vez el orden sucesivo de conexión de los aparatos:

- 1.º Conectar el controlador de Floppys DDI-1
- 2.º Conectar la unidad X de disco
- 3.º Conectar el ordenador CPC 464.

Inmediatamente después de conectado el equipo, se obtiene la siguiente situación:

- El VDOS está conectado.
- La unidad de disco A es la unidad de 3".
- La unidad de disco B es la unidad X.
- La unidad de disco de 3" puede administrar como máximo 64 registros de directorio (repertorio de direcciones).
- La unidad X de disco puede administrar como máximo 128 registros de directorio.

## **Puesta en marcha de los ordenadores CPC 664/CPC 6128 con unidad X de disco**

Sacar de los enchufes todas las clavijas.

La puesta en marcha de estos equipos es muy sencilla. Saque del embalaje el módulo VDOS. Verá que sobresale un cable de cinta plana

de 50 polos; que lleva en su extremo una base de enchufe. Coloque el módulo VDOS detrás de su ordenador CPC, de modo que el cable quede mirando hacia usted y la marca VORTEX quede arriba. Enchufe ahora la base en el "bus de expansión" de 50 polos de su ordenador CPC.

Ahora hay que conectar todavía la unidad X de disco. En la parte posterior de su ordenador CPC encontrará una clavija de enchufe con la denominación:

#### Disc drive 2

y en la unidad X de disco un cable de 34 polos, en cuyo extremo hay una base de enchufe, que coincide exactamente con la clavija que lleva la denominación de "Disc drive 2". El enchufe hembra se mete de modo que el cable de cinta plana salga hacia arriba (obsérvese la etiqueta adhesiva "OBEN" = ARRIBA).

Ahora meter todas las clavijas en sus correspondientes bases, conectar el ordenador CPC y la unidad X de disco (véase el pulsador en la parte posterior de dicha unidad) y todo terminado.

Inmediatamente después de la conexión de cada equipo, se dispone de la siguiente situación:

- El VDOS está conectado.
- La unidad de disco A es la unidad de 3".
- La unidad de disco B es la unidad X.
- La unidad de disco de 3" puede administrar como máximo 64 registros de directorio (repertorio de direcciones).
- La unidad X de disco puede administrar como máximo 128 registros de directorio.

## **Discos flexibles, discos flexibles y siempre discos flexibles**

¿Qué discos flexibles (disquetes) necesitamos?

Todos los Floppys VORTEX están equipados con unidades de disco de 80 pistas y disponen de 2 cabezas de escritura/lectura (caras delantera y posterior del disquete).

Según que se haya decidido por el formato de 5.25" o de 3.5", para que la estación VORTEX pueda trabajar con seguridad y sin problemas necesitará disquetes del tipo siguiente:

Para las unidades de disco de 5.25": DS/DD/96tpi

Para las unidades de disco de 3.5" : DS/DD/135tpi.

DS significa DoubleSided (de dos caras). DD significa Double-Density (densidad doble de datos). 96tpi (o bien 135tpi) significa 96 (o bien 135) tracks per inch, es decir, 80 pistas.

Utilice únicamente disquetes con esta denominación. Cómprelos de buena calidad, ya que el material de mala calidad no sólo repercute en la seguridad de los datos registrados, sino que además puede dañar a las unidades de disco.

Para los disquetes de 5.25" se recomienda:

2/96 Flexydisk 5.25 de BASF

y para los de 3.5":

MF2DD 135tpi de FUJI FILM.

Ambos disquetes o discos flexibles han sido probados concienzudamente por nosotros en pruebas de larga duración.

#### **No deberá nunca:**

- Tocar directamente el disco magnético.
- Flexionar ni doblar el disquete.
- Someter el disquete a otra temperatura diferente a la indicada en el mismo.
- Exposar los disquetes a campos magnéticos.
- Escribir las etiquetas directamente sobre el disquete.
- Fumar en las proximidades (mejor sería que deje de fumar).
- Colocar sobre el disquete objetos puntiagudos pesados.
- Apilar muchos disquetes unos sobre otros y guardarlos apilados.

### **Lo que deberá hacer siempre:**

- Volver a guardar los disquetes en su funda de protección después de usarlos.
- Guardar los disquetes en posición vertical en una caja adecuada.
- Introducirlos cuidadosamente en la hendidura de la unidad de disco, sin doblarlos ni meterlos inclinados.
- Colocar los disquetes de modo que la etiqueta adhesiva quede hacia arriba y la abertura de escritura/lectura quede mirando hacia la unidad de disco y el disquete se encaje en su sitio.

Todos los puntos anteriores deben ser observados imprescindiblemente.

Una vez que el disquete está encajado en su sitio, pulsar fuertemente el botón de expulsión.

La “expulsión” del disquete, para que salga de la unidad de disco, se consigue volviendo a pulsar otra vez el botón de expulsión.

**Mientras esté trabajando la unidad de disco, no se debe extraer nunca el disquete.**

“VDOS” es una abreviatura de:  
“Vortex Disk Operating System”  
Sistema Operativo de Discos Vortex.

### **¿Qué es un “DOS”?**

“DOS” traducido al español significa sistema operativo de disco.

Un sistema operativo no es otra cosa que un programa, que se encarga de controlar un determinado equipo de hardware. El hardware es una combinación lógica de varios módulos electrónicos (circuitos integrados, chips). En el caso de su ordenador Amstrad, estos módulos son concretamente el microprocesador Z80, la memoria RAM (memoria de escritura/lectura), la memoria ROM (memoria de sólo lectura), el video controlador, el módulo de puntos de interconexión, el generador de sonido y algunos otros chips o circuitos integrados.



Para que este equipamiento o hardware funcione tal como suele hacerlo su ordenador y como usted espera, se necesita el sistema operativo. El sistema operativo trabaja sin interrupción, ocultamente, y tiene en su mano todos los hilos de control. Le proporciona las órdenes más sencillas, que le permiten manejar fácilmente los equipos, sin que tenga que llegar a comprender cómo funciona el conjunto del ordenador.

Una unidad de disco flexible (disquetes o Floppys) no es en realidad tampoco otra cosa que un equipo o hardware. Consta de las unidades de disco propiamente dichas, un grupo de red y un controlador, denominado interfaz o punto de interconexión. Para hacer que su unidad de disco funcione junto con su ordenador, se necesita una ampliación del sistema operativo, un "DOS". También aquí tiene validez la norma básica de que todo debe poderse manejar en la forma menos complicada posible, sin necesidad de conocimientos previos especiales.

Aunque hemos dicho que se trata de una ampliación del sistema operativo, esto es en realidad algo exagerado. El "DOS" es una cosa muy complicada, que por un lado debe manejar toda la unidad de disco y, por otro lado, debe incorporarse en el sistema operativo ya existente, de modo que no puedan producirse en ningún punto contradicciones ni fricciones.

El "VDOS", es decir, el Sistema Operativo de Disco Flexible de la firma VORTEX ha sido desarrollado especialmente para su ordenador Amstrad. Este sistema es capaz de alimentar unidades de disco con una alta capacidad (700K) y contiene todas las funciones que debe tener un sistema operativo profesional "DOS".

Una característica especial del VDOS es la administración relativa de ficheros de datos bajo el sistema BASIC. Si hasta ahora, bajo el sistema AMSDOS sólo tenía la posibilidad de utilizar simultáneamente dos ficheros o grupos de datos secuenciales, con el VDOS puede utilizar simultáneamente dos ficheros secuenciales y 16 ficheros relativos. Únicamente el acceso "opcional" a los datos de un fichero relativo, permite, por ejemplo, el desarrollo de un sistema profesional de banco de datos con el sistema BASIC.

El VDOS proporciona algo muy especial a los amigos de la programación de máquinas: el monitor o programa de control Z80 en lenguaje de máquina. Si con el sistema AMSDOS no existía ninguna posibilidad razonable de “mirar al interior del sistema”, con el VDOS esto ya no es ningún problema. El VDOS es un sistema operativo que supera las posibilidades del AMSDOS. Pero, ¡convéncase usted mismo!

## Fundamentos

Todas las órdenes del sistema operativo VDOS tienen una cosa en común:

Comienzan con una raya vertical. Esta raya le comunica al ordenador CPC que ahora llega una orden del sistema operativo VDOS y no una orden BASIC. Las órdenes que comienzan con una raya vertical se denominan también frecuentemente “órdenes RSX” u “órdenes externas” y no son ninguna especialidad del sistema operativo VDOS.

¿Cómo conseguimos la raya vertical?

Entre la tecla “P” y que tiene el signo de paréntesis recto “[” se encuentra la tecla con el signo de “arroba” “@”. Pulsando ésta y la tecla **[SHIFT]** aparece la raya vertical “|”.

Para comunicarle al ordenador CPC que lo que acabamos de introducir debe ser entendido como una orden, tenemos que pulsar la tecla **[ENTER]**. (En adelante, simbolizaremos mediante la expresión **[ENTER]**.) Por lo tanto, una introducción general de una orden tiene la siguiente configuración:

**|VDOS-ORDEN [ENTER]**

Por ejemplo, introduzca lo siguiente:

**|DIR [ENTER]**

Unidad A: usuario 0

TEST .BAS INSTALL .COM TEST .BIN RUNNER .BAS

Ocupados 60K; 4 grupos de datos; libres 109K

Ready (preparado).

Aquí se nos indica qué grupo de datos están en el disquete de 3" de la unidad de disco de 3".

En las órdenes del sistema VDOS (son órdenes externas) hay que tener en cuenta además otra característica especial. Si hay que transmitir también alguna variable, junto con una orden VDOS, en algunos casos habrá que señalar la variable con una "arroba" "@". En otros casos, se puede transmitir también directamente el contenido de las variables. Así, por ejemplo:

```
A$="*.BAS" [ENTER]
```

```
!DIR,@A$ [ENTER]
```

Unidad A: usuario 0

```
TEST .BAS RUNNER .BAS
```

Ocupados 60K; 2 grupos de datos; libres 109K.

En la orden **DIR** podríamos haber dicho también:

```
!DIR,"*.BAS" [ENTER]
```

y habríamos conseguido el mismo resultado que arriba.

En la orden **GET** (perteneciente a la administración relativa de grupos de datos), por el contrario, sólo se puede utilizar la forma de escritura con la "arroba":

```
NOMBRE$="otto"
```

```
LUGAR$="Essen"
```

```
TELE$="0123456"
```

```
!GET,1,173@NAME$,@ORT$,@TELE$ (nombre, lugar, teléfono)
```

La "arroba" no es el resultado de un "mal" sistema operativo, sino una peculiaridad del sistema BASIC, que está montado en su ordenador CPC. Aquí le indicaremos cuándo está permitida sólo la indicación de datos con la "arroba". En todos los demás casos, se pueden utilizar ambas formas, aunque es más agradable ciertamente la segunda posibilidad arriba indicada.

## **La orden ¡FORMAT del sistema operativo VDOS: formatear un disquete**

Antes de poder trabajar con un nuevo disquete, éste debe ser previamente preparado para ello. Esto se lo puede uno imaginar como si en el nuevo disco flexible vacío no hubiera absolutamente nada y fuera totalmente homogéneo. Para que la unidad de disco (y naturalmente también el VDOS) pueda hacer algo con este disquete, hay que grabar en el mismo una estructura básica.

Esta estructura básica consta de una determinada cantidad de pistas (Tracks) y sectores (Sectors). En lugar de pista se podría decir también línea circular. En las caras delantera y posterior de un disquete VORTEX se marcan en cada una 80 pistas de 9 sectores cada una. Además, en todas las zonas situadas entre las marcas se inscribe o graba un valor determinado. Se dice también que se formatea el disco.

Consideremos ahora lo que ocurre, si realizamos el formateado en un disco ya inscrito o grabado. Dado que al realizar el formateado no sólo se inscriben las marcas, sino también se “rellenan” los espacios existentes entre las marcas, esto tendría naturalmente un efecto catastrófico sobre los datos que eventualmente estuvieran ya registrados allí: se borrarían absolutamente todos. Así que, ¡mucho cuidado!

El formateado borra todos los datos que haya en el disquete

Coloque ahora un disquete totalmente nuevo en su unidad X de disco VORTEX.

El formateado se realiza con ayuda de la orden ¡**FORMAT** del sistema VDOS. Introduzca ahora esta orden:

**¡FORMAT,2** ENTER

Unidad B: disquete en unidad S/N?

S

Inicio cara, pista 0 a

Ready (preparado)

Con esto queda formateado este disquete.  
Ahora introduzca:

```
!B [ENTER]  
!DIR [ENTER]
```

Unidad B: usuario 0

Ocupados 4K, 0 grupos de datos; libres 704K.  
Ready (preparado).

Como ve, el disquete está totalmente vacío: "0 grupos de datos".

Ahora introduzca lo siguiente:

```
!A [ENTER]  
!FORMAT,1 [ENTER]  
FORMATO (S,D) ? S
```

Unidad A: disquete en unidad S/N ?  
S

Inicio cara, pista 0 a

Ready (preparado).

De este modo se formatea el disquete de 3" en la unidad de disco A (la unidad de 3"). Como ve, aquí se puede elegir también entre dos formatos (S/D): formato de Sistema y formato de Datos.

Luego se le planteará fundamentalmente la pregunta **FORMATO (S/D) ?** si "llega" a la unidad de 3" con **!FORMAT,1 [ENTER]** o bien, **!FORMAT,2 [ENTER]**.

La orden **!FORMAT,2 [ENTER]** "llega" a la unidad de 3", si ha cambiado previamente las unidades de disco A y B con la orden **!X** del sistema VDOS, de modo que entonces la unidad A sea la unidad X de disco VORTEX y la B sea la unidad de 3".

Introduzca "S" y entonces se formatea el disquete de 3" con 40 pistas de 9 sectores cada una (números de sector 41H-49H). Este formato es idéntico al "formato del sistema" y al "formato Vendor". Ambos formatos tienen dos pistas reservadas, pero sólo el formato de sistema contiene el sistema operativo CP/M. El formato Vendor está pre-

visto para el caso de que quiera vender software de CP/M desarrollado por usted mismo en el disquete de 3". Naturalmente, no podrá entregar estos disquetes junto con el sistema CP/M; por ello debe copiar su software en un disquete formateado en el formato Vendor ("formato de venta"), que contiene ciertamente las 2 pistas que pueden alojar el sistema CP/M, pero en el que falta el propio sistema CP/M. El cliente podrá entonces copiar muy sencillamente (por ejemplo, con **BOOTGEN** y **SYSGEN**) "su" sistema CP/M en este disquete.

Introduzca ahora "D" y entonces el disquete se formatea en el formato DATA-Only. Este formato tiene 40 pistas de 9 sectores cada una, pero no tiene pistas reservadas; por lo tanto no puede contener nunca un sistema operativo CP/M. La numeración del sector es C1H-C9H. Un disquete formateado en DATA-Only tiene una capacidad de memoria un poco más alta, debido a que faltan las pistas del sistema, que un disquete formateado en el sistema Vendor.

Mencionaremos aquí todavía que los discos se pueden formatear también en el sistema CP/M. Para ello se dispone en el mismo de un programa DISKTOOL.

## **Puesta en marcha y conservación de un programa BASIC**

Cuando conecta su ordenador CPC con la unidad X de disco VORTEX acoplada, recibirá, entre otras cosas la indicación siguiente:

© 1985 VDOS 2.0-X by vortex GmbH

Ready (preparado).

Dicha indicación nos confirma que la unidad X de disco está funcionando, existe el sistema operativo VDOS 2.0 y el ordenador CPC está esperando que realicemos una introducción (indicado por la palabra "Ready").

Ahora vamos a indicarle cómo puede proteger un programa BASIC, aunque seguramente ya lo sabe. Se trata de escribir en el disco, volver a cargarlo otra vez y ponerlo en marcha.

Introduzca simplemente el siguiente programa:

```
10 MODE 2
20 PRINT "**Esto es un programa de prueba **"
30 FOR A=1 TO 10
40 PRINT A
50 NEXT A
60 PRINT "** Esto es todo . . . **"
70 END
```

Si ahora desconecta el ordenador, se perderá todo el programa. En este caso no importaría mucho, puesto que se trata sólo de un programa de prueba, sin ninguna finalidad concreta. En cambio, si ha estado "ideando" un programa durante todo un fin de semana y no lo protege o asegura y luego desconecta el ordenador, empezará a buscar seguramente al culpable . . .

Por este motivo hay que asegurar el programa en el disquete. Luego se podrá volver a cargar otra vez en cualquier momento, aunque haya conectado y desconectado entre tanto el ordenador. Para este pequeño programa hemos pensado un nombre sencillo: por ejemplo, TEST (prueba).

No obstante, antes de asegurar el programa, tenemos que tener previamente un disquete en el que podamos realizar su conservación.

Tome un disquete totalmente nuevo (recién sacado de su embalaje); todavía hay que formatearlo, tal como se ha indicado en la orden **!FORMAT**. Introduzca el disquete en la unidad X de disco VORTEX y escriba la orden **!X [ENTER]**. (De este modo, la unidad A es la unidad X de disco y la B es la unidad de 3".)

Ahora podemos ya almacenar nuestro programa en el disquete. De todos modos, naturalmente, no es necesario que utilice cada vez un disquete nuevo.

Además, debe introducir la orden siguiente:

```
SAVE "TEST" [ENTER]
```

Entonces se pone en marcha la unidad de disco, se almacena en la memoria el programa y ya está terminado. Ahora introduzca la orden siguiente:

```
CAT [ENTER]
```

Unidad A: usuario 0

```
TEST .BAS 1K
```

Ocupado 1K, 1 grupo de datos; libres 700K

En el disquete volvemos a encontrar otra vez nuestro programa de prueba "TEST". Para comprobarlo, introduzca ahora la siguiente orden:

```
NEW [ENTER]
```

```
LIST [ENTER]
```

Ready (preparado).

Pues bien, ahora hemos borrado nuestro programa. Pero, no tenga miedo, lo hemos protegido. Recupérelolo de nuevo con la línea siguiente:

```
LOAD "TEST" [ENTER]
```

Ready

```
LIST [ENTER]
```

```
10 MODE 2
```

```
20 PRINT "*" Esto es un programa de prueba *
```

```
30 FOR A=1 TO 10
```

```
40 PRINT A
```

```
50 NEXT A
```

```
60 PRINT "*" Esto es todo ... *
```

```
70 END
```

Efectivamente, aquí esta otra vez el programa. Ahora ponemos en marcha el programa con la orden **RUN** [ENTER]. Entonces se borra la pantalla y luego sale esencialmente las cifras 1 a 10; no es nada extraordinario, sólo algo para la demostración. También podría haber introducido directamente la orden **RUN "START"** [ENTER]. Entonces



se habría cargado el grupo de datos **TEST.BAS** del disquete y se habría puesto en marcha inmediatamente.

Todavía un par de palabras respecto a la orden **SAVE**. Existen cuatro posibilidades de utilizar esta orden:

1. **SAVE "DATEI"**  (DATEI = FICHERO, grupo datos)
2. **SAVE "DATEI",A**
3. **SAVE "DATEI",P**
4. **SAVE "BILD",B,&C000,&4000**  (BILD = pantalla, imagen)

La primera variante ya la hemos comprendido arriba. El programa se almacena aquí en "forma comprimida", que requiere muy poco espacio de almacenamiento del disquete.

Con la segunda posibilidad se coloca un grupo de datos llamado ficheros ASCII. Se trata de un grupo de datos únicamente con texto, que no contiene ningún tipo de signos especiales, etc. El programa se deposita efectivamente tal como lo hemos visto en el listado de la pantalla.

Con la tercera posibilidad se almacena el programa "protected" (protegido). Ya no es "LISTbar" = (listable).

La última posibilidad incluye lo siguiente: todos los datos existentes en la memoria entre las direcciones C000 y FFFF (= C000+4000), es decir, el contenido de la pantalla, se depositan en el fichero BILD.BIN (BILD = pantalla; BIN = binario).

## ¿Qué es un fichero o grupo de datos?

¿No lo sabe? No importa, se lo explicaremos brevemente. Suponga que tiene en su oficina un armario archivador, en el que hay una serie de carpetas o archivadores. En cada archivador están ordenados los papeles de una determinada operación. Si no es así, tendrá un archivo desolador.

La similitud con nuestro caso es la siguiente: el armario archivador es el disquete; las carpetas o archivadores son los ficheros o grupos

de datos y los papeles o documentos son los datos que se almacenan en un fichero.

Si coloca una etiqueta en cada archivador, indicando su contenido, para poder volverlo a encontrar con facilidad, cada grupo de datos recibirá una denominación. Grupo de datos o fichero equivale en inglés a la palabra FILE.

La denominación del fichero o grupo de datos está sometida a ciertas limitaciones: por ejemplo, no está permitida una denominación como, "este es mi primer fichero". La denominación del fichero o grupo de datos admitida por el sistema VDOS (y también por el CP/M) es la siguiente:

Nombre del fichero. Tipo de fichero.

El nombre del fichero puede tener hasta 8 signos, mientras que para el tipo de fichero se permiten como máximo 3 signos. Es obligatorio colocar un punto entre el nombre y el tipo.

En la denominación de un fichero puede utilizar todos los signos, excepto los siguientes:

< > . ; : = ? \* [ ] % | ( ) / \

El tipo de fichero proporciona frecuentemente una cierta orientación sobre el contenido del fichero. Ejemplos de denominaciones de ficheros son, por ejemplo: TEST.BAS, SPIELE (juegos), BILD.BIN, PROGRAMM.BAS.

En el sistema BASIC se encontrará generalmente con los siguientes tipos de ficheros (en inglés se les denomina también "extensión", es decir extensiones):

**BAS:** esto es un programa BASIC, que se pone en marcha con **RUN** o que se puede cargar con **LOAD**, por ejemplo: **SPIEL.BAS** o bien **PROGRAMM.BAS**.

**BAK:** fichero de seguridad o fichero BACKUP. Si desea almacenar un programa BASIC con **SAVE**, el sistema operativo VDOS "investigará" en el disquete, si existe ya allí un fichero con la denominación elegida por usted. Si ya existe uno con dicho nombre, su tipo será cambiado por BAK (antes era BAS). Sin

embargo, el fichero que quiere almacenar realmente recibe el tipo BAS. Por ejemplo:

**SAVE "TEST" [ENTER]**.

Este fichero o grupo de datos se deposita en el disquete con la denominación TEST.BAS. Sin embargo, allí hay ya un grupo de datos que se llama TEST.BAS. Entonces, el sistema VDOS le cambia simplemente el nombre por TEST.BAK. Por consiguiente, no puede ocurrir nada, si borra por equivocación un grupo de datos.

**BIN:** BIN significa "binario". El contenido de un fichero o grupo de datos BIN puede ser, por ejemplo, una imagen completa de la pantalla o también un programa de máquina. Un grupo de datos de este tipo se coloca, por ejemplo, mediante la orden **SAVE** siguiente:

**SAVE "BILD",B,&C000,&4000 [ENTER]**

En el disquete encontramos el grupo de datos BILD.BIN. La "B" situada después de "BILD" obliga a emplear un grupo de datos binario.

Si hay que cargar un grupo de datos y no se indica ningún tipo de fichero al dar la orden de carga, se carga con el nombre de fichero indicado el primero encontrado, después de la búsqueda por orden sucesivo de " ", "BAS", "BIN" y "BAK".

## **Las órdenes !A y !B del sistema operativo VDOS: conmutación entre las unidades de disco**

En el sistema VDOS dispone como máximo de 2 unidades de disco. Les hemos llamado "unidad de disco A" y "unidad de disco B" (también les llamamos abreviadamente A y B).

La característica especial del equipo que posee es que tiene unidades de disco de diferentes formatos (3" y 5.25" o bien 3" y 3.5") con distintas capacidades (aproximadamente 170K y 708K).

**Después de conectar el ordenador CPC y la unidad X de disco, la unidad A es la unidad de disco de 3" y la unidad B es la unidad X de disco.**

Las órdenes introducidas se refieren entonces en la unidad A a la de 3”:

**| DIR** **[ENTER]**

Unidad A: usuario 0

TEST .BAS TEST .BIN  
PROGRAMM .BAS TEST .BAK

Ocupados 100K; 4 ficheros; libres 69K

Si deseamos ver qué hay en el disquete colocado en la unidad X de disco, tendremos que realizar entonces la conmutación a esta unidad:

**| B** **[ENTER]**

Las órdenes introducidas “actúan” entonces en la unidad B sobre la unidad X de disco:

**| DIR** **[ENTER]**

Unidad B: usuario 0

PIP .COM M80 .COM VDOS .MAC  
STAT .COM L80 .COM WS .COM  
WSMSGs .OVR. WS0VLY1 .COM

Ocupados 304K; 8 ficheros; libres 400K

Como ve, tenemos a nuestra disposición un espacio de almacenamiento considerablemente mayor. Ahora volvemos a conectar la unidad de 3”:

**| A** **[ENTER]**

Nos podemos imaginar, por ejemplo, el siguiente programa:

```
10 MODE 2
20 PRINT "Esto es el directorio del disquete en la unidad A"
30 CAT
40 |B
50 PRINT "Esto es el directorio del disquete en la unidad B"
60 CAT
70 |A
80 END
Ready
RUN [ENTER]
```

Una vez que pone en marcha este programa (a través de la orden **RUN**), se borra la pantalla y aparecen sucesivamente los índices de contenido de los disquetes de la unidad A (disquete de 3") y de la unidad B (disquete de 3.5" o de 5.25"), acompañados del respectivo texto. Finalmente, se vuelve a conmutar otra vez a la unidad A (línea 70).

Si no sabe lo que significa "MODE 2, PRINT y END", antes de continuar leyendo aquí, debería tomar primeramente el manual BASIC, que se le entregó junto con su ordenador CPC y aprender allí "determinados" conocimientos fundamentales.

### **La orden !X del sistema operativo VDOS: intercambiar las unidades de disco A y B**

Conecte todos los aparatos, coloque un disquete en la unidad de 3" y en la unidad X de disco VORTEX; introduzca ahora la orden **CAT** **[ENTER]** y recibirá entonces el índice de contenido del disquete de 3". Introduzca las órdenes **!B** **[ENTER]** y **CAT** **[ENTER]** y entonces recibirá el índice de contenido del disquete en la unidad X de discos VORTEX.

El estado de conexión del ordenador CPC y de la unidad X de discos es, por lo tanto:

Unidad A: unidad de 3" (montado en el ordenador CPC 664 y CPC 6128 y externamente en el DDI-1 en el ordenador CPC 464).

Unidad B: unidad de disco VORTEX (5.25" en el modelo F1-X y 3.5" en el modelo M1-X).

Pero, introduzca ahora la orden:

**!X** **[ENTER]**

y entonces se intercambian ("eXchange") la unidad de disco A con la B. Es decir, se produce una nueva coordinación:

Unidad A unidad de disco VORTEX (5.25" en el modelo F1-X y 3.5" en el modelo M1-X).

Unidad B: unidad de 3" (incorporado en el ordenador CPC 664 y CPC 6128 y en el exterior del DDI-1 en el ordenador CPC 464).

Vuelva a conmutar ahora otra vez a la unidad A (antes habíamos conmutado a B), introduciendo la orden ¡A [ENTER]. Ponga ahora en marcha CAT [ENTER] y obtendrá el índice de contenido del disquete en la unidad X de disco VORTEX (ahora en la unidad A).

Ahora dé marcha atrás al ordenador, para tener un estado claro de partida, pulsando simultáneamente las teclas [CTRL], [SHIFT] y [ESC].

Coloque ahora un disquete de 3" en la unidad de 3" y un disquete de 5.25"/3.5" en la unidad X de disco. Ambos disquetes deben ya tener un contenido.

Observe el programa siguiente:

```
10 MODE 2
20 PRINT "Esto es el contenido del disquete de 3" en la uni-
dad A"
30 CAT
40 CLS
50 ¡B
60 PRINT "Esto es el contenido del disquete de 5.25"/3.5""
70 PRINT "en la unidad B"
80 CAT
90 ¡A:REM conmutar otra vez a la unidad A (unidad de 3")
100 ¡X:REM intercambiar las unidades A y B
110 CLS
120 PRINT "Esto es el contenido del disquete de 5.25"/3.5""
130 PRINT "en la unidad A"
140 CAT
150 ¡B
160 PRINT "Esto es el contenido del disquete de 3" en la uni-
dad B"
170 CAT
180 ¡A:REM conmutar otra vez a la unidad A (unidad de dis-
co X).
```

## Muy importante

Si pone en marcha el sistema operativo CP/M o hace retroceder su ordenador hasta el punto de partida (pulsando simultáneamente las teclas **CTRL**, **SHIFT** y **ESC**), para conexión/desconexión del ordenador), la unidad A será otra vez la unidad de 3" y la unidad B será la unidad de disco X. Esto significa que la configuración de las unidades de disco en el sistema operativo CP/M se fija al cargar (boot) el sistema CP/M.

## Las órdenes **CAT** y **!DIR** del sistema operativo **VDOS**: para examinar el índice de contenido del disquete

Así como en cada libro hay un índice, en el que se resumen por orden sucesivo los títulos de los capítulos, también cada disquete posee un índice de contenido, en el que se incluyen todos los grupos de datos o ficheros existentes en dicho disquete.

En lugar de índice se utiliza generalmente la palabra "directorio" (Directory) y frecuentemente se usa también la palabra "catálogo".

En el sistema operativo **VDOS** hay dos órdenes, con las que se puede hacer que aparezca en la pantalla el índice de contenido del disco.

Una de estas órdenes ya la hemos usado anteriormente con frecuencia: es la orden **!DIR** (abreviatura de **DIR**ectorio). La otra se denomina **CAT** (abreviatura de **CAT**álogo), que tiene una pequeña peculiaridad. Para esta última orden no se necesita utilizar la raya vertical "|", debido sencillamente a que esta orden la encontrará también sin la unidad de disco X **VORTEX** en su ordenador **CPC** y, como ya hemos dicho anteriormente, únicamente hay que emplear la raya vertical para las órdenes externas (órdenes adicionales, que entran al conectar el **Floppy** u otros aparatos periféricos).

La más sencilla de estas dos órdenes es **CAT**. Aquí puede ver su aplicación:

**CAT** **ENTER**

Unidad A: usuario 0

\$OSC	.SYS	3K	PIP	.COM	8K#
FAST	.COM	1K	READ	.BAK	1K
HELLO	.BAS	1K	READ	.ME	1K
MAN	.003	19K	STAT	.COM	6K
MAN	.004	27K	VDOS	.BAK	6K
MAN	.005	5K	WS192N	.COM	16K
MAN	.006	1K	WSMSGs	.OVR	30K
MAN	.007	15K	WSOVLY1	.OVR	34K
MAN	.BAK	15K			

Ocupados 228K; 17 ficheros o grupos de datos; libres 480K.

Ready (preparado)

Este es, por ejemplo, el contenido de mi disquete de trabajo Word-Star. Con él obtenemos informaciones sobre el tamaño de los diversos grupos de datos, cuántos grupos de datos se encuentran en el disco y cuánto espacio de almacenamiento queda todavía libre o se ha utilizado ya. Además, los grupos de datos están por orden alfabético.

Tenga en cuenta una pequeña advertencia. La unidad de disco X VORTEX puede contener un máximo de 128 ficheros o grupos de datos por cada disquete. Con una unidad de disco de 3" sólo puede contener 64 grupos de datos.

**El Floppy VORTEX puede contener 128 registros en el índice. El Floppy de 3" sólo 64 registros.**

Así pues, **CAT** es una orden sencilla, con la que se pueden obtener rápidamente informaciones importantes.

Sin embargo, a esta orden le falta una propiedad importante: el "Filtrado de depuración" de determinados ficheros. En este caso resulta interesante la orden **!DIR**. En relación con esta propiedad de filtrado, explicaremos aquí brevemente el significado del concepto "Wilcard":

Puede ocurrir frecuentemente que usted utilice el mismo disquete para los sistemas operativos BASIC, VDOS y CP/M. Por lo tanto, en



este disquete habrá grupos de datos que no pueda utilizar en el sistema CP/M, o al revés. Ya hemos dicho anteriormente que, por ejemplo, los ficheros del tipo BIN son típicos del sistema BASIC, mientras que un fichero con el tipo COM es un grupo de datos típico del sistema CP/M. De todos modos, sería muy ventajoso el poder “buscar” sólo ficheros del tipo BIN en el disquete con el sistema BASIC, es decir, que sólo deberían aparecer en la pantalla estos ficheros o grupos de datos. Por tanto, necesitamos una orden, con la que sea posible reproducir selectivamente determinados ficheros o grupos de ficheros. Precisamente con esta finalidad disponemos de la orden ¡DIR. Vea su aplicación:

```
!DIR, "*.COM" [ENTER]
```

Unidad A: usuario 0

```
FAST .COM PIP .COM  
STAT .COM WS192N .COM
```

Ocupados 228K; 4 ficheros; libres 480K.

Se han visualizado todos los ficheros o grupos de datos del tipo COM.

```
!DIR, "MAN.*" [ENTER]
```

Unidad A: usuario 0

```
MAN .003 MAN .004  
MAN .006 MAN .005  
MAN .007 MAN .BAK
```

Ocupados 228K; 6 ficheros; libres 480K.

Se han visualizado todos los ficheros que llevan el nombre “MAN”.

```
!DIR, "MAN.00?" [ENTER]
```

```
MAN .003 MAN .004  
MAN .006 MAN .005  
MAN .007
```

Ocupados 228K; 5 ficheros; libres 480K.

Aquí se han reproducido en la pantalla sólo los ficheros o grupos de datos, cuyo nombre es MAN y cuyo tipo tiene una longitud de 3 signos, debiendo ser "00" los dos primeros signos, mientras que el tercer signo puede ser opcional, pero debe estar siempre presente (por ejemplo, un fichero MAN.00 no resultaría reproducido en la pantalla).

Los dos signos "\*" "?" son signos llamados Wildcard (tarjeta libre). Un "?" permite siempre opcionalmente sólo el signo al que sustituye (pero debe estar presente).

Un asterisco "\*" permite opcionalmente todavía todos los demás signos restantes.

La diferencia entre ambos signos se demuestra también mediante la siguiente ecuación (no matemática):

?????????.??? = \*.\*

Como seguramente ha notado, en las órdenes CAT y !DIR aparece siempre en la primera línea el aviso:

Unidad A: usuario 0.

¿Qué importancia tiene ésto? Muy sencillo. Mediante la indicación "unidad A" se nos indica que aquí se reproduce un índice del disquete colocado en la unidad A. Es uno de los 16 posibles índices. Al desarrollar el sistema operativo VDOS se ha pensado ya en que, en determinadas circunstancias, varios usuarios deberán repartirse un sólo disquete. A fin de que entonces no se produzca ningún desorden (qué fichero le corresponde a cada uno), el disquete se puede subdividir en hasta 16 sectores de usuario (sectores de USER). (Véase también la orden !SELECT.) No obstante, hay que tener en cuenta que la suma de todos los ficheros o grupos de datos (de todos los sectores de usuarios) no deberá ser nunca superior a 128 y que la suma de todos los valores de los grupos de datos no debe sobrepasar nunca los 704K. En ambos casos estará totalmente lleno el disquete. En la unidad de 3", la suma de todos los ficheros no debe sobrepasar la cantidad de 64K. La suma de todos los tamaños de ficheros debe ser inferior a 169K.

## La orden **!ATTRIBUT** del sistema operativo VDOS: para proteger un fichero o para “hacer invisible” un fichero o grupo de datos.

Seguramente ya conoce la posibilidad de proteger contra “grabación” un disco completo (borrar es también escribir o grabar). Se trata de una protección mecánica: en el disquete de 5.25” se consigue simplemente mediante una etiqueta adhesiva alrededor de su borde; en los disquetes de 3” y de 3.5” por medio de un pequeño travesaño de bloqueo en el propio disquete.

Pero, si desea proteger contra la grabación algún fichero o grupo de datos concreto, necesitará para ello la orden **!ATTRIBUT** del sistema operativo VDOS. Esta orden tiene por objeto dotar a los ficheros o grupos de datos de un “atributo”. El atributo proporciona informaciones sobre el “estado” del fichero. El sistema operativo VDOS distingue los siguientes atributos habituales:

- S El fichero o grupo de datos es un fichero del tipo “SYS” o “sistema”. Este fichero no es reproducido en la pantalla al introducir las órdenes **CAT** y **!DIR**; es “invisible”.
- D Este fichero es del tipo “DIR” o “directorio, índice”. Es indicado en la pantalla con las órdenes **CAT** y **!DIR**. Si no modifica nada en los atributos del fichero o grupo de datos, todos los ficheros tendrán este atributo y el siguiente:
- W El fichero es R/W (read/write), es decir, legible/escrible. También puede ser borrado.
- R El fichero es R/O (read only), es decir, sólo para lectura y, por lo tanto, está protegido contra escritura. Por consiguiente, tampoco puede ser borrado.

Como ya hemos indicado, todos los ficheros o grupos de datos están provistos de los dos atributos “D” y “W” en su “estado original” (sin que haya modificado nada en los mismos). La orden **!ATTRIBUT** se utiliza en la forma siguiente:

```
!ATTRIBUT,“PIP.COM”,“R” ENTER
```

Este fichero "PIP.COM" se marca como "sólo legible". Por tanto, no puede ser ya borrado nunca. Ahora, inspeccionemos el disquete con la orden **CAT** y veremos que este fichero ha quedado marcado.

**CAT**

Unidad A: usuario 0.

\$OSC	.SYS	3K	PIP	.COM	8K #
FAST	.COM	1K	READ	.BAK	1K
HELLO	.BAS	1K	READ	.ME	1K
MAN	.003	19K	STAT	.COM	6K
MAN	.004	27K	VDOS	.BAK	6K
MAN	.005	5K	WS192N	.COM	16K
MAN	.006	1K	WSMSGs	.OVR	30K
MAN	.007	15K	WSOVLY1	.OVR	34K
MAN	.BAK	15K			

Ocupados 228K; 17 ficheros; libres 480K.

Ready (preparado)

Todos los ficheros marcados con "#" están protegidos contra escritura. Aventurémonos un poco en ellos: intente borrar el fichero "PIP.COM" introduciendo la línea siguiente:

**!ERA."PIP.COM"**

PIP .COM sólo legible

Ready.

Verá que este fichero no se puede borrar, de todos modos, debe tener en cuenta que esta "protección" naturalmente queda sin ningún efecto, si se utiliza la orden **!FORMAT**. Con esta orden se anula rigurosamente todo lo que exista anteriormente en el disquete.

Naturalmente, también se puede anular otra vez este atributo:

**!ATTRIBUT,"PIP.COM","W"**

El fichero o grupo de datos puede ser borrado otra vez.

## La orden ¡ERA del sistema operativo VDOS: borrar un fichero o grupo de datos

Con esta orden se pueden borrar ficheros, grupos de ficheros o también disquetes enteros. Esta orden se refiere siempre al campo o sector de usuario (USER) ajustado en cada momento (véase ¡SELECT). ERA es la abreviatura de "ERASE" (que significa: borrar). Con la línea siguiente:

```
¡ERA, "TEST.BAS" [ENTER]
```

se borra y elimina el grupo de datos TEST.BAS del disquete colocado en este momento. Si el grupo de datos hubiera estado provisto del atributo "R", no habríamos podido borrarlo. Si no existe este fichero o grupo de datos en este disquete, el sistema operativo VDOS nos indica:

```
TEST .BAS no ha sido encontrado.
```

Con la línea siguiente:

```
¡ERA, "TEST.*" [ENTER]
```

se borran en el campo o sección momentánea del usuario (USER) en este disquete todos los ficheros o grupos de datos (por ejemplo, TEST.BAS, TEST, TEST.COM, TEST.123, TEST.12). Además de este signo de tarjeta libre "\*" se puede utilizar también el otro "?".

La línea siguiente sólo deberá ser introducida a plena conciencia y con el máximo cuidado, puesto que ocasiona el borrado completo de todo el disquete:

```
¡ERA, "*.*" [ENTER]
```

En esta tarjeta libre Wildcard (\*.\*) "cabén" todos los ficheros y, por lo tanto, son borrados también. Hay que tener siempre mucho cuidado. Seguramente son también muy raros los casos en los que haya que borrar disquetes enteros.

### ATENCIÓN

¡Con ¡ERA, "\*.\*" [ENTER] se borra el disquete entero!

También podría hacerse lo siguiente:

```
!B [ENTER]  
!ERA,"A:PIP.COM" [ENTER]
```

Con ello hemos realizado primeramente la conmutación a la unidad B y luego, desde allí, hemos borrado en el disquete situado en la unidad A el fichero o grupo de datos PIP.COM.

### **La orden !REN del sistema operativo VDOS: modificación de la denominación de un fichero o grupo de datos**

Seguramente ocurrirá alguna vez, que ya no le gusta el nombre dado anteriormente a un fichero, o que lo necesita en otro sitio. Para poder variar la denominación de un fichero, tenemos la orden **REN** (REN es la abreviatura de "Rename", que significa cambiar de nombre). Aquí no se permiten tarjetas libres Wildcard, porque no son convenientes.

```
!REN,"FICHERO.NUEVO", "FICHERO.ANTIGUO" [ENTER]
```

El grupo de datos con la denominación "FICHERO.ANTIGUO" se cambia de nombre y recibe el de "FICHERO.NUEVO". Si ya existe un grupo de datos con la denominación "FICHERO.NUEVO", se interrumpirá la orden y el sistema operativo VDOS nos indicará:

FICHERO .NUEVO ya existe.

Entonces habrá que borrar primeramente este grupo de datos o copiarlo en algún otro sitio.

Si no existe el grupo de datos "FICHERO.ANTIGUO", el sistema operativo VDOS nos anuncia:

FICHERO .ANTIGUO no ha sido encontrado.

La línea siguiente

```
!REN,"B:TEST.123","B:SPIEL.BAS" [ENTER]
```

realiza desde la unidad de disco A el cambio de nombre del grupo de

datos SPIEL.BAS, que se encuentra en el disquete situado en la unidad B, dándole el nombre de grupo de datos TEST.123, también en la unidad de disco B. Observe que un cambio de denominación sólo se puede realizar siempre dentro de un mismo y único disquete. Una línea como, por ejemplo:

`!REN,"A:DDT.COM","B:SID.COM" [ENTER]` no tendría ningún sentido.

### **La orden !SELECT del sistema operativo VDOS: ajuste de los campos del usuario USER**

En el sistema operativo VDOS (y también en el CP/M) se puede subdividir el disquete en hasta 16 sectores del usuario. Esta subdivisión puede ser interesante, cuando utilicen varios usuarios conjuntamente un disquete y no se desee llegar a una zona exclusiva de otro usuario.

No obstante, también se pueden hacer varios sectores de usuario, aunque sea uno sólo el usuario de este disquete. De este modo, es posible obtener una estructura clara y diferenciada en el disquete. De todos modos, se utiliza poco esta orden, limitándose al sector 0 del usuario. La siguiente línea:

`!SELECT,"6" [ENTER]`

efectúa la conmutación al sector 6 del usuario.

Ahora introduzcamos una orden **CAT**, que llevará en su línea superior la información:

Unidad A: usuario 6

que nos proporciona en la pantalla la lista de grupos de datos o ficheros que se encuentren en este sector del usuario.

Además, aparte del número del usuario se puede indicar también al mismo tiempo la referencia de la unidad de disco (A o bien B).

`!SELECT,"B5" [ENTER]`

realiza la conmutación a la unidad de disco B en el sector 5 del usuario.

Ahora introduzcamos, por ejemplo:

```
!CAT 
```

y obtendremos, como primera línea del “catálogo”:

unidad B: usuario 5

Si se utiliza aquí la forma de escritura con la “arroba” “@”, se puede indicar “dinámicamente” dentro de un programa BASIC el número del usuario y la referencia de la unidad de disco:

```
10 INPUT "Referencia de la unidad de disco", A$
20 INPUT "Número del usuario", B$
30 C$=B$+A$
40 !SELECT,@C$
50 CAT
```

Dependiendo de la referencia de la unidad de disco y del número de usuario introducidos, obtendremos el índice de contenido del sector correspondiente en el respectivo disquete.

## **La orden !RESET del sistema operativo VDOS: cambio de disquetes**

Suponga que ha abierto para escribir un fichero o grupo de datos y que de vez en cuando se van incluyendo y escribiendo datos en este fichero. Si entretanto, cambia el disquete, esto podrá conducir a una pérdida de los datos, porque el fichero está todavía “abierto”.

Esta orden sirve para comunicarle al sistema operativo VDOS que ha cambiado el disquete en la misma unidad de disco durante una operación de escritura.

Conviene que se acostumbre a introducir esta orden antes de cambiar el disquete, sobre todo, si no está seguro de que haya todavía un fichero abierto para escribir. Si así fuera y olvida la orden de !RESET , podrían perderse quizás los últimos datos de 64K. Si ha cerrado adecuadamente (CLOSEOUT, !CLOSE) el fichero, antes de cambiar el disquete, resultará superflua la orden !RESET.



Si ha estudiado ya un poco el sistema operativo CP/M, encontrará aquí un cierto parentesco con la orden **CTRL C** del sistema CP/M.

## **La orden ¡CODE del sistema operativo VDOS: protección del programa**

Anteriormente hemos mencionado la posibilidad de proteger un programa con la orden **SAVE** y con la “P” (protección simple P) antes de realizar el “listado”. Esta protección (que posee también sin el Floppy VORTEX) resulta lamentablemente muy poco efectiva, porque ya se ha “roto” antes de tiempo.

Fundamentalmente se puede decir que no existe ninguna protección, que no resulte atravesable. Se puede hacer más difícil romper la protección, pero no se puede evitar absolutamente.

Con la orden ¡CODE del sistema operativo VDOS puede montar una protección personal del programa. Por ejemplo, si desea proteger el programa TEST.BAS con las líneas siguientes:

```
¡CODE,12325 [ENTER]  
SAVE "TEST",P [ENTER]
```

ha puesto una clave personal de codificación al TEST (12325).

Introduzca ahora la orden ¡CODE,0 [ENTER]. Con ella, o también haciendo retroceder el ordenador hasta el principio (pulsar simultáneamente las teclas [CTRL], [SHIFT] y [ESC]) activará la protección.

Introduzca ahora la orden RUN "TEST" [ENTER]. Ahora saldrá algo totalmente absurdo, porque no se había indicado “clave”.

Por ello, introduzca la orden ¡CODE,12325 [ENTER] y ponga de nuevo en marcha TEST. Ahora todo funciona correctamente.

La cifra en la orden de **CODE** debe estar entre -32768 y 32767. Esto supone entonces nada menos que 65536 posibilidades de proteger su programa.

## **La orden !AMSDOS del sistema operativo VDOS: conexión del sistema operativo AMSDOS**

Introduciendo la orden **!AMSDOS** **[ENTER]** se desconecta el sistema VDOS y se conecta al mismo tiempo el AMSDOS.

Si posee programas que no puedan trabajar en el sistema VDOS sin una adaptación previa, puede efectuar la conmutación a AMSDOS y así se solucionan todos los problemas. Si ha efectuado ya la conmutación a AMSDOS, es como si hubiera conectado una segunda unidad de disco a su ordenador CPC. Esta orden realiza un RESET. Es decir, se borran todos los datos existentes anteriormente en la memoria.

### **La orden !AMSDOS [ENTER] borra todos los datos existentes en la memoria**

Si ha realizado la conmutación al sistema AMSDOS, deberá tener en cuenta que entonces la unidad X de disco no tendrá ya la capacidad de almacenamiento de 708K, sino sólo de 169K (igual que la unidad de disco de 3") formateada. Esto se debe a que el sistema AMSDOS sólo puede gestionar unidades de disco de 40 pistas y una cabeza de escritura/lectura. (De aquí resulta también una capacidad de almacenamiento de aproximadamente 170K.)

Todas las unidades de disco X poseen 80 pistas con dos cabezas de escritura/lectura. En el sistema AMSDOS, el disco flexible, situado en la unidad de disco X se maneja sólo por una cara y únicamente con 40 pistas. Sin embargo, hay que tener en cuenta entonces que las 40 pistas no están distribuidas uniformemente por todo el radio del disquete, sino que utilizan sólo la mitad exterior del mismo. Por lo tanto, es como si la unidad de disco trabajara con 80 pistas, pero sólo utilizara realmente las 40 primeras.

Es importante que sepa que un disquete, preparado bajo el sistema AMSDOS en la unidad de disco X, no se puede utilizar nunca con el sistema VDOS en esta unidad.

Un disquete fabricado con el sistema VDOS no se debe utilizar nunca y en ningún caso con el sistema AMSDOS.

Un disquete fabricado con el sistema AMSDOS, a su vez, no se puede utilizar tampoco en ningún caso con el sistema VDOS.

Si no tiene usted en cuenta estas advertencias, se destruirá irremediablemente el contenido de los disquetes correspondientes.

Mediante la puesta a cero del ordenador CPC (pulsando simultáneamente las teclas **CTRL**, **SHIFT** y **ESC**) se conecta otra vez el sistema operativo VDOS.

## **Reproductor de cintas en el ordenador CPC**

El primero de los ordenadores CPC, el CPC464 está equipado en serie con un reproductor de datos. Los otros ordenadores CPC664 y CPC6128 pueden ser equipados posteriormente también con un reproductor de datos.

Si se dispone adicionalmente de una unidad de reproducción de disquetes, entonces será importante poder transferir datos entre el disquete y la cinta.

En los ordenadores CPC hay un canal de entrada y un canal de salida entre el ordenador y la memoria de almacenamiento (Floppy o reproductor de datos). Si se ha conectado un Floppy y un reproductor de datos y se desea, por ejemplo, traspasar datos del Floppy al reproductor, únicamente hay que colocar en éste el canal de salida y en el Floppy el de entrada.

Hay que mencionar aquí, que una vez conectado el ordenador CPC con el Floppy acoplado, se le adjudican canales de entrada y de salida, aunque exista todavía el reproductor de datos.

Existen exactamente seis posibilidades de "conectar entre sí" los dos canales. Para cada uno de los casos se dispone de una orden respectiva del sistema VDOS.

## **La orden !DISC del sistema operativo VDOS**

Los canales de entrada y de salida están asignados a la estación de disquetes. Éste es el "estado de conexión" del ordenador CPC con el Floppy acoplado. Cuando se leen o escriben datos, son leídos desde el Floppy o bien son escritos sobre el disquete situado en él.

## **La orden !CAS del sistema operativo VDOS**

Los canales de entrada y salida están asignados al reproductor de datos. Si tiene un ordenador CPC464 y desconecta el Floppy, ello corresponderá exactamente a este estado.

Las operaciones de escritura/lectura se refieren siempre al reproductor. Si introduce aquí, por ejemplo, !CAS [ENTER] y CAT [ENTER], entonces aparece el aviso Press PLAY then any key (pulse [PLAY] y luego cualquier tecla). Por consiguiente, tenemos que pulsar la tecla [PLAY] en el reproductor. Luego recibiríamos un índice de contenido desde la cinta situada en el reproductor, en caso de que exista. Resulta evidente que esto requiere un cierto tiempo.

## **La orden !DISC.IN del sistema operativo VDOS**

El canal de entrada está asignado a la estación de disquetes. Es decir, si ahora se leen los datos, proceden del Floppy.

## **La orden !CAS.IN del sistema operativo VDOS**

El canal de entrada está asignado al reproductor. Es decir, si ahora se leen datos, proceden del reproductor.

## **La orden !DISC.OUT del sistema operativo VDOS**

El canal de salida está asignado al Floppy. Si ahora se realiza una operación de escritura, estos datos se escriben sobre el disquete situado en la unidad de disco.

## La orden **|CAS.OUT** del sistema operativo VDOS

El canal de salida está asignado al reproductor. Los datos se escriben ahora sobre la cinta situada en el reproductor.

Si desea, por ejemplo, copiar en la cinta un fichero o grupos de datos procedentes del disquete, puede efectuarse en la forma siguiente:

```
|CAS.OUT [ENTER]
|DISC.IN [ENTER]
LOAD "TEST" [ENTER]
SAVE "TEST" [ENTER]
```

La orden **LOAD** se referiría al disquete y la orden **SAVE** a la cinta. Después de pulsar la última vez **[ENTER]**, aparecería en la pantalla lo siguiente:

Press REC and PLAY then any key: (pulsar **[REC]** y **[PLAY]** y luego cualquier tecla), con la petición de ajustar el reproductor a funcionamiento de escritura.

## La orden **|XMON** del sistema operativo VDOS: el monitor **VORTEX "Z80"**

Detrás de esta orden se esconde un monitor Z80, muy potente y de gran rendimiento, en el lenguaje de la máquina.

Aquí la palabra "monitor" no tiene nada que ver con una pantalla o algo similar. Un monitor es un programa, con cuya ayuda se pueden comprobar otros programas de la máquina. Se trata de encontrar los errores, que puedan existir todavía en este programa de la máquina.

Para los errores se utiliza aquí también preferentemente la expresión inglesa "Bug" (parásito). Por ello, se ha popularizado también el nombre de "Debugger" (desparasitador) de un monitor para la eliminación de errores en un programa.

Además, se denomina también "Z80". Seguramente se ha dado cuenta por qué. Efectivamente, porque en su ordenador CPC está incorporado un microprocesador Z80.

Las funciones esenciales del monitor son:

- Monitor de Disquetes
- HEX/ASCII Dump (vaciado de memorias)
- Listado Mnemotécnico/desensamblamiento
- Breakpoints (puntos de ruptura)
- Tracer (rastreador/localizador)

Hay que decir claramente, que el monitor no es ningún juguete. Usted comprenderá para qué sirve y todo lo que puede hacer, cuando se haya acostumbrado un poco al lenguaje de la máquina del Z80. (Para ello, recomiendo el libro de Rodney Zaks "El Z80", publicado por la editorial Sybex Verlag.) Si no sabe absolutamente nada sobre él, será mejor que, de momento, no lo utilice con la orden **!XMON**.

A los que ya lo conocen, no necesito explicarles las insospechadas posibilidades que se les ofrecen con esta "herramienta".

El monitor se pone en marcha mediante la introducción de:

**!XMON** ENTER

Entonces se borra la pantalla y aparece el símbolo inductor (Prompt) "\*" del monitor. La orden **!XMON** espera ahora que le introduzcamos las órdenes.

La operación del monitor se termina pulsando la tecla ESC. Si, por ejemplo, ha solicitado la intervención del monitor, interrumpiendo un programa, y vuelve a salir del mismo otra vez a través de ESC, en dicho programa se continúa otra vez exactamente en el mismo lugar, en el que estaba al solicitar la actuación del monitor.

A continuación aparecen por orden alfabético todas las órdenes del monitor. Aquí debe tenerse en cuenta que los valores solicitados o indicados aparecen básicamente en forma hexadecimal, en la que los valores de 8 bits tienen o deben tener dos cifras y los valores de 16 bits tienen o deben tener 4 cifras.

**Dirección** `ENTER`

– Introducir la cadena de texto.

\***A3000** `ENTER` permite la introducción en la memoria de una cadena completa de textos, a partir de la dirección 3000H:

\***A3000**, esto es una cadena de prueba `ENTER`.

La cadena puede contener un máximo de 255 signos.

**@** – Poner en marcha el monitor de disquete.

El signo **@** se llama “arroba” y se obtiene pulsando la correspondiente tecla (situada entre la “P” y la “[”). La orden **!XMON** dispone de un pequeño monitor de disquete, cuya activación se solicita por medio de:

\***@** (No hay introducción `ENTER`)

que anuncia su presencia con

L:00 T:00 S:41 B:A93E

>

El “>” es el inductor del monitor de disquete. Cada una de las anotaciones en la línea del estado tiene el siguiente significado: (los valores en cifras aparecen en forma hexadecimal y se deben introducir también en forma hexadecimal.)

L:xx Número de la unidad de disco xx = 00 o bien 01 para la unidad de disco lógica A o bien B. Al entrar la señal, en xx se indica el número de la unidad seleccionada en este momento.

T:yy Número de pista. Gama de valores  
0 ... 27H (unidad de 3")  
0 ... 9FH (5.25"/3.5")

S:zz Número de sector. Gama de valores 41H ... 49H o bien C1H ... C9H o bien 01 ... 08 en la unidad de disco de 3"; 01 ... 09 en la unidad de disco de 5.25"/3.5". El formato de un disquete de 3" es reconocido automáticamente en la orden de mando R (véase abajo) y entonces se modifica en forma adecuada el número de sector.

**B:abcd** Dirección tampón de memoria intermedia para escribir/leer el sector. Al poner en marcha el monitor de disquetes, aparece aquí la dirección de la memoria intermedia BIOS (véase la parte interna).

A continuación se explican las diversas órdenes del monitor de disquetes:

- >**Lxx** **[ENTER]** Modificar el número de la unidad de disco.
- >**Sxx** **[ENTER]** Modificar el número del sector.
- >**Txx** **[ENTER]** Modificar el número de pista.
- >**Babcd** **[ENTER]** Poner la dirección de memoria intermedia.
- >**R** Introducir en la memoria intermedia, mediante lectura, el sector con las coordenadas definidas por medio de **L**, **S** y **T**.
- >**W** Escribir en el sector el contenido de la memoria intermedia con las coordenadas **L**, **S** y **E**.
- >**Cabcd** **[ENTER]** Modificar el contenido de las zonas de la memoria en la memoria intermedia. Corresponde concretamente a la orden **S** del monitor (véase también allí).
- >**Aabcd** **[ENTER]** Introducir la cadena (véase arriba).
- >**P** Conectar/desconectar la impresora. Corresponde a la orden **P** del monitor.
- >**Dabcd** **[ENTER]** / ,xx **[ENTER]** /  
Reproducir en la pantalla el contenido de la memoria intermedia en forma de HEX/ASCII Dump (volcado o copias de memorias). Opcionalmente se puede reproducir también la cantidad de sectores, que deban ser copiados de la memoria en total.
- >**[ESC]** Retorno al monitor.

Todos los datos introducidos se comprueban tanto sintáctica como lógicamente. Si comete un error de introducción, el monitor de disquete retrocede hasta el bucle de mando principal y avisa >>, a fin de indicarle a usted el error; naturalmente no se ha modificado nada.



Si se produce un error en el disquete al leer o al escribir, el monitor de disquetes avisa con:

ERR:xx

y salta otra vez hasta el bucle de mando. xx contiene el código de error. (Véase el párrafo “posibilidades de detección de errores en los disquetes”.)

**B** `[ENTER]` – Comprobar/modificar los puntos de ruptura.

El monitor proporciona la posibilidad de colocar opcionalmente hasta 8 puntos de ruptura (puntos de comprobación). (B corresponde a puntos de ruptura.) Si el programa a comprobar llega en uno de estos puntos, se produce la parada del programa, se emite el inductor “\*” y entonces se puede examinar y comprobar tranquilamente la grabación o registro, para ver si todo ha ido bien hasta ahora.

El ajuste de los puntos de interrupción se realiza por medio de la orden B:

\*B `[ENTER]`

-B1-	-B2-	-B3-	-B4-	-B5-	-B6-	-B7-	-B8-
0000	0000	0000	0000	0000	0000	0000	0000

Luego puede modificar el primer punto de ruptura, o bien llegar hasta el siguiente punto por medio de `[ENTER]`, etc., hasta que aparezca otra vez el inductor “\*” del monitor, después del octavo punto de ruptura o bien pulsando la tecla `[ESC]`.

**D** dirección 1, dirección 2 `[ENTER]` – Reproducir la lista en forma HEX/ASCII.

La orden **D** (D significa Dump = volcado de memoria) le proporciona la posibilidad de indicar los contenidos de la memoria en el siguiente formato:

```
0100 00 00 41 00 33 57 00 00 00 51 00 35 00 00 00 00
..A. !9...3.#....
```

Al comienzo de la línea se le indica en la memoria la dirección (0100H), a partir de la cual se indican en esta línea 16 contenidos de direcciones. Luego vienen los 16 valores (hexadecimales) y totalmen-

te a la derecha sus equivalencias en ASCII; si el valor es menor de 32 o mayor de 127, aparece un punto “.” y en los demás casos el signo correcto de ASCII.

Después de pulsar la tecla D, introduzca sólo una dirección (dirección 1), y entonces se reproducirán en la pantalla los siguientes 100H contenidos de la memoria; de todos modos, después de dirección 1 no pulse la tecla **ENTER**, sino la de la coma, de modo que puede luego introducir una segunda dirección (dirección 2). Luego se copian de la memoria (Dump) en forma HEX/ASCII los contenidos almacenados desde la dirección 1 hasta la dirección 2. Este listado lo puede mantener durante algún tiempo, pulsando una tecla cualquiera (preferentemente la tecla vacía) y luego puede volver a poner en marcha pulsando otra vez. Después de dicha parada, pulse la tecla **ESC** y aparecerá otra vez el inductor “\*” del monitor.

**F** dirección 1, dirección 2, valor      – Rellenar una zona de la memoria con un valor determinado.  
**ENTER**

Ocurre frecuentemente que hay que borrar determinadas zonas de la memoria. Esto no es ningún problema con la orden F (F significa Fill, rellenar). Sin embargo, hay que tener cuidado con lo que se rellena. Hay zonas en las que se desarrolla activamente un programa. Si se realiza aquí el relleno, puede ocurrir que se venga abajo todo el ordenador, así que ¡cuidado!

\***FC000,FFFF,FF** **ENTER**  
\*

rellena la memoria de la pantalla de imagen (direcciones C000 hasta FFFF) con el valor FF.

**G** dirección **ENTER**      – Realizar un programa de la máquina.

Esta orden (G significa Go, en marcha) realiza un programa en lenguaje de máquina, que se encuentra a partir de la dirección introducida. Si se alcanza entonces un punto de ruptura (véase anteriormente), el control vuelve al monitor y aparece el inductor “\*” preparado.

\***GBD19** **ENTER**

realiza el programa en lenguaje de máquina a partir de la dirección BD19.

**I** **dateiname.dateityp** **[ENTER]** – Cargar un fichero binario de disquete/cinta.  
(I nombre fichero, tipo fichero)

Con esta orden (I significa Input, introducir) se puede cargar en el monitor un fichero o grupo de datos binario (tipo de fichero BIN). Si ha introducido usted, por ejemplo, **!CAS.IN [ENTER]** antes de la puesta en marcha de la orden **!XMON**, se carga desde la cinta el fichero pedido por usted en la orden **!XMON**. Si no introduce nada, se carga desde el disquete.

El fichero o grupo de datos binario se carga en la memoria original, es decir en la zona de la memoria, en la que ha sido producido. Se indica en la pantalla la dirección de carga, la longitud, así como una dirección de comienzo eventualmente existente.

```
*ITEST.BIN [ENTER]
<2000 0380 2140>
*
```

carga el fichero TEST.BIN en la memoria. La dirección inicial obtenida es 2000H, la longitud 380H y la dirección de comienzo 2140H.

**L** dirección 1, dirección 2 – Desensamblar una zona de la memoria.

La introducción y el significado de las direcciones corresponden a los de la orden **D** (véase anteriormente), aunque el contenido de la memoria se reproduce en forma mnemotécnica.

```
*L0100,0106 [ENTER]
0100 3E64 LD A,64H
0102 ED5B3204 LD BC,(0432H)
0106 E3 EX (SP),HL
```

reproduce el listado de la zona desde 100 hasta 106. Al comienzo aparece la dirección, luego su contenido en forma hexadecimal y finalmente los datos mnemotécnicos legibles del Z80.

**M** dirección 1, dirección 2, dirección 3 **[ENTER]** – Desplazamiento de zonas de la memoria.

El contenido de la zona de la memoria entre la dirección 1 y la dirección 2 se copia en la zona de la memoria comenzando con la dirección 3. Aquí hay que tener un poco de cuidado de que la dirección 3 no quede dentro de la zona dirección 1/dirección 2 o tan cerca de la dirección 1, que se produzca un solapado o superposición.

**\*M0000,3FFF,C000**

copia la zona de 0-3FFF en la zona de la memoria a partir de la dirección C000 (memoria de imagen).

**Odateiname.dateityp**   
**adresse1,länge,adresse2**

– Almacenamiento de un fichero o grupo de datos.

(0 nombre fichero. Tipo fichero dirección 1, longitud, dirección 2)

Si ha introducido una denominación permitida de ficheros o grupos de datos después de la **O** (**O** significa Output, salida) se pregunta de acuerdo con la orden **SAVE** del sistema BASIC, el “nombre”, **B** después de la dirección inicial de la zona de la memoria, su longitud y una dirección de comienzo. Luego se escribe esta zona de la memoria en disquete/cassette con la denominación introducida.

**\*O TEST.BIN**  **C000,4000,0000**

\*

En el fichero binario TEST.BIN se almacena la zona de la memoria desde C000 hasta FFFF, sin dirección de comienzo.

**P** – Impresora conectada/desconectada.

Si hay una impresora acoplada al equipo, la reproducción aparecida en la pantalla se registra documentalmente en dicha impresora, pulsando una vez la tecla **P**. Al pulsar otra vez **P** se desconecta de nuevo la impresora (**P** significa Print, imprimir).

**R** – Indicar/modificar los contenidos de registro.

Los contenidos de registro para el desarrollo de un programa en el lenguaje de la máquina (con la orden **G** o con la orden **T**) se pueden modificar con esta orden **R** (**R** significa Register, registro). Estas órdenes se exponen en forma similar a los puntos de ruptura y se modifican exactamente del mismo modo. Sólo se puede modificar la primera secuencia de registro.

**S** dirección [ENTER] – Indicar y modificar el contenido de la memoria.

Después de introducir la **S** (**S** significa Substitute, sustituto) y la dirección, verá su contenido y podrá modificarlo adecuadamente en una tercera columna. Pulsando la tecla [ENTER] se pasa a la siguiente célula de almacenamiento y en ella se anota el valor eventualmente modificado de la dirección antigua. Pulsando la tecla [ESC], se vuelve otra vez al inductor del monitor “\*”.

\***S400** [ENTER]

**4000** \$ **24 34** [ENTER]

**4001** **C 43 32** [ENTER]

**4002** : **58 44** [ENTER]

**4003** [ESC]

\*

realiza la modificación a partir de la célula de almacenamiento 4000. Los valores antiguos de las direcciones 4000, 4001 y 4002 (24, 43 y 58) se sustituyen por los nuevos valores (34, 32 y 44).

**T** dirección, cantidad [ENTER] – Realizar paso a paso el programa de la máquina.

Con esta orden se puede realizar paso a paso un programa de la máquina, a fin de descubrir los errores que podría contener eventualmente. Se parte de los valores de registros introducidos con la orden **R** y el control se devuelve al monitor después de cada paso, es decir, usted ve el registro eventualmente modificado y puede interrumpir también el desarrollo del programa, si ha indicado una cantidad opcional de pasos. Esta interrupción la alcanza, como de costumbre, con la tecla [ESC]. Tenga en cuenta que los programas en la memoria ROM o las órdenes que trabajen con la secuencia gemela de registro del Z80 no se pueden desarrollar paso a paso. Además, el puntero de pila “Stackpointer” SP no debería estar en una memoria ROM, lo que puede producir el derrumbamiento del sistema.

\***T4000,0007** [ENTER]

realiza siete pasos, a partir de la dirección 4000H. Los contenidos de registro son indicados siempre y el desarrollo se puede detener (pulsando la tecla vacía) o interrumpir ([ESC]), igual que en la orden D.

- X** – Conectar/desconectar la ROM, microprogramación en memoria muerta (Firmware).

La orden X es también un conmutador, igual que la orden P (véase anteriormente). Después de poner en marcha el monitor por medio de la orden **XMON** **[ENTER]**, queda desconectado X, es decir, se intercala la memoria interna RAM en el espacio de dirección desde 0000H hasta 3FFFH. Si ahora se introduce la orden X (que no debe enviarse con **[ENTER]**, igual que ocurre con P), verá la microprogramación en memoria muerta (Firmware) ROM desde 0000H hasta 3FFFH.

- Y** – Seleccionar la memoria ROM superior.

Si pulsa una Y, el monitor le proporcionará una lista de todas las memorias ROM de ampliación en el sistema. Esta lista tiene la siguiente constitución:

```
xy   rsabvw   name
     .....
     .....
```

Aquí significan:

xy	=	Número de la memoria ROM de ampliación
rs	=	Número de marca de la memoria ROM
ab	=	Número de versión de la memoria ROM
vw	=	Número de modificación de la memoria ROM
name	=	Nombre RSX de la rutina ROM WALK.

Directamente después de la lista de la memoria ROM llega entonces:

**Yac** (ac = memoria ROM actual de ampliación).

y el monitor espera a que usted haga su introducción. Ahora puede modificar el número de memoria ROM de ampliación (terminar la introducción con **[ENTER]**). Los números permitidos de memoria ROM de ampliación están entre 00 y FBH.

ac es aquí siempre la memoria ROM que ve en la zona de dirección de C000H hasta FFFFH. Si introduce el número de una memoria ROM que no existe, el monitor conectará siempre la memoria ROM

del sistema BASIC incorporada en el equipo. Si desea ver el contenido de la memoria en la pantalla, introduzca un número de memoria ROM entre FCH y FFH.

**C** – Buscar series sucesivas de Bytes.

Con ayuda de la orden **C** puede buscar en la memoria determinados Bytes aislados o también series enteras de Bytes. La longitud máxima es aquí de 255 Bytes.

**Cabcd** [ENTER],**xy** [ENTER]

**abcd** dirección de la serie de Bytes

**xy** longitud de la serie de Bytes.

Previamente deberá señalar en la memoria con la orden **S** o bien **A** (véase anteriormente) la serie sucesiva de Bytes que desea.

Luego, la orden **C** le reproduce una lista de todas las direcciones, en las que ha encontrado la serie de Bytes.

## La orden **!ROMOFF** del sistema operativo VDOS

La orden **!AMSDOS** desconecta selectivamente sólo la memoria ROM del sistema VDOS, es decir, que luego no se dispone ya de ninguna de las órdenes RSX de la memoria ROM del sistema VDOS. En forma totalmente equivalente a la de la orden **!AMSDOS** trabaja, entre otras, también la orden **!DISBOS** de la tarjeta RAM de VORTEX, la cual desconecta selectivamente la memoria ROM del sistema BOS. Aquí se ve el conflicto originado, puesto que no es posible desconectar simultáneamente con estas dos órdenes las memorias ROM de los sistemas VDOS y BOS. Por este motivo, se ha incluido la orden **!ROMOFF**, que permite desconectar simultáneamente varias memorias ROM.

**!ROMOFF** [ENTER]

desconecta todas las memorias ROM de fondo.

**!ROMOFF,6,7** [ENTER]

desconecta sólo las memorias ROM de fondo con los números 6 y 7.

(Aquí pueden estar los números 0 . . . 7 de ROM. Las memorias ROM de fondo se pueden conectar otra vez, naturalmente, rearmando (RESET) el ordenador CPC (conexión/desconexión del mismo, o bien pulsar simultáneamente las teclas **CTRL**, **SHIFT** y **ESC**.)

## La orden **!ROMS** del sistema operativo VDOS

La orden **!ROMS** del sistema operativo VDOS proporciona una tabla de todas las memorias ROM de fondo, con números entre 0 y 7, en la forma siguiente:

N.º	Nombre	Versión
7	V ROM	0.31

etc.

Por lo tanto, si desea trabajar con la orden **!ROMOFF** del sistema operativo VDOS, primeramente puede conseguir con la orden **!ROMS** del sistema VDOS los números de la memoria ROM de fondo, que desea desconectar.

## Administración relativa y secuencial de ficheros o grupos de datos

Hasta ahora hemos hablado siempre sólo sobre ficheros o grupos de datos, cuyo contenido es un programa (ficheros de programa).

Suponga ahora que desea escribir en BASIC un pequeño programa de gestión o administración de direcciones, con el que pueda tramitar las direcciones (nombre, dirección postal, etc.) de sus amigos o conocidos, es decir, que pueda introducir las, modificarlas y buscarlas. Esto presupone que puede almacenar permanentemente estos "datos" (en disquete o en cinta). Por lo tanto, deberá construir su programa de modo que forme un fichero de direcciones, en el que estén todas. Este fichero sería entonces un grupo de datos o fichero propiamente dicho, pero su contenido no sería ningún programa realizable.



Si no tiene acoplado a su ordenador CPC ningún Floppy VORTEX, le falta a usted el sistema operativo VDOS.

Usted tiene ahora dos “canales” (canal de entrada y de salida; véanse también las órdenes |CAS y |DISC, párrafo “Reproductor de cintas en el ordenador CPC”), a través de los cuales puede enviar y recibir también datos al aparato reproductor o a la unidad de disco de 3” eventualmente disponible (las expresiones técnicas en inglés para estas operaciones son INPUT-Stream y OUTPUT-Stream).

Esto quiere decir que puede tener “abiertos” siempre simultáneamente sólo un fichero para escribir y un fichero para leer.

Antes de “acceder” a un fichero o grupo de datos, éste debe ser primeramente “abierto”. Si no necesita de momento este fichero, deberá ser cerrado otra vez.

Por lo tanto, no es posible tener abiertos simultáneamente dos ficheros para escribir o dos ficheros para leer. Esto es, desde luego, una cierta limitación, con la que hay que conformarse.

Un inconveniente bastante mayor es que el contenido de estos ficheros sólo puede ser leído o escrito sucesivamente. Este “sucesivamente” indica la forma en que estos ficheros son “gestionados” por el sistema operativo. Por eso, esta gestión o administración de los grupos de datos se denomina también “gestión secuencial de los ficheros”.

Si tiene por ejemplo 1000 direcciones en su fichero de direcciones y quiere leer la última, deberá leer previamente todas las 999 direcciones precedentes, lo que requiere mucho tiempo y proporciona poca eficacia al sistema.

La existencia de la gestión secuencial de ficheros está condicionada históricamente. El primer ordenador Amstrad (el CPC464) estaba equipado sólo con un aparato reproductor de cintas. Resulta claro que en este caso la gestión de ficheros sólo puede ser secuencial, puesto que el medio de almacenamiento es aquí una cinta. Si se desea ir desde X hasta Y, hay que dejar pasar toda la cinta existente entre ambas.

Luego llegaron las unidades de disco de 3”. Con ello se ha cambiado

prácticamente muy poco en el sistema operativo; se ha sustituido simplemente el aparato reproductor por una unidad de disco. Se continuaba dependiendo de la gestión secuencial de ficheros, a pesar de que con la unidad de disco se pueden hacer muchas más cosas.

A continuación encontrará un pequeño ejemplo de la gestión secuencial de ficheros.

Las órdenes BASIC típicas, que aparecen en relación con la lectura de un fichero secuencial son:

```
OPENIN EOF INPUT#9 LINE INPUT#9 CLOSEIN
```

Las órdenes BASIC típicas, que aparecen en relación con la escritura de un fichero secuencial son:

```
OPENOUT PRINT#9 WRITE#9 CLOSEOUT
```

```
10 OPENOUT (cifras.datos)  
20 FOR A=1 TO 100  
30 A$=STR$(100)  
40 PRINT#9,A$  
50 NEXT  
60 CLOSEOUT  
70 OPENIN (cifras.datos)  
80 INPUT "¿Qué cifra busca usted?", cifra  
90 ZAHL$=STR$(cifra)  
100 INPUT#9,VAR$  
110 IF ZAHL$=VAR$ THEN GOTO 140  
120 Z=Z+1  
130 GOTO 100  
140 PRINT "La cifra está siempre en la posición n.º 7 en el fichero"  
150 CLOSEIN  
160 END
```

Con la línea 10 se abre el fichero CIFRAS.DATOS para escribir; si no existía todavía, se registra ahora. Dentro del bucle FOR/NEXT se escriben en este fichero las cifras 1 a 100 (transformadas en cadenas) con **PRINT#9,A\$**. La línea 60 cierra adecuadamente el fichero o grupo de datos.

En la línea 70 se abre otra vez el fichero para lectura. Buscamos una determinada cifra o bien su situación en el fichero CIFRAS.DATOS. La línea 110 muestra una comparación típica para ficheros secuenciales: hay que ir comparando a través del fichero, hasta que se ha encontrado la secuencia de datos que se busca.

Así pues, ahora acoplamos la unidad de disco X VORTEX y disponemos de este modo de un sistema operativo VDOS.

Este sistema domina naturalmente también la gestión secuencial de ficheros descrita anteriormente, pero puede hacer todavía mucho más. Le facilita una “gestión relativa de ficheros”.

Lo característico de una gestión relativa de ficheros es que saca de cualquier parte del fichero una secuencia de datos, o bien la puede intercalar escribiéndola en el mismo, sin tener que haber leído previamente todas las secuencias de datos.

Aquí se puede acceder opcionalmente a cada secuencia de datos por cada número de secuencias de datos, lo que naturalmente ahorra tiempo y trabajo de programación.

Además, el sistema VDOS le permite abrir simultáneamente hasta 16 canales relativos, además de los dos canales secuenciales. Con ello se puede transformar óptimamente en un programa cualquier problema planteado.

Los ficheros relativos, igual que los ficheros secuenciales, constan de secuencias de datos (denominadas también records o registros). A diferencia de un fichero secuencial, en un fichero relativo estos records o registros son totalmente independientes entre sí. Cada registro tiene un número, a través del cual puede ser seleccionado. Se pueden leer o escribir por cualquier orden consecutivo que se desee. La longitud de una secuencia de datos puede ser elegida libremente dentro de unos límites determinados.

Para cada fichero relativo abierto, se reserva una zona determinada en la memoria RAM, que se denomina “memoria intermedia de registro”. Si se lee o escribe una secuencia de datos, ésta llega siempre a través de esta memoria intermedia desde el 0 hasta el fichero. Es claro que el tamaño de esta memoria intermedia depende de la longitud de registro elegida en principio.

Debe tenerse en cuenta que, en un fichero relativo abierto, no se puede modificar la longitud de secuencias de datos.

Otra diferencia entre las dos “clases de gestión” consiste en que, en un fichero relativo, las secuencias de datos (registros) pueden ser todavía estructuradas; se dice también que el registro puede ser todavía subdividido en diferentes “campos”.

También se puede decir, por lo tanto, que el fichero relativo consta de una determinada cantidad de secuencias de datos, en la que todas ellas son todavía subdivisibles en una determinada cantidad de campos. La división de los campos en un fichero relativo debe ser siempre la misma en todas las secuencias de datos.

Todo fichero relativo abierto necesita un almacén de datos en la memoria RAM para su memoria intermedia de registro, para su FCB (File Control Block, que contiene también la denominación del fichero relativo) y varios indicadores y contadores internos (todo en conjunto son 90 Bytes + memoria intermedia de registro por cada canal).

Este almacén de datos se reserva al hacer la definición del canal (véase la orden **!FILES**) en la zona superior de la memoria de almacenamiento del ordenador CPC mediante una rebaja dinámica del **HIMEM**. Cuando se ha definido ya un canal, se puede abrir sobre el mismo un fichero relativo (véase la orden **!OPEN**). Debe tenerse en cuenta que, tanto en la definición del canal (**!FILES**), como también al abrir un fichero en este canal (**!OPEN**), se deben indicar las longitudes de registro.

Esto tiene el siguiente motivo, fácilmente comprensible: desde luego, es posible abrir y cerrar en un canal varios ficheros relativos sucesivamente, sin tener que hacer cada vez la definición del canal (**!FILES**). Únicamente se indica la longitud máxima de registro prevista al hacer la primera definición del canal y luego se abren los diferentes ficheros (**!OPEN**) con sus respectivas longitudes individuales de registro.

Una vez abierto un fichero relativo (**!OPEN**), se debe estructurar para el fichero la memoria intermedia de registro, antes del primer acceso al fichero (**!PUT**, **!GET**), o sea, hay que subdividir en campos (**!FIELD**), es decir, indicar qué partes del registro se deben introducir por lectura en variables al hacer el acceso.

Ahora está preparado ya el fichero relativo para la transmisión de datos y se pueden leer y escribir registros opcionalmente hasta cerrar el fichero (**!CLOSE**). (Se dice frecuentemente también “casualmente”, por lo que se emplea a veces la expresión inglesa “Random File”.)

Hay que tener en cuenta que en los registros únicamente se pueden escribir y volver a leer variables de cadena. Si se desea almacenar variables numéricas o campos (también campos de cadenas) en ficheros relativos, éstos hay que transformarlos previamente en variables de cadenas, o bien asignarlos a una variable de cadena y almacenar luego ésta en la memoria (véase la orden **STR\$** del sistema BASIC en el manual de BASIC de su ordenador CPC).

Si desea “recuperar” un valor numérico, hay que transformarlo después de extraído de la lectura del fichero, con la orden **VAL** de BASIC.

Un registro puede ser dividido de nuevo en segmentos varias veces (**!FIELD**), después de abierto el fichero, sin que haya que cerrar y volver a abrir de nuevo el fichero cada vez.

Pero, ahora trataremos de las órdenes del sistema operativo VDOS, que permiten trabajar con ficheros o grupos de datos relativos.

### **La orden !FILES del sistema operativo VDOS: reservar memorias de trabajo para uno o varios ficheros relativos**

Antes de poder trabajar con un fichero o grupo de datos relativo, hay que definir para ello un canal y una memoria intermedia de registro.

La apertura de un fichero relativo, sin haber reservado previamente una memoria intermedia para este número de canal con la orden **!FILES**, proporciona el siguiente aviso de error:

campo no definido

Por este motivo, la orden **!FILES** debería estar (aunque no es imprescindible) al comienzo de un programa BASIC y debería reservarse una memoria intermedia de registro suficiente para todos los números de canal utilizados posteriormente.

La orden **!FILES** trabaja con números dobles, es decir, que para cada número de canal indicado hay que indicar también simultáneamente una longitud máxima de registro. El número de canal puede tener valores desde 0 hasta 127 (pueden estar abiertos simultáneamente un total de 16 canales); la longitud máxima de registro puede elegirse opcionalmente y está limitada sólo por las memorias de trabajo disponibles.

Cada nueva orden **!FILES** anula la orden dada anteriormente. Si sólo se introduce la orden **!FILES** **[ENTER]**, se borran en la memoria RAM todas las zonas de trabajo de los ficheros relativos.

Mediante la introducción de:

```
!FILES,1,128,2,64,12,20 [ENTER]
```

se reservan para el canal 1 una memoria intermedia de registro de 128 Bytes, para el canal 2 una memoria intermedia de registro de 64 Bytes y para el canal 12 una memoria intermedia de registro de 20 Bytes. Ahora se pueden abrir ficheros relativos en estos 3 canales.

## **La orden !OPEN del sistema operativo VDOS: abrir un fichero o grupo de datos relativo**

Con la orden **!FILES** hemos creado la premisa indispensable para poder utilizar un fichero o grupo de datos relativo.

El paso siguiente es “abrir” el fichero relativo, para poder escribir en el mismo los datos o sacar del mismo la lectura de datos. Para ello sirve la orden **!OPEN**.

```
!FILES,3,300 [ENTER]
```

```
!OPEN,"DATEI.REL",145,3 [ENTER] (DATEI = fichero).
```

Examinemos ahora con más detenimiento la orden **!OPEN**. Evidentemente, aquí se abre un fichero relativo denominado DATEI.REL (fichero.rel). Si ya existe este fichero en el disquete, se abre simplemente; si todavía no existe, hay que colocarlo primeramente y luego abrirlo.

“145” es la longitud actual de registro, que debe ser menor o igual que la longitud máxima de registro (ajustada a 300 en la orden **|FILES**.

El fichero se abre en el canal 3, en el que en este momento no debe estar abierto ningún otro fichero.

Repetimos de nuevo: en el mismo canal sólo debe estar abierto simultáneamente un fichero. Pero, si necesita más ficheros, deberá elegir simplemente otro número de canal.

Si abre un fichero, que ya esté abierto, recibirá el siguiente aviso:  
canal ya abierto.

Olvide ahora la orden **|OPEN** y utilice luego, a pesar de todo, la orden **|PUT** o bien **|GET** y entonces obtendrá el siguiente aviso:  
canal no definido.

## **La orden |FIELD del sistema operativo VDOS: estructurar la memoria intermedia de registro**

Después de abierto un fichero relativo, se debe dividir en segmentos su memoria intermedia de registro, es decir, hay que indicar cuántas variables y con qué longitud deben existir en un registro del fichero.

Se desean tener en una secuencia de datos las 3 variables siguientes:

**NAME\$** (nombre) con una longitud máxima de 12 Bytes, **ORT\$** (lugar) con una longitud máxima de 20 Bytes y **TEL\$** (teléfono) con una longitud máxima de 15 Bytes. La orden **|FIELD** para esta combinación es, por ejemplo, como sigue:

```
|FIELD,4,12,20,15 ENTER
```

la longitud actual de la memoria intermedia en la orden **|OPEN** debería ser entonces  $12 + 20 + 15 = 47$  o mayor.

Esta división en segmentos se refiere a la memoria intermedia de registro del canal 4.

En relación con la orden **|FIELD** podemos recibir los siguientes avisos:

Campo demasiado largo:

hemos definido una longitud de campo, que sobrepasa la longitud de la memoria intermedia de registro, fijada con la orden **|FILES**.

Campo no definido:

en una orden **|PUT** o bien **|GET** hemos indicado una variable, para la que no se ha previsto ningún sitio en la orden **|FIELD**.

Campo no segmentado:

con las órdenes **|PUT** o bien **|GET** hemos querido escribir o leer una secuencia de datos, a pesar de que no la hemos subdividido (segmentado) todavía en campos individuales con la orden **|FIELD**.

## **La orden |GET del sistema operativo VDOS: leer una secuencia de datos**

Con la orden **|GET** se lee la secuencia de datos con el número indicado, procedente del fichero en este canal, en las variables indicadas, que han debido ser definidas previamente. Si no se han definido antes las variables, recibiremos el aviso de error:

improper argument (argumento inadecuado).

```
10 NAME$="" : ORT$="" : TEL$="" (nombre, lugar, teléfono)
20 |FILES,1,100
30 |OPEN,"ADRESS.REL",47,1 (dirección)
40 |FIELD,1,12,20,15
60 |GET,1,174,@NAME$,@ORT$,@TEL$ (nombre, lugar,
teléfono)
70 PRINT NAME$,ORT$,TEL$ (nombre, lugar, teléfono)
80 |CLOSE,1
```

Aquí se definen primero las variables (línea 10) y luego se introduce por lectura la secuencia de datos con el número 174, procedente del fichero en el canal A, en las variables **NAME\$**, **ORT\$** y **TAL\$**.



Tenga en cuenta, que aquí sólo está permitido utilizar el signo “@” (arroba). Previamente han de ser puestas en marcha correctamente las órdenes **!FILES**, **!OPEN** y **!FIELD**.

Si el registro leído no contiene datos, recibiremos el aviso:  
el registro no contiene datos.

### **La orden !PUT del sistema operativo VDOS: escribir una secuencia de datos**

La orden **!PUT** trabaja igual que la orden **!GET**, pero escribe el contenido de las variables indicadas en el fichero en el canal elegido.

```
10 NAME$="":ORT$="":TEL$="" (nombre, lugar, teléfono)
20 !FILES,1,100
30 !OPEN,"ADRESS.REL",47,1 (dirección)
40 !FIELD,1,12,20,15
50 INPUT NAME$.ORT$.TEL$ (nombre, lugar teléfono)
60 !PUT,1,174,@NAME$,@ORT$,@TEL$ (nombre, lugar,
teléfono)
70 !CLOSE,1
```

La orden **!PUT** se puede utilizar también sin la “@” (arroba):  
**!PUT,1,15,"Müeller","Koeln","123456"** (Müeller, Colonia).

Aquí se quitan directamente de la escritura las tres palabras de la secuencia de datos con el número 15 en el fichero en el canal 1.

Naturalmente, se entiende que el número de secuencia puede ser sustituido también por una variable, lo que es aplicable también para la orden **!GET**. Así, por ejemplo:

```
10 NAME$="":ORT$="":TEL$="" (nombre, lugar, teléfono)
20 !FILES,1,100
30 !OPEN,"ADRESS.REL",47,1 (dirección)
40 !FIELD,1,12,20,15
45 INPUT "Marque el número de secuencia de datos", REC
50 INPUT NAME$.ORT$.TEL$ (nombre, lugar teléfono)
60 !PUT,1,174,@NAME$,@ORT$,@TEL$
70 !CLOSE,1
```

Mediante la elección de REC se puede determinar el número de secuencia de datos, en la que se escribirá el contenido de las tres variables. Si marca un número de registro, que supera el tamaño del fichero, recibirá el aviso:

número de registro demasiado grande.

### **La orden |CLOSE del sistema operativo VDOS: cerrar un fichero relativo**

Si se debe desarrollar adecuadamente la utilización de un fichero o grupo de datos relativo, al final hay que cerrar otra vez el fichero relativo.

```
|CLOSE,1 [ENTER]
```

cierra el fichero, que estaba abierto en el canal 1.

Si en este canal no hay ningún fichero, recibirá el aviso:

canal no abierto.

### **Dos pequeñas aplicaciones**

A continuación indicamos los listados de dos pequeños programas, que le permitirán ver claramente otra vez la forma de trabajar con ficheros o grupos de datos relativos. En el primer ejemplo se abre el fichero TEST.REL y se lee e indica en la pantalla un registro seleccionable, o bien se inscribe un registro con un contenido introducible.

```
10 |FILES,1,600
20 |OPEN,"TEST.REL",520,1
30 |FIELD,1,250,250
40 X$="":Y$=""
50 PRINT "(L) leer o bien (S) escribir un registro o (E)
final"
60 K$=UPPER$(INKEY$):IF K$ <>"L" and K$<>"S"
THEN GOTO 60
70 INPUT (número de registro), REC
80 IF K$<>"L" THEN GOTO 10
```

```

90 |GET,1,REC,@X$,®Y$
100 PRINT X$,Y$:GOTO 50
110 IF K$<>"S" THEN GOTO 160
120 INPUT "2.ª variable",X$
130 INPUT "2.ª variable".Y$
140 |PUT,1,REC,@X$,@Y$
150 GOTO 50
160 |CLOSE,1:END

```

- 10: Preparación de una memoria intermedia de 600 Bytes de longitud para el canal con el número 1.
- 20: Abrir el fichero o grupo de datos TEST.REL con la longitud actual de secuencia de 520 Bytes en el canal 1.
- 30: Se escribe sólo el contenido de 2 variables en la memoria intermedia. Los contenidos de ambas variables deben tener una longitud máxima de 250 Bytes.
- 40: Definir las 2 variables para la orden |GET
- 50: Preguntar lo que debe hacerse.
- 60: Bucle de introducción.
- 70: Introducir el número de secuencia de datos.
- 90: Aquí llega usted, si ha introducido antes "L" (lectura). El contenido de la secuencia seleccionada con REC se introduce en las variables X\$ e Y\$.
- 100: Reproducción del contenido.
- 120: Aquí llega usted, si ha introducido arriba "S" (escribir). Hay que dar un valor a las variables X\$ e Y\$ (línea 130).
- 140: El contenido de las dos variables se escribe en la secuencia de datos seleccionada mediante REC.
- 150: Retroceso a la línea 50 para una nueva introducción.
- 160: Aquí llega usted, si ha introducido "E" (fin). El fichero queda cerrado correctamente.

Deberá marcar por lo menos un par de veces la orden de selección "S" (escribir), a fin de que el fichero TEST.REL reciba un contenido, que pueda luego ser leído con "L" (lectura).

En el segundo ejemplo se escriben sucesivamente 1000 registros en el fichero relativo ZUFALL.REL (casual.relatoivo). A continuación se accede "por pura casualidad" en el sentido más verdadero de la pala-

bra, a las secuencias de datos del fichero, a través de la función casual (RND = random).

```
10 |FILES,10,120
20 |OPEN,"ZUFALL.REL",25,10           (casual.relATIVO)
30 |FIELD,10,25
40 X$=""
50 FOR REC=1 to 1000
60 X$="Rekord Nr." +STR$(REC)       (Registro n.º)
70 INPUT,10,REC,@X$
80 NEXT REC
90 REC1=INT (1000*RND(1))+1
100 |GET,10.REC1,@X$
110 PRINT X$
120 GOTO 90
```

- 10: reservar una memoria intermedia de 120 Bytes para el canal 10.
- 20: Abrir el fichero relativo ZUFALL.REL (casual.relATIVO) en el canal 10 con la longitud actual de registro de 25.
- 30: Segmentación de la memoria intermedia. Se coloca sólo una variable.
- 40: Definición de la variable X\$ para la orden !GET.
- 60: Se escriben en el fichero ZUFALL.TXT (casual.TXT) 1000 registros: "registro n.º 1", "registro n.º 2" ... "registro n.º 1000".
- 90: Ponga ahora una cifra casual entre 1 y 1000.
- 100: Lectura de la secuencia de datos con el número de la cifra casual.
- 110: Indicación de la secuencia de datos.
- 120: Poner de nuevo una cifra casual y leer la secuencia de datos.

Con este ejemplo se demuestra muy bien la alta velocidad de acceso a los datos del fichero relativo.

El programa siguiente cumple los mismos objetivos que el ejemplo anterior, con la diferencia de que aquí se utilizan ficheros secuenciales. Haga primeramente la introducción. Se quedará asombrado de que todo se desarrolla mucho más lento. Esto le indica una vez más, que para trabajos profesionales es indispensable la "herramienta" de una gestión de datos relativa.

```

10 OPENOUT "ZUFALL.SEQ" (casual.secuencial)
20 FOR REC=1 TO 1000
30 X$="Rekord Nr." +STR$(REC) (registro n.º)
40 PRINT #9,X$
50 NEXT
60 CLOSEOUT
70 OPENIN "ZUFALL.SEQ." (casual.secuencial)
80 REC1=INT(1000*RND(1))+1
90VGL$="Rekord Nr." +STR$(REC1) (registro N.º)
95 FOR A=1 TO 1000
100 INPUT#9,X$
110 IF X$=VGL$ THEN GOTO 120
115 NEXT
120 PRINT X$
125 CLOSEIN
130 GOTO 70

```

En las líneas 10 a 60 se coloca el fichero ZUFALL.SEQ (casual.secuencial). Verá que los datos han sido desplazados sin señalarlos con números.

En la línea 110 se compara la cadena obtenida por el generador casual en la línea 90 con la cadena que se acaba de leer. Si son idénticas ambas, se indica la cadena en la pantalla (línea 120). Si no son iguales, hay que seguir comparando, hasta que quizás se coincida en la secuencia número 1000 (en el caso más desfavorable). Si se consigue que coincida la primera comparación (registro n.º 1), tendremos mucha suerte y el caso más favorable.

En la gestión relativa de ficheros no existen casos favorables ni casos desfavorables: todos son iguales.

Como puede comprobar fácilmente, la misma tarea es solucionada por la gestión relativa de ficheros del sistema operativo VDOS unas 10 veces más rápido que con la gestión secuencial de ficheros.

## Posibilidades de detección de errores en los discos flexibles.

### La orden ¡DERROR del sistema operativo VDOS

Hay que evitar que un programa BASIC resulte interrumpido por un aviso de la unidad de disco (en general avisos de errores) y que el BASIC avise con "Ready" (preparado). Para ello se pueden detectar los avisos de errores con la orden ¡DERROR, procesándolos en el programa. Es decir, se puede impedir totalmente un "abandono". Esto es muy importante, sobre todo para las aplicaciones PROFESIONALES: un abandono significaría aquí el fin del programa.

Después de la introducción de la orden ¡DERROR, 0 [ENTER] no aparecen ya en la pantalla los errores del disco flexible. Con la orden ¡DERROR, 1 [ENTER], se conecta otra vez la reproducción del error. Esto corresponde también al estado de conexión del ordenador CPC con el Floppy VORTEX acoplado. Si están desconectados los avisos de errores, se puede colocar la variable DERR del código de errores, mediante la introducción de la orden ¡DERROR [ENTER] (esto es, sin parámetros). A continuación de esta explicación de la orden, encontrará una lista de todos los códigos de errores.

```
10 ¡DERROR,0
20 CAT
30 ¡DERROR
40 IF DERR=72 then input "coloque el disquete y pulse
   cualquier tecla", A:GOTO 20
50 ....
```

El código de errores 72 significa que hay un disquete en la unidad de disco. En la línea 10 hemos desconectado la reproducción de errores. En la línea 30 hacemos que se coloque la variable del código de errores. Si la variable tiene el valor 72, aparece el aviso; en todos los demás casos continúa simplemente el programa.

ATENCIÓN: Determinados errores no se pueden detectar sólo mediante la orden ¡DERROR,0, sino que se debe trabajar adicionalmente con la orden:

**ON ERROR GOTO**

Suponemos que, para lo siguiente, no hay ningún disquete colocado en la unidad de disco. Por lo tanto, un acceso a la unidad produciría, sin detección de error, el aviso:

unidad d: falta disquete.

Si detectamos este error con la orden **!DERROR,0**, el sistema BASIC nos indica, a pesar de todo, un error. Deje que se desarrolle el siguiente programa:

```
10 !DERROR,0
20 LOAD,"TEST"
30 ..
```

Entonces recibirá el siguiente aviso de error:

File already open in 20

y se interrumpirá el programa.

Pruebe ahora el programa siguiente:

```
10 ON ERROR GOTO 100
20 !DERROR,0
30 LOAD "TEST"
40 ..
100 !DERROR:PRINT DERR:RESUME 40
```

Ahora no aparece ningún aviso de error y la rutina de detección de errores en la línea 100 reproduce un "72" en la pantalla. (72=&48), es decir, Bit 6 y Bit 3 están colocados y esto, a su vez, significa que se trata de un error FDC (Floppy Disc Controller Fehler = error del controlador de discos flexibles Floppy): "Drive not ready" (accionamiento no preparado), es decir que no hay ningún disquete en la unidad de disco.)

## **Códigos de error y su significado**

A continuación exponemos una lista de todos los avisos de error del sistema operativo VDOS: código de error y aviso en la pantalla. Además, se indican también al mismo tiempo las posibles causas.

“⟨...⟩” es un signo para guardar espacio para la denominación de un fichero o grupo de datos. “FDC” significa Floppy Disk Controller, Controlador de Discos Flexibles. Con “d” nos referiremos a una unidad de disco y concretamente a la denominación de la misma que aparece sobre la del error (por ejemplo, “A:” o bien “B:”). Todos los códigos de error deben indicarse en forma decimal.

**Código    Texto de aviso en la pantalla  
de error    y su significado**

- 17            Impresora no preparada  
Este aviso de error sólo puede aparecer en el sistema operativo CP/M y aun así, únicamente cuando la impresora no esté todavía preparada al cabo de unos 40 seg. después del intento de grabar en la misma un signo. Los orígenes del error pueden ser:
- cable de la impresora no enchufado
  - la impresora no está conectada
  - la impresora no está seleccionada.
- 10            Unidad de disco “d”: se ha cambiado el disquete y se cierra ⟨...⟩  
Se ha abierto un fichero con **OPENOUT** y luego se ha cambiado el disquete, sin marcar **CLOSEOUT**.
- 21            Unidad de disco “d”: disquete lleno  
Está lleno el disquete en la unidad “d”.
- 22            Unidad de disco “d”: directorio (índice) lleno  
El índice de contenido del disquete está lleno. Tenga en cuenta que este aviso puede aparecer antes de que vea realmente 128 registros con, por ejemplo, **CAT** **[ENTER]**. Motivo: por cada 64K de datos por programa se necesita un registro de índice o directorio. Un fichero que tenga, por ejemplo, 129K, ocupa ya tres registros en el índice de contenido.
- 23            ⟨...⟩ no se ha encontrado  
Este aviso aparece siempre que se intenta acceder a un fichero, que no exista en el índice. El error puede aparecer con las órdenes siguientes: **LOAD**, **SAVE**, **RUN**, **OPENIN**, **OPENOUT**, **!ERA**, **!REN** y **!ATTRIBUT**.



- 24 <...> ya existe  
Se intenta dar una nueva denominación a un fichero o grupo de datos, pero ya existe otro fichero con dicha denominación.
- 25 <...> sólo puede ser leído  
Se intenta borrar un fichero (con la orden **!ERA**), o bien escribirlo, que lleva el atributo “sólo legible” (colocado con la orden **!ATTRIBUT**).
- 33 Introducción errónea  
Se ha hecho una falta de escritura (error de sintaxis), o no se ha mantenido el formato de una determinada orden (error de semántica).
- 39 Campo demasiado largo  
La longitud actual del campo es mayor que la indicada en la orden **!FILES**. Sólo puede ser “más pequeña o igual”. Este aviso puede aparecer cuando:
- es demasiado grande la longitud actual en **!OPEN**
  - la suma de las longitudes individuales de campo en **!FIELD** es mayor que la definida en **!OPEN**.
- 40 Campo no definido  
Se intenta abrir un fichero o grupo de datos en un canal (con **!OPEN**), sin haber introducido previamente la orden **!FILES**.
- 41 Campo no segmentado  
Se intenta trabajar con las órdenes **!GET** o bien **!PUT**, sin haber segmentado previamente la memoria intermedia de registro con la orden **!FIELD**.
- 42 Canal ya abierto  
Se ha intentado abrir otra vez un canal que ya estaba abierto con **!OPEN**.
- 43 Canal no abierto  
Se ha intentado realizar una de estas órdenes (**!FIELD**, **!CLOSE**, **!GET** o bien **!PUT**) con un fichero en un canal no abierto.

- 44      Número de canal incorrecto  
Se ha marcado un número de canal no permitido junto con las órdenes **!CLOSE**, **!FIELD**, **!OPEN**, **!GET**, **!PUT** o bien **!FILES**. Los números de canal permitidos tienen que estar entre 0 y 127, ambos inclusive.
- 45      Número de registro demasiado grande  
El sistema operativo VDOS puede gestionar como máximo 65.536 registros de 128 Bytes. Dependiendo de la longitud actual de memoria intermedia de registro definida en **!OPEN**, varía la cantidad máxima de registros. Por ejemplo, con un tamaño de memoria intermedia de 512 Bytes, se pueden gestionar como máximo 16.384 registros.
- 53      Canal no definido  
Se ha intentado abrir un canal no definido previamente.
- 55      El registro no contiene datos  
Se ha intentado leer un registro, que ya no figura en la tabla de ocupación de bloques. Atención: este error no puede ser utilizado imprescindiblemente para comprobar el final de un fichero o grupo de datos relativo. Motivo: si la longitud actual de registro es menor que la longitud del bloque (4K en los Floppys VORTEX; 1K en la unidad de disco de 3", caben varios registros en un bloque. Entonces, por ejemplo, si sólo está ocupado el primer registro en el último bloque incluido en la tabla de ocupación de bloques, sólo aparecerá el error 55 cuando se intente acceder a un registro en el bloque siguiente. Con longitudes pequeñas de campos, su número de registro se puede diferenciar, en ocasiones fuertemente, del número del último registro ocupado realmente. Por ello, debería trabajarse siempre con ficheros de índice, es decir, que debería llevarse, además del fichero de datos, un segundo fichero de índice, en el que se gestionen los números de registros de datos y otras informaciones.
- 14      No hay ningún texto de aviso; el fichero no está abierto  
Se ha intentado cerrar un fichero secuencial con **CLOSEIN** o bien **CLOSEOUT**, que no estaba abierto.

- 15 No aparece ningún texto de aviso.  
Se ha alcanzado el final físico (efectivo) de un fichero o grupo de datos secuencial.

Todos los avisos anteriormente indicados son generados por el sistema operativo VDOS.

Los códigos de errores, que proceden directamente del controlador de discos flexibles FDC (tipo uPd 775 de NEC) han puesto el Bit 6, han borrado el Bit 7 y, por lo tanto, están entre 64 y 127. El significado de cada uno de los Bits es el siguiente:

**Bit Significado (en caso de que se haya puesto el Bit)**

- 7 Siempre 0.
- 6 Siempre 1.
- 5 Error CRC en el campo de datos o del ID.
- 4 El controlador FDC no ha sido manejado dentro de un determinado espacio de tiempo en un proceso de transmisión de datos con el ordenador.
- 3 La unidad de disco da el aviso de no preparado "not ready". Esto significa generalmente que no hay colocado ningún disquete en el hueco correspondiente de la unidad o que no ha sido correctamente bloqueado el botón de expulsión.
- 2 El controlador FDC no puede encontrar el sector deseado durante la realización de una operación de lectura.
- 1 El controlador FDC recibe una señal de protección de la escritura "Write protect" de la unidad de disco, durante una operación de escritura o de formateado. Esto significa que el disquete tiene una protección contra escritura.
- 0 El controlador FDC no puede encontrar el campo ID después de "sobrevolar" por segunda vez el agujero del índice.

Un código frecuente de error podría ser el 72, o bien binariamente 01001000 y significa que no hay disquete en la unidad de disco.

Si no se han desconectado con la orden **!DERROR** los avisos de error del disco, se produce su aparición en la pantalla. Para indicarle esto al programador, se coloca, por lo tanto, en este caso, el 7.º Bit en el Byte de error, es decir, se suman 128 al código de error anterior.

Por ejemplo:

<b>Código de error</b>	<b>Significado</b>	<b>Aviso de error</b>
21	Disquete lleno	Desconectado
149	Disquete lleno	Conectado

**Muy importante:**

Los errores del Floppy, que no producen ningún error del sistema BASIC en las operaciones sucesivas, no se pueden detectar tampoco con "ON ERROR GOTO" sino que debe preguntarse en cada caso por la orden de búsqueda de errores:

```
10 ON ERROR GOTO 100
20 !DERROR,0:'suprimir avisos
30 CAT
40 !DERROR:'colocar la variable de errores DERR
50 IF DERR= .....
.....
100 'tratamiento del error
```

Si en la línea 30 no hay ningún disquete en la unidad de disco, la indicación **CAT** cae en el "vacío", pero no produce ningún aviso de error en el sistema BASIC, es decir, que no se desvía después de la línea 100. El error se detecta en las líneas 40 y 50.

**Muy importante:**

La orden **!DERROR** sin parámetros se suprime en los ordenadores CPC 664 y CPC 6128, ya que el sistema BASIC 1.1 coloca de por sí la variable **DERR**, en caso de un error en el disquete, y no es necesario ponerla uno mismo como en el sistema BASIC 1.0 (CPC 464).

Las órdenes **!DERROR,0** y **!DERROR,1** pueden seguir siendo utilizadas, naturalmente, para conectar y desconectar los avisos de error de los disquetes.

Téngase en cuenta también el apéndice 1 "El problema EOF", al final de este manual.

## La función “giro de llave y en marcha” (turnkey) del sistema operativo VDOS, para el equipo CPC 464 + DDI-1 y unidad X de disco flexible

“Turnkey” significa en inglés simplemente girar la llave, lo que quiere decir “meter la llave, girar y ya está”. En el sistema de ordenador que posee, esto significa: “meter el disquete, conectar el aparato y en marcha”.

con la función “turnkey”, se puede llevar efectivamente la función del sistema hasta un estado determinado de funcionamiento, desde la situación de conexión. Véase el ejemplo siguiente. Suponga que desea utilizar el ordenador CPC sólo para mantener su correspondencia epistolar con el WordStar. Es decir, que a usted le resulta en realidad muy molesto tener que cargar el sistema CP/M antes de cada utilización, poner en marcha luego el WordStar e introducir eventualmente todavía una carta antigua. Preferiría sin duda tener que colocar únicamente el disquete, conectar el aparato y “depositar” inmediatamente su carta en el WordStar.

Esta tarea no supone ningún problema para el sistema operativo VDOS. Para ello hemos montado la función “turnkey” es decir, de “giro de llave y en marcha”: cada vez que se pone a cero el ordenador (rearme, o bien pulse simultáneamente las teclas `CTRL`, `SHIFT` y `ESC`), teniendo en cuenta que también la desconexión/conexión es un rearme o reposición, el sistema operativo VDOS busca un fichero o grupo de datos con la denominación “HELLO.BAS” en un disquete colocado eventualmente en la unidad de disco de 3”. Si el sistema VDOS encuentra este fichero, se pone en marcha automáticamente (equivale a `RUN “HELLO” ENTER`).

El fichero Hello tiene, por ejemplo, la siguiente configuración:

```
10 |CPM
Ready
SAVE “HELLO” ENTER
```

Con la orden **SAVE** hemos puesto inmediatamente “en seguridad” el fichero o grupo de datos.

Así de sencillo es esto. Esta pequeña orden de una sólo línea pone en marcha el sistema operativo CP/M. Esto es el primer paso.

Ahora, hagamos uso de la posibilidad de las cadenas de autoarranque. Esto es una línea, que consta de órdenes CP/M, que son realizadas directamente por el sistema CP/M después de su carga. Podemos grabar esta cadena de autoarranque con la orden **INSTALL** del sistema CP/M. La cadena podría tener, por ejemplo, la forma siguiente:

**WS CARTA.TXT<OD>**

Es decir, después de cargar el CP/M, se pone en marcha inmediatamente el WordStar (WS.COM). Al mismo tiempo se carga la correspondencia epistolar CARTA.TXT y habremos alcanzado lo que deseamos.

Si desea suprimir la búsqueda del fichero Hello, permanezca simplemente en RESET o en la conexión de puesta en marcha, por ejemplo, en la tecla **[ESC]** o también en la tecla **[ENTER]**. Entonces no se pierde el tiempo buscando, sino que aparece inmediatamente la indicación de preparado "Ready".

## **Unidades X de disco flexible en los ordenadores CPC 464 con DDI-1, CPC 664 y CPC 6128**

Todo lo que se expone en este capítulo se refiere a los siguientes equipos de aparatos:

CPC 464 + DDI-1	VORTEX F1-X y M1-X
CPC 664	VORTEX F1-X y M1-X
CPC 6128	VORTEX F1-X y M1-X

Estos ordenadores CPC se caracterizan porque están equipados ya desde el principio con una unidad de disco de 3". Por ello, dispone también de un controlador de Floppys. Por este motivo, con estos aparatos se suministra igualmente el sistema operativo CP/M 2.2 y en el ordenador CPC 6128 adicionalmente el sistema operativo CP/M 3.0 (llamado también CP/M Plus).

En el conjunto del suministro de su unidad X VORTEX de disco encontrará un disquete del sistema VORTEX, que no contiene el CP/M.

¿Por qué? Usted ya lo ha “comprado” y únicamente hay que pasarlo al disquete del sistema VORTEX. A continuación le indicamos la forma de hacerlo.

Para los escépticos . . . :

Tenga en cuenta, que no se trata de ningún modo de un truco de “desviación de licencia”. Lo decisivo es que no aumenta la cantidad de ordenadores (puestos de trabajo), en los que se utiliza el sistema operativo CP/M (2.0 y Plus). Usted continua teniendo un sólo ordenador.

## **Producción de un disco flexible del sistema VORTEX para el sistema operativo CP/M 2.2**

Dado que los Floppys VORTEX pueden gestionar, con el sistema CP/M 2.2, no sólo 64 (igual que en el de 3”), sino 124 registros en el índice de contenido del disquete y que la gestión del índice de contenido cuesta también memoria RAM, antes de comenzar con la producción propiamente dicha del disquete del sistema VORTEX hay que reducir el tamaño del sistema CP/M 2.2 al de su disquete de 3”.

Por lo tanto coloque una copia de su disquete de 3” del sistema CP/M 2.2 (en ningún caso debe colocarse el original) en la unidad de disco de 3” y ponga en marcha el sistema operativo CP/M 2.2 mediante la orden de introducción ¡**CPM** **[ENTER]**!. Entonces aparecerá en la pantalla el siguiente aviso de carga:

CP/M 2.2 — Amstrad Consumer Electronics plc

A)

Si observa el contenido del disquete con **A)DIR** **[ENTER]**, encontrará también las dos órdenes **MOVCPM.COM** y **SYSGEN.COM** del sistema CP/M. Precisamente necesitamos estos dos programas para reducir un poco el tamaño del CP/M. No se preocupe, pues necesita-

mos muy poco espacio de memoria. Sólo 256 Bytes (lo que llamamos una página). Su sistema CP/M 2.2 tiene todavía 179 páginas, que reducimos a 178 páginas.

En primer lugar ponemos en marcha el programa **MOVCPM**:

**>MOVCPM 178 \***

CONSTRUCTING 44K CP/M vers 2.2  
READY FOR "SYSGEN" OR  
"SAVE 34 CPM44.COM"

luego ponemos en marcha el programa SYSGEN, con el que escribimos en las pistas del sistema del disquete de 3" el sistema CP/M que acabamos de reducir. Tenga en cuenta, que no se ha pasado previamente a este disquete el "bloqueo de protección de escritura".

**A>SYSGEN \***   
SYSGEN V2.0

Please insert DESTINATION disc into drive A then press any key:  
Do you wish to reconfigure another disc (Y/N) ? :N

Please insert a CP/M system disc into drive A then press any key:  
SYSGEN V2.0 finished

**A>**

(Por favor, introduzca el disco DESTINO en la unidad A y luego pulse cualquier tecla: ¿Desea volver a formatear otro disco (sí/no)? : No.

Por favor, introduzca un disco del sistema CP/M en la unidad A y luego pulse cualquier tecla: ...).

Deje siempre este disquete en la unidad de 3", conteste con "N" (no) y pulse, por lo demás, siempre cualquier tecla, hasta que aparezca otra vez la indicación de preparado del CP/M "A").

Ahora introduzca lo siguiente:

**A>AMSDOS**

(C) 1985 VDOS 2.0-X by vortex GmbH  
BASIC 1.1

Ready



Ahora ha vuelto otra vez al sistema BASIC. Coloque ahora el disquete VORTEX suministrado con el aparato en la unidad X de disco y ponga en marcha el sistema operativo modificado CP/M 2.2, introduciendo:

**!CPM,1**

Ahora aparecerá en la pantalla el aviso de carga:

CP/M 2.2 - Amstrad Consumer Electronics plc  
44K CP/M vers. 2.2-03/86 vortex GmbH

A)

Con ello, tenemos acceso a la unidad X de disco, que es ahora la unidad B con una capacidad de 708K.

El siguiente paso consiste en pasar el CP/M 2.2 del disquete de 3" al disquete VORTEX. Para ello, introduzca las siguientes órdenes:

**A>B:**

**B>SYSGEN**

SYSGEN 2.0 (C) 1985 vortex GmbH

¿Disquete fuente en unidad A o B? A

Colocar el disquete origen en la unidad A y pulsar cualquier tecla.

¿Disquete de destino en unidad A o B? B

Colocar el disquete destino en la unidad B y pulsar cualquier tecla.

¿Todavía un disquete más, sí/no (Y/N)? N

Colocar el disquete con la pista del sistema en la unidad de disco y pulsar cualquier tecla.

Con ello hemos realizado la transferencia del sistema CP/M. Ahora pasemos las órdenes del sistema CP/M al disquete VORTEX:

XSUB	.COM	SUBMIT	.COM	LOAD	.COM
DUMP	.COM	MOVCPM	.COM	ED	.COM
STAT	.COM	PIP	.COM	DDT	.COM

Para ello introducimos las órdenes siguientes:

```
B>A: [ENTER]
A>PIP [ENTER]
*B:=XSUB.COM [ENTER]
*B:=SUBMIT.COM [ENTER]
"
"
*B:=STAT.COM [ENTER]
*B:=PIP.COM [ENTER]
[ENTER]
A>
```

Ya hemos terminado, en la unidad X de disco está ahora su nuevo disquete del sistema VORTEX CP/M 2.2, que necesita, si desea cargar el sistema CP/M desde la unidad X.

Introduzca ahora la orden:

```
A>AMSDOS [ENTER]
```

con lo que volvemos al sistema BASIC:

```
(C) 1985 VDOS 2.0-X by vortex GmbH
BASIC 1.1
```

Ready

Ahora ponga en marcha el sistema CP/M 2.2 desde la unidad de disco VORTEX, introduciendo:

```
!CPM,2 [ENTER]
```

Aparece entonces el siguiente aviso de carga:

```
44K CP/M vers.2.2-03/86 vortex GmbH
```

```
A>
```

La unidad A es ahora la unidad X VORTEX de disco con 708K y 128 direcciones de índice. La unidad de disco B es la unidad de 3" con aproximadamente 170K y 64 registros de índice.

Los siguientes programas de servicio en el disquete de 3" **no deben ser realizados** con este sistema CP/M:

BOOTGEN.COM DISCKIT2.COM SETUP.COM  
FORMAT.COM SYSGEN.COM Todos los ficheros LOGO

para trabajar con LOGO y con la unidad X, debe colocar una copia del disquete LOGO en la unidad de 3" y ponerlo en marcha con **!CPM** **[ENTER]**. Tenga en cuenta que la unidad X de disco tiene aquí sólo una capacidad de aproximadamente 170K con 64 registros de índice.

Por eso, lo mejor es hacerse ahora inmediatamente una copia de trabajo del disquete del sistema VORTEX CP/M 2.2 que acabamos de producir. A partir de ahora, deberíamos trabajar únicamente con esta copia de trabajo y con eventuales copias de la misma, pero no con el disquete del sistema preparado anteriormente.

Para la producción de la copia, el disquete del sistema recién producido permanece en la unidad de disco; el disquete de 3" ya no lo necesitamos más. Además, precisamos un nuevo disquete VORTEX en blanco.

Todavía no hemos abandonado el CP/M cargado anteriormente y ahora ponemos en marcha el programa de servicio VORTEX, con el que podemos copiar disquetes enteros, mediante la introducción de:

**A)DISKTOOL [ENTER]**

A continuación obtenemos en la pantalla lo siguiente:

DISKTOOL 1.0 (C) 1986 vortex GmbH



- 1 — Inicializa disquete
  - 2 — Genera copia completa disquete
  - 3 — Copia pistas sistema CP/M
  - 4 — Copia sólo sector de parámetros
- Vuelta a CP/M con **[ESC]**

Pulse la tecla correspondiente. Pulse ...

Entonces pulse la tecla "2", puesto que queremos copiar un disquete completo. Después de pulsada, aparece en la pantalla el texto siguiente:

DISKTOOL 1.0 (C) 1986 vortex GmbH

Subrutina COPIA DISQUETE COMPLETO

Unidad fuente?  
Unidad destino?  
Inicializa destino (S/N)?  
Verificar disquete destino (S/N)?  
Vuelta a pregunta anterior **[ESC]**

Pulse la tecla correspondiente. Pulse ...

El origen es el disquete del sistema VORTEX producido anteriormente. El destino es el nuevo disquete VORTEX, todavía vacío. Las unidades fuente y destino son la "A".

Por lo tanto, conteste las preguntas siguientes de este modo:

Unidad fuente ? A

Unidad destino ? A

La pregunta "inicializa formato ..." contéstela con "Y" (sí), porque el nuevo disquete vacío debe ser primeramente formateado.

La pregunta "verificar disquete ..." contéstela también con "Y" (sí), porque deseamos excluir también hasta el más pequeño error de copia.

El texto continuo le informa siempre cuándo y qué disquete debe colocar en la unidad A. Tenga en cuenta lo siguiente:

**Si olvida alguna vez cambiar el disquete, resultará destruido su original para copias. ¡Mucho cuidado!**

Ahora deberá estar en la unidad de disco la copia recién producida. Déjela allí y pulse la tecla **[ESC]**.

Entonces se sale de DISKTOOL y aparece otra vez la indicación de preparado "A)" del CP/M. El disquete situado en la unidad X de disco es la copia exacta del producido anteriormente en el sistema CP/M 2.2.

## **Ordenadores modelo CPC 464 y CPC 664 con unidad de disco flexible X y tarjeta de memoria RAM VORTEX**

¿Ha incorporado ya una tarjeta RAM VORTEX a su ordenador CPC? Si no lo ha hecho todavía, ya va siendo hora. Está desperdiciando demasiadas posibilidades. Así no puede disfrutar de su CP/M 2.2 de 62 K. No dispone de "dBase", ni de "Multiplan", ni de un "WordStar" adecuado, ni de ...

Pero, si ya ha incorporado la tarjeta, puede trabajar con su nuevo Floppy VORTEX en el "gran CP/M". Para ello, debe poner en marcha únicamente el programa 'XPATCH.COM', que se encuentra en el disquete del sistema VORTEX CP/M 2.2 recién producido.

Este programa funciona exactamente igual que el programa "PATCH.COM" (lo sustituye totalmente), que forma parte del suministro de la tarjeta RAM.

## **Confección de un disco VORTEX para el sistema operativo CP/M Plus (sólo en el ordenador tipo CPC 6128)**

Si posee un ordenador CPC 6128, dispone en él del sistema CP/M Plus. Esto no debe ser desperdiciado de ningún modo. También aquí deberemos confeccionar el correspondiente disquete VORTEX, desde el cual puede cargar el sistema CP/M.

Antes de que pueda confeccionar el disquete del sistema VORTEX CP/M Plus, deberá haber confeccionado (como ya hemos explicado

antes) un disquete VORTEX CP/M 2.2. Introduzca en la unidad de 3" el disquete del sistema de 3" CP/M 2.2 modificado anteriormente.

Introduzca el disquete del sistema VORTEX CP/M 2.2 en la unidad de disco X y ponga en marcha el sistema operativo modificado CP/M 2.2 desde la unidad de 3", mediante la introducción de:

**!CPM,1** **[ENTER]**

Entonces aparece el aviso:

```
CP/M 2.2 – Amstrad Consumer Electronics plc  
44K CP/M vers. 2.2-03/86 vortex GmbH  
A)
```

Ponga en marcha ahora el programa Patch del CP/M Plus, que se encuentra en el disquete VORTEX suministrado con el equipo, que ahora es ya su disquete original VORTEX del sistema CP/M 2.2, introduciendo las órdenes:

**A>B:** **[ENTER]**

**B>XPATCH30** **[ENTER]**

XPATCH30 sólo puede utilizarse en un sistema CP/M 2.2, que haya sido puesto en marcha en un ordenador CPC 6128 con la orden **!CPM,1** **[ENTER]**. Si intenta poner en marcha XPATCH30 con el sistema CP/M Plus, recibirá el aviso siguiente:

XPATCH30 sólo puede ponerse en marcha con el sistema CP/M 2.2 y se interrumpirá el programa.

Pero, supongamos que todo funciona según lo previsto; entonces, después de poner en marcha XPATCH30, aparece el aviso:

**CP/M 3.0 (V1.1S-X) PROGRAMA MAESTRO**  
**(C) 1986 vortex s. a.**

(XPATCH30 cambia su fichero .EMS de CP/M Plus para uso con unidad X.)

Introduzca una copia del disquete sistema 3" CP/M Plus en unidad 3" y pulse cualquier tecla.

Coloque ahora una copia de su disquete del sistema CP/M Plus en la unidad de disco incorporada de 3" y pulse cualquier tecla. XPATCH30 proporciona el siguiente aviso:

Cargando fichero C10CPM3.EMS . . . . .

Por lo tanto, ahora XPATCH30 registra el sistema operativo CP/M Plus (que se encuentra en el ordenador 6128 en un fichero de la extensión EMS) desde la copia del disquete de 3" CP/M Plus y lo modifica para su utilización con la unidad de disco X. A continuación, XPATCH30 pregunta si desea usted disponer de un disquete de 3" o de 5.25" del sistema CP/M Plus. Supongamos que deseamos un disquete de 5.25"; entonces respondemos con N (No) en la forma siguiente:

Esta definida la unidad 3" como unidad A (S/N) ? N

Ahora, el XPATCH30 escribe en el disquete de 5.25" en la unidad de disco B (allí debe haber naturalmente un disquete formateado, ¿o no lo hay?), el sistema operativo modificado CP/M Plus, que recibe el nombre de CPM3.EMS:

Escribiendo CPM3.EMS . . .

Naturalmente, el disquete de 5.25" necesita también un nuevo sector Boot, que sea adecuado para cargar el sistema operativo CP/M Plus. Por lo tanto, el XPATCH30 emite en este caso el aviso:

Escribiendo sector de arranque

5.25"/3.5" creado disquete maestro CP/M Plus-X

A continuación, el XPATCH30 le comunica otra vez en forma abreviada las posibilidades que tiene de poner en marcha el sistema CP/M Plus:

Tiene tres formas de arrancar el CP/M Plus:

1) Disquete original CP/M Plus en unidad 3":

Arranque desde unidad de 3" por medio de ¡CPM [ENTER].

Se tomará unidad B como simple cara 40 pistas y 64 directorios.  
Atención: Nunca use disquete 5.25" escrito de esta manera -----  
con VDOS o CP/M arrancado con **CPM,1** o **CPM,2**.

- 2) Disquete CP/M Plus modificado en unidad 3":  
Arranque desde unidad 3" ¡**CPM** [ENTER] o ¡**CPM,1** [ENTER]  
Configuración: unidad A=3", unidad B=5.25" con 708K y 128 directorios.
- 3) Arranque desde unidad 5.25"/3.5" ¡**CPM,2** [ENTER].  
Configuración: unidad A=5.25" con 708K y 128 directorios.  
unidad B=3"

Introduzca el disquete CP/M 2.2 con XPATCH30 en unidad 3" y pulse cualquier tecla

Le recomendamos que haga inmediatamente una copia de trabajo del disquete del sistema CP/M Plus, que acaba de confeccionar; (con DISKTOOL, tal como se ha explicado al confeccionar un disquete del sistema VORTEX CP/M 2.2).

A continuación, un par de observaciones:

- 1) La aceleración de la imagen en la pantalla FAST.COM y la orden ¡**XMON** del monitor (programa) Z80 sólo trabajan con el sistema operativo CP/M 2.2 de 44K, pero no con el CP/M Plus. Los programas de servicio DISKTOOL, FILECOPY, SYSGEN, INSTALL, CASCOPY, VDOS trabajan sólo con un sistema CP/M 2.2 (también de 62K).
- 2) Si pone en marcha el sistema CP/M Plus desde el BASIC (¡**CPM,2** [ENTER]), debe tener cuidado de que haya un disquete en la unidad de disco de 3". Si no hay ninguno, el sistema CP/M Plus sólo detecta una unidad de disco. (Esto también le ocurrirá sin el Floppy VORTEX, si conecta, por ejemplo, la segunda unidad de disco de 3" a su ordenador CPC 6128). El aviso correcto de carga aparece entonces en la forma siguiente:

```
CP/M Plus  Amstrad Consumer Electronics plc  
v 1.0, 61K TPA, 2 disc drives
```

A)



- 3) Se pueden traspasar desde el disquete de 3" al disquete VORTEX todos los programas del CP/M Plus, excepto el programa:

**DISCKIT 3.COM**

Si utiliza este programa junto con las unidades de disco X, en determinadas circunstancias podrá tener consecuencias devastadoras.

Por lo tanto, si desea por ejemplo, trasladar ficheros o grupos de datos en el sistema CP/M Plus desde la unidad de disco A a la B, o al revés, utilice para ello el programa PIP. Si desea copiar disquetes enteros, salga del sistema CP/M Plus, ponga en marcha el CP/M 2.2 y trabaje allí con DISKTOOL.

## **La carga del sistema operativo CP/M**

En las unidades de disco X existen 3 posibilidades de cargar el sistema operativo CP/M:

## **La orden ¡CPM del sistema operativo VDOS**

El sistema operativo CP/M es cargado por la unidad de disco de 3" desde una copia del disquete original del sistema Amstrad CP/M. Entonces obtenemos el siguiente aviso de carga:

```
¡CPM ENTER  
CP/M 2.2 - Amstrad Consumer Electronics plc  
A>
```

La unidad de disco A de 3" tiene aproximadamente 170K.

La unidad de disco B es la unidad de disco X con 170K aproximadamente.

Aquí se utiliza la unidad de disco B como unidad de una sola cara de 40 pistas con una capacidad de aproximadamente 170K. Esto se

debe a que el sistema original AMSDOS sólo puede gestionar unidades de disco de esta capacidad. (Sólo dicho sistema garantiza una compatibilidad al 100% con el software de 3" existente).

No obstante, tenga en cuenta que estas 40 pistas no están distribuidas por todo el radio del disquete (en el caso normal), sino que la unidad de disco puede continuar recorriendo 80 pistas, pero sólo utiliza las primeras 40 de la cara principal o delantera.

Esta posibilidad de carga es idéntica a la del caso de que usted no tuviera la unidad de disco X, sino por ejemplo, una unidad de disco auxiliar de 3".

Los disquetes confeccionados y utilizados con este sistema operativo CP/M no se pueden utilizar, en el caso de que esté conectado el sistema operativo VDOS. Mantenga cuidadosamente separadas estas dos "clases de disquetes" (señalícelas en su parte exterior). Si, a pesar de todo, las utiliza, deberá contar con una pérdida total de los datos.

Los disquetes que necesita aquí para su unidad de disco X, deben ser ajustados a este formato especial con la orden **FORMAT** del sistema CP/M o con **DISCKIT** (en la copia del disquete original del sistema Amstrad CP/M).

## La orden ¡CPM,1 del sistema operativo VDOS

El sistema operativo CP/M es cargado por la unidad de disco de 3" desde una copia modificada del disquete original del sistema Amstrad CP/M. Entonces se recibe el siguiente aviso de carga:

¡CPM,1

CP/M 2.2 - Amstrad Consumer Electronics plc  
44K CP/M vers. 2.2-03/86 vortex GmbH

A>

La unidad de disco A es la unidad de 3" con aproximadamente 170K.

La unidad de disco B es la unidad de disco X con 708K.

Lo que se ha modificado en este sistema CP/M lo ha visto en el párrafo "unidades X de disco en los ordenadores CPC 464 con DDI-1, CPC 664 y CPC 6128". Este sistema operativo CP/M sólo tiene 178 páginas de tamaño (en lugar de 179 como anteriormente; esto se debe a que se gestiona la unidad de disco X con 128 registros de índice).

## **La orden ¡CPM,2 del sistema operativo VDOS**

El sistema operativo CP/M se carga desde el disquete del sistema en la unidad de disco X. (Confección de este disquete, véase en el párrafo "unidades X de disco en los ordenadores CPC 464 + DDI-1, CPC 664 y CPC 6128"). Entonces recibimos el siguiente aviso de carga:

```
¡CPM,2 
```

```
44K CP/M vers. 2.2-03/86 vortex GmbH
```

```
A>
```

La unidad de disco A es la unidad de disco X con 708K.

La unidad de disco B es la unidad de 3" con aproximadamente 170K.

### **Observación importante:**

Las tres órdenes de carga mencionadas para el sistema operativo CP/M hacen que en el ordenador CPC 6128 y disquete colocado CP/M Plus se realice la carga del sistema CP/M Plus.

Si pone el ordenador a cero (pulsando simultáneamente las teclas ,  y ) , si efectúa la Desconexión/Conexión del ordenador, o bien si carga el sistema operativo CP/M, se pierde otra vez el intercambio realizado con la orden ¡X.

## **Sistema operativo CP/M con la unidad de disco flexible X VORTEX**

Las órdenes del sistema operativo CP/M VORTEX aquí descritas las necesita para trabajar con la unidad de disco flexible X VORTEX en el sistema CP/M 2.2. Por lo tanto, si ha cargado el CP/M 2.2 con `!CPM,1` `[ENTER]` o bien con `!CPM,2` `[ENTER]`, entonces puede utilizar estas órdenes.

Las informaciones sobre el sistema operativo CP/M y sus órdenes puede verlas en los correspondientes manuales, que habrá recibido al comprar su ordenador CPC (664 y 6128) o su estación de Floppys DDI-1 (CPC 464).

### **La orden FILECOPY del sistema operativo CP/M: copiar ficheros o grupos de datos**

Si no era posible con la orden **PIP** del sistema operativo CP/M copiar ficheros o grupos de datos o incluso disquetes enteros, como poseedor sólo de una unidad de disco (recordará que ello se debía siempre a **BDOS Errors** y al **CTRL C**), esto ya no es ningún problema con la orden **FILECOPY**. Además, la orden **FILECOPY** es notablemente más sencilla de manejar, en lo que se refiere a su utilización.

También se suprime el complicado trabajo de copiar ficheros entre diferentes campos del **USUARIO**. Además de la unidad de disco origen y destino, puede indicar todavía una zona de **USUARIO** de origen y otra de destino.

Además se refuerza el servicio de las 16 unidades de disco posibles (Floppys, discos duros o Floppys de memoria RAM). Inmediatamente después de poner en marcha **FILECOPY**, verá qué unidad de disco está disponible y qué capacidades de memoria formateadas (reales) tiene usted.

Aquí hay que mencionar una pequeña peculiaridad: si ha incorporado una ampliación de memoria **VORTEX** en su ordenador CPC, entonces tiene, a parte de memoria RAM adicional de 128K, un Floppy

de memoria RAM bajo el sistema operativo CP/M. Para este Floppy de memoria RAM está reservada siempre la denominación "C" de la unidad de disco. Sin embargo, tenga en cuenta que con FILECOPY no se puede realizar ninguna transferencia de datos entre unidades externas.

FILECOPY es únicamente un programa de servicio para la transmisión de datos de una a otra unidad de disco.

Se pone en marcha el programa FILECOPY mediante la introducción de:

**A) FILECOPY** **[ENTER]**

FILECOPY dispone de dos menús:

El menú de "ajuste", en el que se ajusta la fuente y el destino, y el menú de "copiar" en el que se pueden "elegir" los ficheros o grupos de datos que se desean copiar. En ambos menús necesitaremos principalmente las teclas del cursor, así como las teclas **[COPY]** y **[ENTER]**. Otras teclas se necesitan muy raras veces.

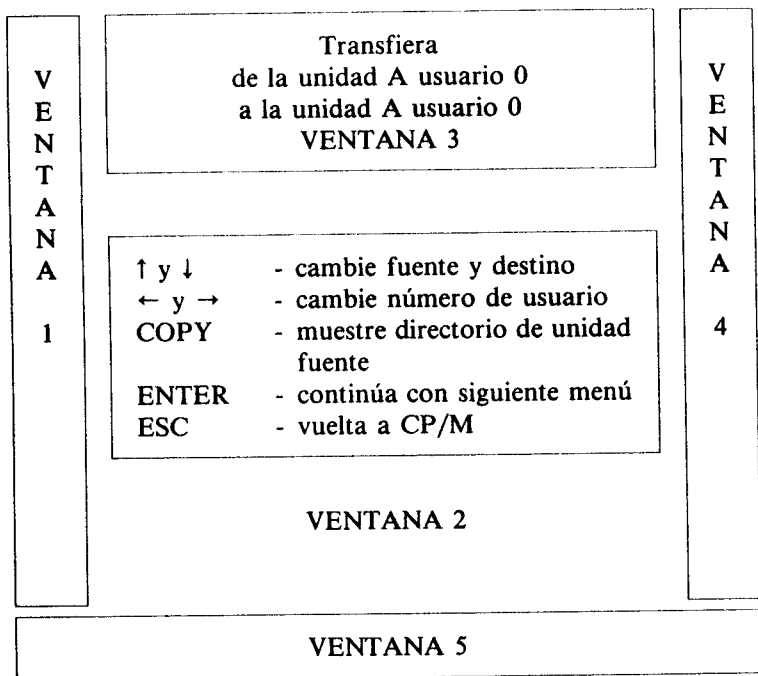
La pantalla de imagen está subdividida siempre en 5 "ventanas" (véase más abajo).

- La ventana 1 indica los ficheros o grupos de datos, que se encuentran en el disquete fuente o que deben ser copiados (exposición inversa video).
- La ventana 2 nos muestra siempre "con qué" teclas y "qué" se puede hacer.
- La ventana 3 muestra qué unidad de disco y qué campo del USUARIO es la fuente, así como qué unidad de disco y qué campo del USUARIO es el destino.
- La ventana 4 nos indica las unidades de disco existentes y sus capacidades de memoria formateadas. Aquí tenemos dos posibilidades, según que carguemos el sistema operativo CP/M con **!CPM,1** **[ENTER]** (unidad de disco A = unidad de disco de 3" con 169K y unidad B = unidad de disco X con 704K), o bien con **!CPM,2** **[ENTER]** (unidad A = unidad de disco X con 704K y unidad B = unidad de 3" con 169K).

En la ventana 3 se pueden modificar también los números de USUARIO. Éstos van desde 0 hasta F; éstas son cifras hexadecimales. A continuación exponemos una breve “traducción” o equivalencia de las mismas:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 decimal  
 0 1 2 3 4 5 6 7 8 9 A B C D E F hexadecimal

- La ventana 5 es lo que se dice una “cinta de movimiento continuo”. El texto que se reproduce va avanzando con una velocidad continua a lo largo del borde inferior de la pantalla. Aquí se indican los avisos de errores y los ruegos de cambiar el disquete y otros avisos semejantes:



La ventana 5 nos indica el siguiente aviso: “pulse la tecla correspondiente”.

Aquí es interesante la ventana 2. Con la tecla "cursor hacia arriba" se puede modificar la unidad fuente. Hay aquí tantas posibilidades de elección, como unidades de disco estén indicadas en la ventana 4.

Con la tecla "cursor hacia abajo" se puede modificar el ajuste de la unidad de disco de destino. Con la tecla "cursor hacia la izquierda" se puede ajustar el campo del USUARIO en el disquete fuente; es decir, de este campo se sacan los datos a copiar. Con la tecla "cursor hacia la derecha" se puede ajustar el campo del USUARIO en el disquete de destino; es decir, en este campo se copian los datos que proceden del disquete fuente.

¿Recuerda todavía qué complicado era esto con la orden **PIP**?

Si pulsa la tecla **COPY**, en la ventana 1 aparece el índice de contenido del disquete fuente, ordenado alfabéticamente. Pulsando cualquier tecla, puede parar esta indicación en la pantalla y luego la puede volver a poner en marcha pulsando otra vez.

Si pulsamos la tecla **ESC** se termina el programa FILECOPY y entonces volvemos otra vez al sistema CP/M, apareciendo el siguiente texto en la banda de movimiento continuo.

"Introduzca disquete CP/M en unidad A y pulse cualquier tecla"

Si pulsa la tecla **ENTER**, tendrá que haber fijado previamente la fuente y el destino. En la ventana 2 aparece un contenido nuevo; la ventana 1 se borra y el índice de contenido de la fuente se desplaza desde abajo hacia arriba hasta las marcas (en los lados derecho e izquierdo de la ventana 1).

La nueva ventana 2 aparece entonces en la forma siguiente:

↑ y ↓	- avance un fichero adelante/atrás
COPY	- marque fichero a copiar
W	- marque por medio de fichas vacías (tarjetas libres)
ENTER	- copie fichero indicado
ESC	- vuelta al menú anterior

La cinta de texto en movimiento continuo indica otra vez "Pulse la tecla correspondiente".

Con las teclas de “cursor hacia arriba” y “cursor hacia abajo” se pueden colocar en la ventana 1 los ficheros que se desee, entre las dos marcas. Si ahora se pulsa la tecla **COPY**, aparece el “video inverso” de este fichero. Hemos marcado este fichero como para copiarlo. Así podemos marcar para copiar toda una serie de ficheros, incluso todos los ficheros (para ello, habrá que ir de nuevo al comienzo del índice contenido y pulsar permanentemente la tecla **COPY**). Si llegamos a un fichero que ha sido marcado y pulsamos entonces la tecla **COPY**, se elimina el marcado realizado anteriormente.

Si pulsamos luego la tecla **ENTER**, comienza la operación de copiado. Si no son iguales la fuente y el destino (dos unidades de disco, o bien una unidad de disco y el Floppy de memoria RAM), no necesitamos cambiar los disquetes. El programa FILECOPY nos comunica entonces cuándo está leyendo desde la fuente y cuándo está escribiendo el disquete de destino.

Si sólo tenemos una unidad de disco, o bien deseamos a propósito copiar sólo con una unidad de disco, la cinta de texto de movimiento continuo nos avisa cuándo tenemos que sacar el disquete fuente y colocar en su lugar el disquete de destino.

También existe la posibilidad de copiar por medio de fichas vacías o tarjetas libres. Marque una “W” y entonces se borra el contenido de la ventana 2 y puede introducir allí una denominación de tarjeta libre. En la fuente y en el destino puede moverse algo dentro del menú, como ya se ha dicho antes, con las teclas “cursor hacia arriba” y “cursor hacia abajo”.

En todo caso, la “búsqueda” (exploración) en el disquete fuente se realiza con “W” y la denominación introducida de la tarjeta libre y entonces se copian en el destino todos los ficheros o grupos de datos que quepan en el esquema de la tarjeta libre. Los símbolos admitidos en la tarjeta libre son, como es normal en el sistema operativo CP/M, el asterisco “\*” y el signo de interrogación “?”.

Un esquema de tarjeta libre podría tener, por ejemplo, la forma siguiente:

**WS\*.\***



Todos los ficheros existentes en el disquete fuente, que comiencen con WS (ficheros WordStar) y lleven a continuación cualquier signo en el nombre y tipo del fichero (por ejemplo, WS.COM, WSOVLY1.OVR, WSMGS.OVR o bien, WSU.COM) se copian entonces en el disquete de destino. Otro esquema de tarjeta libre sería, por ejemplo:

\*.\*

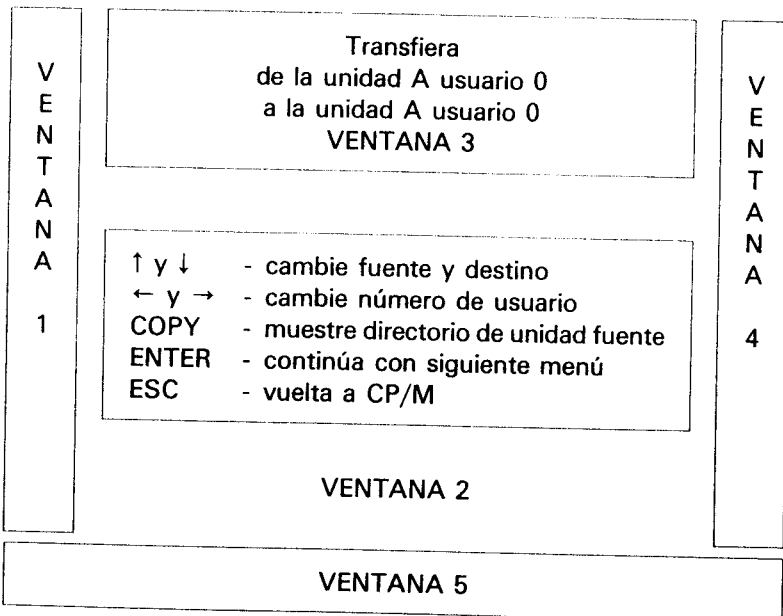
Con ello habríamos trasladado los ficheros del disquete fuente al de destino. El nombre y el tipo de fichero son absolutamente indiferentes y pueden elegirse a voluntad. En este esquema caben todos los ficheros del disquete fuente.

Ahora pongamos un ejemplo. Ponga en marcha el programa FILECOPY, mediante la siguiente introducción:

A>FILECOPY

Entonces se borra la pantalla y aparece el cuadro siguiente:

FILECOPY V3.0 (C) 1986 by vortex GmbH



La ventana 3 nos muestra la unidad de disco fuente y de destino utilizados en el momento. Al poner en marcha el programa FILECOPY aparece aquí siempre automáticamente la disposición indicada en el cuadro anterior.

La ventana 4 nos muestra las unidades de disco disponibles; en nuestro caso tenemos dos unidades de disco (unidad A = unidad de disco X; unidad B = unidad de 3"); por lo tanto, el sistema CP/M ha sido cargado con ¡CPM,2 [ENTER]). La ventana 4 indica entonces concretamente lo siguiente:

capacidad disponible de unidades de disco	
A:	704K
B:	169K
C:	-----
D:	-----
E:	-----
F:	-----
G:	-----
H:	-----

K:	-----
L:	-----
P:	-----

Como ve, el programa FILECOPY es muy "inteligente" y puede gestionar todas las unidades de disco posibles en el sistema CP/M (16 unidades o aparatos A . . . P). En la ventana 5 vemos en la cinta de texto de movimiento continuo el texto siguiente:

Pulse la tecla correspondiente Pulse la tecla . . .

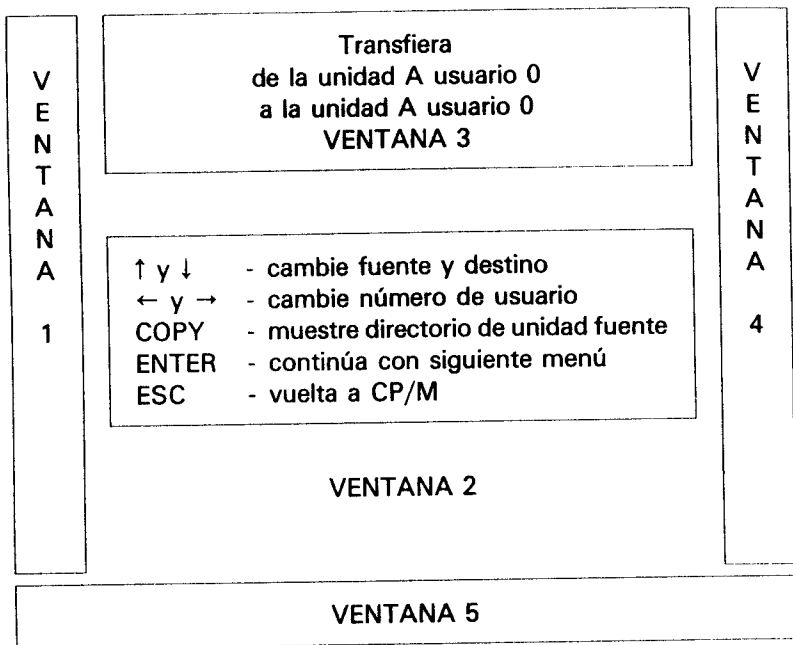
Supongamos que el disquete, desde el que hemos puesto en marcha el programa FILECOPY, no es el disquete fuente; entonces sacamos el disquete colocado todavía en la unidad de disco X y en su lugar

lugar colocamos el disquete fuente. Veamos sencillamente lo que hay dentro de este disquete, en cuanto a ficheros. Para ello pulse la tecla **COPY**. Entonces, en la ventana 1 aparece lo siguiente:

```
INSTALL .COM
TEST    .BAS
TEST    .COM
UTIL    .COM
VDOS    .COM
WS      .COM
WSMSGs .OVR
WSOVLY1.OVR
```

Por lo tanto, todos estos son ficheros o grupos de datos que se encuentran en el disquete fuente. Si nos interesamos por un determinado fichero: "VDOS.COM" y lo queremos trasladar al disquete de destino, que está junto a nosotros.

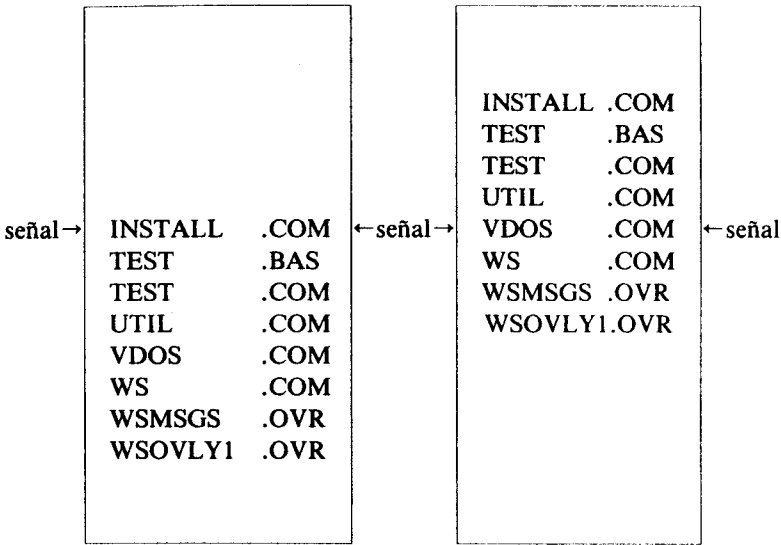
En la ventana 2 encontrará el punto del menú "ENTER - Continuar hasta el siguiente menú". En el siguiente menú podemos buscar el fichero "o los ficheros", que deseamos copiar. Pulse entonces la tecla **ENTER**. A continuación, la ventana 2 recibe un nuevo contenido y se borra la ventana 1.



La ventana 1 se llena de nuevo con el índice de contenido del disquete fuente (véase la figura 1).

(Figura 1)

(Figura 2)



Mediante las teclas “cursor hacia arriba/hacia abajo” desplazamos arriba todo el índice de contenido, hasta que el fichero VDOS.COM quede parado entre las “dos señales” (véase la figura 2).

Pulse ahora la tecla **COPY**. “VDOS.COM” está señalado en video inverso.

Pulsando la tecla **ENTER** ponemos en marcha la operación de copiado. Se borra la indicación de la ventana 1 y aparece el fichero individual VDOS.COM. Entonces la ventana 5 nos indica lo siguiente:

“Introduzca disquete fuente y pulse cualquier tecla”

Se borra la ventana 2 y se nos comunica:

“Leyendo datos. Espere ...”

El texto informativo de la banda de movimiento continuo nos indica ahora:

Introduzca disquete destino y pulse cualquier tecla"

Entonces sacamos el disquete fuente y colocamos en su lugar el disquete de destino (este disquete debe ser previamente formateado) y luego pulsamos cualquier tecla.

Entonces la ventana 2 contiene una nueva información:

"Escribiendo datos. Espere ..."

Una vez terminado el copiado, la ventana 2 se llena otra vez con el primer texto del menú. Pulse otra vez la tecla **ESC**, y entonces el aviso de la banda de texto de movimiento continuo se cambia al siguiente:

"Introduzca disquete CP/M en unidad A y pulse cualquier tecla"

Extraiga ahora el disquete fuente y coloque el disquete, con el que ha puesto anteriormente en marcha el sistema operativo CP/M y del que ha cargado también el programa FILECOPY. Pulse luego cualquier tecla y aparecerá otra vez la indicación de preparado "A)".

## **La orden XMON del sistema operativo CP/M: el monitor Z80 con el sistema CP/M**

El monitor (programa de corrección) puede utilizarse también con el sistema operativo CP/M.

**Atención:** Únicamente el CP/M 2.2, sin ampliación de memoria RAM.

Por motivos propios del sistema, con el CP/M quedan bloqueadas las órdenes siguientes: **I, O, Y**; todas las demás órdenes se pueden utilizar sin ninguna limitación.

El programa monitor es solicitado con la ayuda del programa llamado XMON.COM. La sintaxis de la orden es la siguiente:

**XMON** **ENTER** o bien

**XMON** nombre del fichero.tipo **ENTER**

o sea, por ejemplo:

**A)XMON** ENTER

o bien

**A)XMON STAT.COM** ENTER

La primera variante únicamente solicita el programa monitor, mientras que la segunda, carga en la memoria además el fichero o grupo de datos “nombre del fichero.tipo” a partir de 100 H. El monitor anuncia su presencia, después de la entrada, con:

<comienzo longitud 0100>

\*

Comienzo: comienzo del fichero o grupo de datos cargado (en el sistema CP/M siempre 0100).

Longitud: longitud del fichero o grupo de datos cargado.

Una vez haya salido del programa monitor con ESC, puede asegurar en el disquete la memoria a partir de 0100H, con la orden **SAVE** del sistema CP/M.

## **La orden SYSGEN del sistema operativo CP/M: escribir el CP/M en pistas del sistema**

Como sabe, con la orden **MOVCPM** se construye un sistema CP/M de “nuevo” tamaño. Lo que necesitamos aquí todavía es una posibilidad de pasar el CP/M modificado a las pistas de nuestro disquete del sistema, que se encuentra en la memoria o en forma de un fichero o grupo de datos en el disquete.

Precisamente para ello, se dispone del programa SYSGEN (SYStem GENerator = generador del sistema) en el disquete del sistema.

SYSGEN es una orden transitoria de CP/M. Antes de continuar, véase cómo están configuradas en realidad las dos pistas del sistema.

P  
i  
s  
t  
a

	Sectores								
	01	02	03	04	05	06	07	08	09
0	Boot	PARA	vacío	vacío	vacío	vacío	vacío	CP/M	CP/M
1	CP/M	CP/M	CP/M	CP/M	CP/M	CP/M	CP/M	CP/M	CP/M

Cada sector tiene una capacidad de memoria de 512 Bytes. El sector "Boot", que es el primer sector en todo el disquete, contiene el denominado "Bootstrap-Loader"; se trata de un pequeño programa, que se carga directamente en la memoria mediante la introducción de **!CPM,2** **[ENTER]** desde el disquete VORTEX y luego se pone en marcha. Este pequeño "cargador" recoge el sistema CP/M propiamente dicho desde las pistas 0 y 1, lo instala en la memoria y deja que comience a funcionar en el momento oportuno. Después de esta operación aparece ya el aviso de carga del CP/M y el aviso de preparado (A) con el cursor a continuación encendiéndose intermitentemente.

El sector "PARA" contiene, por ejemplo, cambios de definiciones del teclado o el color de la pantalla de imagen, que luego se ajusta automáticamente al cargar CP/M. El contenido de este sector se puede modificar con el programa INSTALL (descrito más adelante).

Los sectores 3, 4, 5, 6 y 7 de la pista 0 están todavía vacíos. Si tiene una ampliación VORTEX de memoria RAM, estos sectores estarán también ocupados. Todos los sectores señalizados con "CP/M", contienen el sistema operativo CP/M. Precisamente estos sectores (y sólo éstos) son escritos por la orden **SYSGEN**.

**A<SYSGEN** **[ENTER]**

**SYSGEN 2.0** (C) 1985 vortex GmbH

Disquete fuente en unidad A o B? **[ENTER]**

Disquete destino en unidad A o B? A

Introduzca disquete destino en unidad A y pulse cualquier tecla

Otro disquete S/N? N

Introduzca disquete sistema en unidad A y pulse cualquier tecla



Mediante la primera orden **ENTER**, comunicamos al **SYSGEN** que el nuevo CP/M está en la memoria y debe ser sacado de allí. Mediante la introducción de “**A**”, como respuesta a la pregunta “¿Disquete de destino en unidad A o B?”, indicamos que el nuevo sistema CP/M debe ser escrito en las pistas del sistema del disquete en la unidad A.

**A<SYSGEN** **ENTER**

SYSGEN 2.0 (C) 1985 vortex GmbH

Disquete fuente en unidad A o B? A

Introduzca disquete fuente en unidad A y pulse cualquier tecla.

Disquete destino en unidad A o B? B

Introduzca disquete destino en unidad B y pulse cualquier tecla

Otro disquete S/N? N

Introduzca disquete sistema en unidad A y pulse cualquier tecla

En este ejemplo, el sistema operativo CP/M no ha sido modificado previamente con **MOVCPM**. Se realiza aquí simplemente su lectura desde las pistas del sistema del disquete en la unidad A y luego se escribe en las pistas del sistema de disquete en la unidad B.

**A>SYSGEN** **ENTER**

SYSGEN 2.0 (C) 1985 vortex GmbH

Disquete fuente en unidad A o B? **ENTER**

Disquete destino en unidad A o B? **ENTER**

Introduzca disquete sistema en unidad A y pulse cualquier tecla

Con esta utilización de la orden **SYSGEN** no pasa absolutamente nada. Ciertamente se lee en la memoria un “nuevo” CP/M, pero luego no se escribe en las pistas del sistema.

## La orden **DISKTOOL** del sistema operativo CP/M

- Inicializa disquete
- Genera copia completa de disquete
- Copia pistas sistema CP/M
- Copia sólo sector de parámetros

La orden **DISKTOOL** (transitoria) del sistema operativo CP/M es una "herramienta de disquete" en el estricto sentido de la palabra. Dicha orden dispone de multitud de funciones importantes, únicamente, con las cuales es posible trabajar con los disquetes en la unidad de disco.

Igual que el programa FILECOPY, también el DISKTOOL es un programa totalmente controlado por el menú, el cual realiza las funciones deseadas con sólo pulsar un botón.

Si pone en marcha el programa DISKTOOL con **A<DISKTOOL** **[ENTER]**, se borra la pantalla de imagen y aparece en la pantalla el cuadro siguiente:

DISKTOOL 1.0            (C) 1986 vortex GmbH



- 1 - Inicializa disquete
  - 2 - Genera copia completa de disquete
  - 3 - Copia pistas sistema CP/M
  - 4 - Copia solo sector de parámetros
- Vuelta a CP/M con ESC

Pulse la tecla correspondiente Pulse . . .

Como ve, desde aquí se puede hacer una desviación o ramificación de cuatro submenús, o bien abandonar el programa con DISKTOOL **[ESC]** y volver al sistema operativo CP/M.

Además, igual que en el programa FILECOPY, aquí tenemos también un texto de banda de movimiento continuo, que nos pide el cumplimiento de determinadas acciones, que estén en relación con la función elegida.

La pantalla está dividida en dos ventanas; la ventana superior muestra la función en la que acabamos de realizar la selección; la inferior muestra informaciones sobre la puesta en marcha y el desarrollo de la función correspondiente.

Hagamos pasar ahora punto por punto el menú principal, para ver lo que se oculta detrás de cada una de las funciones.

Si pulsamos la tecla "1", se modifica el contenido de la pantalla:

DISKTOOL 1.0 (C) 1986 vortex GmbH

Subrutina INICIALIZA DISQUETE

Inicializando en unidad ?  
Vuelta a pregunta anterior

Pulse la tecla correspondiente Pulse . . .

Está claro que aquí sólo podemos indicar unidades de disco. Con este "Formateador" no se pueden procesar un Floppy de memoria RAM (VORTEX SP 512) o una placa fija (VORTEX WD 10 o bien WD 20). La elección de la unidad de disco se efectúa mediante la correspondiente señalización de la unidad (normalmente A o B). Pulsar simplemente la tecla "A" o bien "B".

Si desea seleccionar una unidad de disco no formateable con el programa DISKTOOL (C = Floppy RAM o la placa fija), entonces el texto en banda de movimiento continuo reproduce el texto de error siguiente:

Error de lectura reintento S/N ?

Si sólo dispone de una unidad de disco (por ejemplo, F1-S) y selecciona entonces la unidad B, está claro que no dispone del mismo (otro motivo más para ampliar su F1-S y formar el F1-D). Entonces el programa DISKTOOL contesta con el siguiente texto en la cinta de movimiento continuo:

Error de lectura reintento S/N ?

Si la introducción es correcta, entonces pueden llegar otros dos submenús, según que esté conectada una unidad de disco de 3" (automáticamente en las unidades de disco X; en las estaciones S/D a través del cable adaptador VORTEX FDA-1). Para las unidades de 3" exis-

ten diferentes formatos. Si la señalización de la unidad introducida por usted coincide con la de la unidad de disco de 3", recibirá en la pantalla la siguiente información:

DISKTOOL 1.0 (C) 1986 vortex GmbH

Subrutina INICIALIZA DISQUETES

- 1 — Formateo sistema con CP/M
- 2 — Formateo sistema sin CP/M
- 3 — Formateo sólo datos
- Vuelta a pregunta anterior

Pulse la tecla correspondiente Pulse ...

Aquí podemos elegir entre tres formatos, porque precisamente estos están denominados también por el controlador de 3" DDI-1 Amstrad, y deseamos continuar siendo compatibles. Estos tres formatos tienen, sólo el sentido de ahorrar un par de kilobytes más de capacidad de memoria del disquete de 3", ya que ésta no es precisamente abundante (aproximadamente 170K). En cambio, los abundantes 700K de un disquete VORTEX son ya otra cosa; por eso aquí es suficiente también un formato.

El "ahorro de la capacidad de K" se consigue, por ejemplo, dejando libres las pistas del sistema, con lo que se pueden llenar de datos dos pistas más. Todavía un par de observaciones breves respecto a la configuración de estos formatos:

Formato del sistema con CP/M:

- 512 Bytes por sector
- 40 Pistas (0...39)
- 64 Registros de índice
- 9 Sectores por pista (41 HEX...49HEX)
- 2 Pistas reservadas, con CP/M.

Formato del sistema sin CP/M (denominado también formato “Vendor”):

- 512 Bytes por sector
- 40 Pistas (0...39)
- 64 Registros de índice
- 9 Sectores por pista (41 HEX...49HEX)
- 2 Pistas reservadas, pero sin CP/M.

Formato de sólo datos:

- 512 Bytes por sector
- 40 Pistas (0...39)
- 64 Registros de índice
- 9 Sectores por pista (C1HEX...C9HEX)
- No hay ninguna pista reservada.

Pulse el número correspondiente para el formato deseado y ya está. Entonces se producen las siguientes informaciones o avisos:

Verificar pistas (S/N) ?  
Iniciando pista 039

La pregunta “verificar . . .” debemos responderla con “S” (sí), a fin de que no se puedan “colar” errores de escritura.

Si ha elegido una unidad de disco VORTEX con la referencia de unidad introducida, recibirá en la pantalla la información siguiente:

DISKTOOL 1.0 (C) 1986 vortex GmbH

Subrutina INICIALIZA DISQUETES

- 1 — Inicializa y copia CP/M
- 2 — Sólo inicialización
- Vuelta a pregunta anterior

Pulse la tecla correspondiente Pulse . . .

Con "1" se copia directamente el CP/M en el disquete recién formateado. Si se dispone de la ampliación VORTEX de memoria RAM de 62K CP/M 2.2, se copiará también automáticamente el fichero o grupo de datos \$OSC.SYS.

Con "2" se realiza sólo el formateado.

El formato VORTEX es un formato estándar:

- 9 Sectores por pista (1...9)
- 512 Bytes por sector
- 80 Pistas por cada cara (0...79)
- 2 Caras
- 2 Pistas reservadas
- 128 Registros de índice.

Si ha puesto en marcha el formateado, recibirá los siguientes avisos:

Verificar pistas (S/N) ?

Inicializando pista 159

La pregunta "Verificar ..." deberá contestarse "S" (sí), a fin de que no se puedan "colar" errores de escritura.

Si pulsamos en el menú principal la tecla "2", cambia el contenido de la pantalla:

DISKTOOL 1.0 (C) 1986 vortex GmbH

Subrutina COPIA DISQUETE COMPLETO

Unidad fuente ?  
Unidad destino ?  
Inicializa disquete destino (S/N) ?  
Verificar disquete destino (S/N) ?  
Vuelta a CP/M con **[ESC]**

Pulse la tecla correspondiente Pulse ...

Con esta función, estamos en situación de copiar disquetes enteros. Al contrario de lo que ocurre al copiar ficheros sueltos o grupos individuales de datos, en los que se realiza la lectura y la escritura "orientadas por los ficheros", en esta operación se efectúa la lectura y la escritura pista por pista. Este proceso de copiado se denomina frecuentemente también "físico".

Tenga en cuenta que una operación de copiado completo presupone siempre que el disquete fuente y el disquete destino tengan el mismo formato. Por lo tanto, no está permitido copiar de 3" a 5.25" o al revés.

A las preguntas "unidad fuente ?" y "unidad destino ?", contestará con las correspondientes denominaciones de la unidad.

Si desea utilizar un disquete completamente nuevo (todavía no formateado), conteste con "S" (sí) a la pregunta de "inicializa ...". Si el disquete está ya formateado, conteste con "N" (no).

**Atención:** Si realiza la copia de la unidad de disco A a la unidad A con formateado y se olvida de cambiar el disquete fuente y el de destino, resultará destruido el disquete fuente.

La pregunta "verificar ..." debe contestarse con "S" (sí), ya que aquí se comprueba permanentemente si han sido transmitidos sin errores todos los datos (esto se denomina también "verificación").

Si pulsamos la tecla "3" en el menú principal, cambia el contenido de la pantalla en la forma siguiente:

DISKTOOL 1.0 (C) 1986 vortex GmbH

Subrutina COPIA PISTAS CP/M

Unidad fuente ?  
Unidad destino ?  
Vuelta a pregunta anterior

Pulse la tecla correspondiente Pulse ...

Al exponer la explicación de la orden **SYSGEN** del sistema operativo CP/M (en capítulos anteriores) hemos explicado ya la “configuración” de las pistas del sistema CP/M (pista 0 y pista 1). Recordará que con **SYSGEN** no se copiaban completamente las pistas 0 y 1, sino sólo una parte de las mismas, aunque sí la mayoría. Se trata en este caso de los sectores CP/M propiamente dichos. Como hemos visto, existe también un sector BOOT (que contiene el cargador del CP/M) y el sector de parámetros (contiene, por ejemplo, las tablas de correspondencia o traducción del teclado). Además tenemos allí todavía un par de sectores vacíos, los cuales están también “llenos” en el caso del sistema operativo CP/M de 62K junto con nuestra ampliación de memoria RAM.

Si se desean copiar las pistas completas del sistema, es decir, si se desea copiar un sistema operativo CP/M capaz de funcionar, necesitamos una función especial y ésta es precisamente la que aquí tenemos. La unidad fuente contiene el disquete, desde el que se debe leer el CP/M completo. La unidad de destino contiene el disquete, en el que se debe copiar este CP/M. Si sólo tenemos una unidad de disco, la banda de paso continuo nos indica cuándo deberemos cambiar los disquetes.

Tenga en cuenta, que esta función copia también el fichero “\$OSC.SYS” de un CP/M de 62K. En todo caso, después del copiado obtenemos un CP/M utilizable o cargable.

Pulsemos en el menú principal la tecla “4” y entonces cambia el contenido de la pantalla:

DISKTOOL 1.0 (C) 1986 vortex GmbH

Subrutina COPIA PISTAS CP/M

Unidad fuente ?  
Unidad destino ?  
Vuelta a pregunta anterior

Pulse la tecla correspondiente Pulse ...



Como ya hemos dicho anteriormente, el sector de parámetros es una parte de las pistas Boot de carga (más exactamente, el sector 2, pista 0). Contiene, por ejemplo, cambios de definición del teclado, cadenas de autoarranque . . .

El contenido de este sector puede ser modificado con la orden **INSTALL** del CP/M. Frecuentemente se desea también traspasar a otros disquetes un determinado “ajuste del CP/M”. Entonces no hace falta copiar todo el CP/M; es suficiente copiar el sector de parámetros.

El sector de parámetros a copiar es leído desde el disquete colocado en la unidad de disco fuente y es escrito en el disquete colocado en la unidad destino. Si sólo tenemos una unidad de disco, la banda de movimiento continuo nos indica cuándo deberemos cambiar el disquete.

## **La orden INSTALL del sistema operativo CP/M: Modificar la ocupación del teclado**

En el sector 2 de la pista 0 de un disquete VORTEX hay ciertos parámetros, que requieren, por ejemplo, otra ocupación diferente del teclado o incluyen constantes de tiempo del disco Floppy. Estos valores pueden ser modificados (instalados) con la orden **INSTALL**. Esta orden está controlada por el menú y subdividida en varios “niveles”. A continuación exponemos un ejemplo para cada nivel, para que comprenda la forma de trabajar con **INSTALL**. La orden **INSTALL** es una orden transitoria, es decir, que en el disquete debe existir ya el fichero **INSTALL.COM**.

**A)INSTALL** **ENTER**

---

**INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.**

**Sector parametro contiene 457 Bytes libres.**

**Cambio de**

- 1 - Constantes de tiempo Floppy disc**
- 2 - Configuración de teclas**
- 3 - Parametros de CP/M (I/O byte, power-up strings)**
- 4 - Comunicacion de parametros de interface**
- 5 - Instalación de parametros/ Arranque CP/M**

**Su seleccion: 1**

---

**INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.**

**Sector parametro contiene 458 Bytes libres.**

**Constantes de tiempo de Floppy disc**

- 1 - Motor on delay (1,2,...,99 secs) : 01 secs**
- 2 - Motor off delay (1,2,...,99 secs) : 03 secs**
- 3 - Steprate unidad 5.25" vortex (2,4,...,14 msec) : 04 msec**
- 4 - Steprate unidad 3" Amstrad (2,4,...,14 msec) : 12 msec**
- 5 - Menu maestro**

**Su seleccion: 1**

**Valor nuevo: 4**

---

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 458 Bytes libres.

Constantes de tiempo de Floppy disc

- 1 — Motor on delay (1,2,...,99 secs) : 01 secs
- 2 — Motor off delay (1,2,...,99 secs) : 03 secs
- 3 — Steprate unidad 5.25" vortex (2,4,...,14 msec) : 04 msec
- 4 — Steprate unidad 3" Amstrad (2,4,...,14 msec) : 12 msec
- 5 — Menu maestro

Su seleccion: 5

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 457 Bytes libres.

Cambio de

- 1 — Constantes de tiempo Floppy disc
- 2 — Configuración de teclas
- 3 — Parametros de CP/M (I/O byte, power-up strings)
- 4 — Comunicacion de parametros de interface
- 5 — Instalación de parametros/ Arranque CP/M

Su seleccion: 5

Instalar parametros nuevos S/N ? S

Arrancar de nuevo CP/M S/N ? S

---

44K CP/M vers. 2.2 - 03/86 vortex GmbH

A)

En este ejemplo hemos pasado del menú principal al submenú "constantes de tiempo del disco flexible Floppy" y hemos aumentado allí el tiempo de "retardo de conexión del motor" desde 1 hasta 4 segundos. Inmediatamente hemos instalado el nuevo parámetro (es decir, este parámetro ha sido escrito en el sector de parámetros) y se ha puesto en marcha de nuevo el CP/M.

Mediante la nueva puesta en marcha del sistema CP/M se garantiza el que el nuevo parámetro entre en vigor inmediatamente, ya que el contenido del sector de parámetros sólo es leído al cargar el CP/M.

El "retardo de conexión del motor" es el tiempo (retardo) que transcurre entre el arranque del motor de la unidad de disco y la operación de escritura/lectura.

El "retardo de desconexión del motor" es el tiempo que continúa todavía funcionando el motor, después de un acceso de escritura/lectura. Si este tiempo fuera de 0 segundos, se produciría un acceso extremadamente inseguro a la unidad de disco. Imagínese que tuviéramos un acceso periódico a la unidad de disco, por ejemplo, uno por segundo: el motor se estaría conectando y desconectando constantemente. Por el contrario, si ahora se ajusta un "retardo de desconexión del motor", que sea mayor de un segundo (por ejemplo, 3 segundos), el motor permanece conectado y entonces tenemos un acceso muy "suave", el cual en último término aumenta naturalmente la duración de vida de las unidades de disco.

El "índice de paso", denominado también tiempo de cambio de pista, es el tiempo que necesita la cabeza de lectura/escritura, accionada por un motor de pasos, para pasar de una pista a la pista que se encuentra inmediatamente junto a ella. Este tiempo no se puede modificar a voluntad. El valor mínimo está fijado por la "agilidad" máxima del motor de pasos; el valor máximo está fijado por las vibraciones de la cabeza de lectura/escritura, ya que al ir aumentando el índice de paso, se hace cada vez más "brusco" el funcionamiento de las cabezas.

Todas estas funciones han sido proyectadas primariamente para todos aquellos usuarios, que se han comprado por separado el controlador VORTEX de Floppys, o sea sin unidades de disco, y quizás desean utilizar otras unidades diferentes (por ejemplo, TEAC o bien, EPSON) a los utilizados por nosotros. (Quizás necesiten unos valores algo diferentes). Por lo demás, no deberán llegar nunca a una situación, en la que tengan que modificar algo aquí.

**A)INSTALL** **ENTER**

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 457 Bytes libres.

Cambio de

- 1 - Constantes de tiempo Floppy disc
- 2 - Configuración de teclas
- 3 - Parametros de CP/M (I/O byte, power-up strings)
- 4 - Comunicacion de parametros de interface
- 5 - Instalación de parametros/ Arranque CP/M

Su seleccion: 2

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 457 Bytes libres.

Instalaciones de teclas

- 1 - Tabla NORM esta vacia
- 2 - Tabla SHIFT esta vacia
- 3 - Tabla CTRL esta vacia
- 4 - Tabla tecla de funciones esta vacia
- 5 - Menú maestro

Su seleccion: 4

---

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.  
Sector parametro contiene 458 Bytes libres.

expansion Lista:

1=borrar todo 2=borrar apuntes 3=apuntes nuevos 4=menu

Su seleccion: 3

Valores permitidos: <80>...<9f>

Token: <80>

(max. 128 letras. ^H=recepc. valores HEX. <CR>=terminar.)

String: stat

(se debe almacenar 2 letras. ^C=conmutar a ASCII input)

Valores HEX: <0D>

Siga almacenando string:

---

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.  
Sector parametro contiene 451 Bytes libres.

expansi3n Lista:

Token String

<80> stat<0D>

1=borrar todo 2=borrar apuntes 3=apuntes nuevos 4=menu

Su seleccion: 4

---

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 451 Bytes libres.

Instalaciones de teclas

- 1 – Tabla NORM esta vacia
- 2 – Tabla SHIFT esta vacia
- 3 – Tabla CTRL esta vacia
- 4 – Tabla tecla de funciones esta llena
- 5 – Menú maestro

Su seleccion: 1

---

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 451 Bytes libres.

NORM Lista:

- 1=borrar todo 2=borrar apuntes 3=apuntes nuevos 4=menu

Su seleccion: 3

Numero de tecla (0,...,79): 64

Token: <80>

---

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 449 Bytes libres.

NORM Lista:

Numero de tecla      Token (<xy>=Valor HEX)

64                      <80>

- 1=borrar todo 2=borrar apuntes 3=apuntes nuevos 4=menu

Su seleccion: 4

---

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 449 Bytes libres.

Instalaciones de teclas

- 1 - Tabla NORM esta llena
- 2 - Tabla SHIFT esta vacia
- 3 - Tabla CTRL esta vacia
- 4 - Tabla tecla de funciones esta llena
- 5 - Menú maestro

Su seleccion: 5

---

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 449 Bytes libres.

Cambio de

- 1 - Constantes de tiempo Floppy disc
- 2 - Configuración de teclas
- 3 - Parametros de CP/M (I/O byte, power-up strings)
- 4 - Comunicacion de parametros de interface
- 5 - Instalación de parametros/ Arranque CP/M

Su seleccion: 5

Instalar parametros nuevos S/N ? S

Arrancar de nuevo CP/M S/N ? S

---

44K CP/M vers. 2.2 - 03/86 vortex GmbH

A)

En este ejemplo hemos hecho uso de la posibilidad de ocupar una tecla con una función (puesta en marcha del programa STAT).

Como hemos visto, a una tecla cualquiera se le puede asignar simplemente otro valor, además de la asignación de función. Por ejemplo,



en la tecla "Y" se puede colocar una "Z" mayúscula y al revés. Aquí aparece repetidas veces la palabra "símbolo" (Token). El símbolo o "Token" no es otra cosa que el signo que hemos asignado a una tecla determinada. Si queremos asignar a una tecla varios signos, o sea, una función, necesitamos los denominados "Expansion Tokens" (símbolos ampliados). El signo "80" en nuestro ejemplo es uno de estos símbolos o señales de ampliación.

La tecla 64 es, por lo demás, la tecla que está situada directamente a la derecha de la tecla [ESC]. Las teclas y sus correspondientes números puede verlas en su manual Amstrad para el ordenador CPC.

En nuestro ejemplo hemos dado otro significado a la tecla 64 sólo en el nivel NORM. si utilizamos la tecla en unión de la tecla [SHIFT] (nivel SHIFT), o con la tecla [CTRL] (nivel CTRL), dicha tecla tiene la misma función que anteriormente.

<OD> Tiene el mismo significado que si se pulsa la tecla [ENTER].

Si hemos salido ya del programa INSTALL y pulsamos luego la tecla 64, se pone en marcha el STAT.

Así pues, con ayuda de INSTALL se pueden ocupar en forma diferente todas las teclas (también las teclas de cursor y el bloque numérico). Con la posibilidad de los símbolos de ampliación (Expansión Tokens), se puede asignar a una tecla funciones enteras.

Tenga en cuenta que no se pueden realizar los cambios de ocupación en la cantidad que se desee. Como puede comprobar, con cada nueva ocupación, resulta cada vez más pequeño el espacio libre en el sector de parámetros.

La posibilidad de la ocupación libre del teclado es importante, por ejemplo, si desea adaptar a su ordenador CPC un programa como WordStar. En lugar de trabajar aquí de forma muy complicada con los códigos de control (por ejemplo, [CTRL] [S] mueve el cursor un signo hacia la izquierda), ocupamos simplemente las teclas con los códigos (por ejemplo, la tecla "cursor hacia la izquierda" con [CTRL] [S]).

**A)INSTALL**

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 449 Bytes libres.

Cambio de

- 1 - Constantes de tiempo Floppy disc
- 2 - Configuración de teclas
- 3 - Parametros de CP/M (I/O byte, power-up strings)
- 4 - Comunicacion de parametros de interface
- 5 - Instalación de parametros/ Arranque CP/M

Su seleccion: 3

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 449 Bytes libres.

Instalaciones CP/M

- 1 - I/O byte actual 81h
- 2 - Arranque automático string no existe
- 3 - Inicio impresora no existe
- 4 - Unidad 3" no activado
- 5 - Mensaje de errores BIOS activado
- 6 - Menu maestro

Su seleccion: 2

(max. 128 letras. ^H=conmutar a recepción valores HEX,  
^C=interrumpir sin alternación. <CR>=terminar string.)

String: install

(se debe almacenar 2 letras. ^C=conmutar a ASCII input)

Valores HEX: <0D>

Siga almacenando string:

---

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 441 Bytes libres.

Instalaciones CP/M

- 1 – I/O byte actual 81h
- 2 – Arranque automático string existe
- 3 – Inicio impresora no existe
- 4 – Unidad 3" no activado
- 5 – Mensaje de errores BIOS activado
- 6 – Menu maestro

Su seleccion: 6

---

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 441 Bytes libres.

Cambio de

- 1 – Constantes de tiempo Floppy disc
- 2 – Configuración de teclas
- 3 – Parametros de CP/M (I/O byte, power-up strings)
- 4 – Comunicacion de parametros de interface
- 5 – Instalación de parametros/ Arranque CP/M

Su seleccion: 5

Instalar parametros nuevos S/N ? S

Arrancar de nuevo CP/M S/N ? S

---

44K CP/M vers. 2.2 - 03/86 vortex GmbH

A)INSTALL

Cada vez que se pone en marcha el sistema operativo CP/M se debe comprobar si existe una cadena o String de autoarranque. Si existe, se realizan sucesivamente todas las órdenes incluidas en la misma.

Si necesita, por ejemplo, el sistema CP/M sólo porque desea trabajar con WordStar, puede entonces definir la cadena de autoarranque en la forma siguiente: **WS TEXT.TXT<0D>**.

Con ello se pone en marcha automáticamente el WordStar y se introduce el fichero o grupo de datos de texto TEXT.TXT. La indicación <0D> tiene el mismo significado que si se pulsa la tecla **ENTER**.

**A)INSTALL** **ENTER**

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 458 Bytes libres.

Cambio de

- 1 — Constantes de tiempo Floppy disc
- 2 — Configuración de teclas
- 3 — Parametros de CP/M (I/O byte, power-up strings)
- 4 — Comunicacion de parametros de interface
- 5 — Instalación de parametros/ Arranque CP/M

Su seleccion: 4

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 458 Bytes libres.

Comunicacion de parametros INTERFACE

- 1 — Rango de Baudios canal A: transmision 19200
- 2 — Rango de Baudios canal A: recepcion 19200
- 3 — Rango de Baudios canal B: transmisión/recepción 9600
- 4 — Canal A: 8 Data bits, 1 stop bit, Paridad: ninguna
- 5 — Canal B: 8 Data bits, 1 stop bit, Paridad: ninguna
- 6 — Menu maestro

Su seleccion: 1

Valores opcionales: 50, 75, 110, 150, 200, 300, 600, 1200, 1800, 2000,  
2400, 3600, 4800, 9600, 19200

Valor nuevo: 3600 **ENTER**

---

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 458 Bytes libres.

Comunicacion de parametros INTERFACE

- 1 — Rango de Baudios canal A: transmision 3600
- 2 — Rango de Baudios canal A: recepcion 19200
- 3 — Rango de Baudios canal B: transmisión/recepción 9600
- 4 — Canal A: 8 Data bits, 1 stop bit, Paridad: ninguna
- 5 — Canal B: 8 Data bits, 1 stop bit, Paridad: ninguna
- 6 — Menu maestro

Su seleccion: 6

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 457 Bytes libres.

Cambio de

- 1 — Constantes de tiempo Floppy disc
- 2 — Configuración de teclas
- 3 — Parametros de CP/M (I/O byte, power-up strings)
- 4 — Comunicacion de parametros de interface
- 5 — Instalación de parametros/ Arranque CP/M

Su seleccion: 5

Instalar parametros nuevos S/N ? S

Arrancar de nuevo CP/M S/N ? S

---

44K CP/M vers. 2.2 - 03/86 vortex GmbH

A)

Si tiene un puesto de interconexión VORTEX RS232C (puesto de interconexión en serie) en su ordenador CPC, entonces puede ajustar parámetros con este punto del menú, como por ejemplo el índice Baud.

Finalmente, vamos a ver ahora una aplicación muy especial de la función "cadena de autoarranque". Se trata de modificar los colores de la pantalla de imagen, al cargar el sistema CP/M.

Elegimos como color de fondo el "gris" y como color de escritura el "negro". Para los poseedores de un monitor en color, se obtiene un contraste muy bueno con este par de colores, sobretodo en el modo de 80 signos.

**A)INSTALL**

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 458 Bytes libres.

Cambio de

- 1 — Constantes de tiempo Floppy disc
- 2 — Configuración de teclas
- 3 — Parametros de CP/M (I/O byte, power-up strings)
- 4 — Comunicacion de parametros de interface
- 5 — Instalación de parametros/ Arranque CP/M

Su seleccion: 3

---

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 458 Bytes libres.

Instalaciones CP/M

- 1 - I/O byte actual 81h
- 2 - Arranque automático string no existe
- 3 - Inicio impresora no existe
- 4 - Unidad 3" no activado
- 5 - Mensaje de errores BIOS activado
- 6 - Menu maestro

Su seleccion: 2

(max. 128 letras. ^H=conmutar a recepción valores HEX,  
^C=interrumpir sin alternación. <CR>=terminar string.)

String:

(se debe almacenar 2 letras. ^C=conmutar a ASCII input)

Valores HEX: <1C> <40> <77> <77> <1C> <61> <40> <40> <1D>  
<77> <77> <0D>

Siga almacenando string:

---

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 446 Bytes libres.

Instalaciones CP/M

- 1 - I/O byte actual 81h
- 2 - Arranque automático string existe
- 3 - Inicio impresora no existe
- 4 - Unidad 3" no activado
- 5 - Mensaje de errores BIOS activado
- 6 - Menu maestro

Su seleccion: 6

---

---

INSTALL 1.0S - Instalaciones Power Up (C) 1986 vortex s.a.

Sector parametro contiene 446 Bytes libres.

Cambio de

- 1 - Constantes de tiempo Floppy disc
- 2 - Configuración de teclas
- 3 - Parametros de CP/M (I/O byte, power-up strings)
- 4 - Comunicacion de parametros de interface
- 5 - Instalación de parametros/ Arranque CP/M

Su seleccion: 5

Instalar parametros nuevos S/N ? S

Arrancar de nuevo CP/M S/N ? S

---

44K CP/M vers. 2.2 - 03/86 vortex GmbH

A>^\@ww^\a@@^\]ww^M  
?

A>

Inmediatamente despues de enviada la cadena anterior, se modifican los colores de la pantalla de imagen.

Tenga en cuenta que al introducir la cadena String en INSTALL, no hay que introducir simultáneamente los signos "<" y ">".

## **La orden CASCOPY del sistema operativo CP/M: Copiar entre disquete y cassette**

En el caso de un ordenador CPC 464 o bien de un CPC 664 y CPC 6128, con reproductor de datos, tiene, además del Floppy, un reproductor de datos, que no deberá olvidar de ningún modo, a pesar de que prácticamente no se ve nunca un CP/M unido a un reproductor de datos.



El reproductor de datos con sus cassettes musicales del tipo normal en el mercado es un medio de acumulación de memoria muy barato, que se puede utilizar perfectamente, por ejemplo, para proteger o asegurar (Back Up) datos que no sean demasiado importantes.

#### **A)cascopy**

CASCOPY 2.0S (C) 1986 vortex s.a.

Indique dirección para copias (<, >):

Disquete > Cassette

Nombre fichero de disquete: ed.com

Nombre fichero de cassette: e1.com

Se debe escribir con 2000 Baudios S/N ? S

Press REC and PLAY then any key: (pulse REC y PLAY y luego cualquier tecla).

Saving E1.COM block 1

En el ejemplo anterior se ve cómo podemos trasladar el fichero ED.COM del disquete al cassette (“>”). Al fichero de este cassette le hemos dado el nombre de E1.COM. Si en lugar de E1.COM hubiéramos introducido sólo la orden , el fichero habría recibido también en el cassette la denominación ED.COM.

La transferencia se realiza a alta velocidad (puede ir todavía más rápido). El aparato reproductor se conecta al estado de “grabación” pulsando simultáneamente las teclas  y , tal como estamos acostumbrados con los aparatos reproductores de música normales.

#### **A)cascopy**

CASCOPY 2.0S (C) 1986 vortex s.a.

Indique dirección para copias (<, >):

Disquete < Cassette

Nombre fichero de cassette: test

Nombre fichero de disquete: test

Press PLAY then any key: (pulse  y luego cualquier tecla).

Loading TEST block 1

Ahora el caso exactamente inverso: transferir un fichero del cassette al disquete. El fichero o grupo de datos recibirá en el disquete también el nombre TEST (ningún tipo de fichero).

Si hubiéramos pulsado simplemente la tecla **ENTER**, al preguntarnos "nombre del fichero del cassette", se habría transferido el siguiente fichero que se hubiera encontrado.

Si el fichero transferido existe ya en el disquete, se pregunta si debe ser borrado o no.

## **La orden VDOS del sistema operativo CP/M: Conexión de retorno al VDOS/BASIC**

Con esta orden del sistema operativo CP/M se abandona en forma adecuada dicho sistema operativo.

También existe la posibilidad del rearme o reposición: pulsando simultáneamente las teclas **CTRL**, **SHIFT** y **ESC**, se pone a cero su ordenador CPC.

**A>VDOS** **ENTER**

A continuación de esto se borra la pantalla de imagen y aparecen los avisos de conexión del ordenador CPC.

## **La orden FAST del sistema operativo CP/M: Aceleración de la imagen en la pantalla.**

Si ha trabajado ya en otros ordenadores con el sistema operativo CP/M, habrá comprobado seguramente que la reproducción del CP/M en la pantalla de su ordenador CPC no es precisamente la más rápida. La reproducción, a veces ya casi "desesperadamente" lenta, seguro que le ha molestado también.

La poca velocidad de reproducción no se debe a una mala programación del controlador de video, sino al hecho de que la memoria de imagen del ordenador CPC está orientada por puntos. Para las múltiples y variadas posibilidades gráficas que posee el ordenador CPC,

esto resulta indispensable. Además, se pueden ajustar 3 modos diferentes (20, 40 y 80 signos). Las partes del programa que “pasan” constantemente, debido a este sistema, hacen que sea lenta su reproducción en la pantalla.

Sin embargo, si se tiene en cuenta que en el sistema operativo CP/M en realidad sólo se necesita la reproducción de 80 signos, y que se puede renunciar a toda la “fantasía gráfica”, entonces la reproducción en la pantalla podría ser notablemente más rápida.

El pequeño programa FAST (rápido), que por lo demás es también una orden transitoria, aprovecha estas circunstancias y se encarga de una cierta aceleración (aproximadamente el factor 2).

Tenga en cuenta que el programa FAST, a pesar de estas limitaciones, sirve de apoyo a toda la “tecnología de ventana”. El programa FAST actúa como un conmutador.

**A>FAST**

FAST queda conectado.

**A>FAST**

FAST queda otra vez desconectado.

Tenga en cuenta que un “Reset” (reposición) desconecta también la “rutina FAST”.

## **Avisos de error del sistema operativo CP/M y su eventual corrección**

Los avisos alemanes son fabricados todos por BIOS. Por ello, puede ocurrir que un aviso de error alemán pueda ser seguido de un aviso inglés.

Tenga en cuenta que la reproducción de los avisos alemanes se puede suprimir con ayuda de la orden **INSTALL** (véase más adelante).

**Unidad de disco d: error de escritura.- Repetir, sí/no ?**

La operación de escritura, que afecta al disquete situado en la unidad de disco definida con "d:", ha sido interrumpida por un error. Si se contesta con "S" (sí), se vuelve a intentar de nuevo. Si se pulsa "N" (no), se interrumpe la operación.

**Unidad de disco d: error de lectura.- Repetir, sí/no ?**

Al leer los datos del disquete en la unidad de disco "d:" ha aparecido un error.

Si se contesta con "S" (sí) se intenta otra vez. Si se pulsa "N" (no), se interrumpe la operación.

**Unidad de disco d: falta el disquete.- Repetir, sí/no ?**

La unidad de disco elegida no contiene ningún disquete o bien no está correctamente cerrado. Coloque un disquete o bien pulse adecuadamente el botón e introduzca la respuesta "S" (sí).

**Unidad de disco d: el disquete tiene protección contra escritura.- Repetir, sí/no ?**

Usted desea escribir datos en un disquete, que está provisto de una tira adhesiva de protección contra escritura. Saque el disquete, quite la tira adhesiva y vuelva a colocarlo. Si pulsa "S" (sí), se puede repetir la operación. Si no desea escribir en este disquete, pulse simplemente "N" (no) y con ello interrumpirá la operación.

**No hay disquete del sistema CP/M.- Repetir, sí/no ?**

Usted quería poner en marcha el sistema operativo CP/M desde este disquete con la orden ¡CPM [ENTER]. Sin embargo, este disquete no pertenece al sistema. O bien, le falta el sector "Boot" de carga (pista 0, sector 1), o no posee sistema CP/M. Lo puede intentar eventualmente todavía con otros disquetes, intercambiándolo y contestando con "S" (sí).

### **Unidad de disco d: error al formatear.- Repetir, sí/no ?**

Se ha cometido un error al formatear este disquete. Quizás esté defectuoso el disquete; también puede probar otra vez, a ver si funciona.

### **Impresora no preparada.- Repetir, sí/no ?**

Usted desea reproducir algo en la impresora con **CTRL** **P** o tiene un programa que desea acceder a la impresora. Puede ocurrir que la impresora esté todavía desconectada o no seleccionada. ¿Ha conectado la impresora? Si comprueba uno de estos puntos, corrija el error y luego pulse "S" (sí).

### **Se repite la operación**

Cuando aparece la pregunta "-Repetir, sí/no ?", puede contestar con un "S" (sí); entonces, el sistema indica el acuse de recibo con "se repite la operación".

### **Se interrumpe la operación**

Si responde con "N" (no) a la pregunta "-Repetir, sí/no ?", el sistema indica el acuse de recibo con "se interrumpe la operación". Aquí puede ocurrir frecuentemente que, después de interrumpida la operación, se indique todavía un error BDOS (siempre estos molestos errores BDOS). Intente salvarlo con **CTRL** **C**.

### **Los indicativos permitidos para las unidades de disco son A o bien B**

**SYSGEN**. Si ha indicado aquí otra unidad de disco diferente a A o bien B, recibirá el anterior aviso de error.

### **El SYSGEN ha sido interrumpido por ^C**

**SYSGEN**. En alguna de las introducciones en el SYSGEN ha marcado **CTRL** **C** y con ello ha interrumpido el programa antes de tiempo.

### **El índice (directorio) está lleno**

**CASCOPY**. Con la operación de copiado que ha realizado ahora mismo (de cassette a disquete) en este disquete, quiere introducir más

de 128 registros en el índice de contenido del disquete. Borre los ficheros o grupos de datos inútiles y ponga en marcha otra vez el programa CASCOPY.

### **El disquete está lleno**

**CASCOPY.** Con esta operación de copiado (de cassette a disquete) se llena el disquete. Borre los ficheros o grupos de datos inútiles, o bien coloque un disquete nuevo.

### **Error. Cassette no conectado**

**CASCOPY.** Se ha producido un error en la transmisión de los datos.

### **¡Fichero protegido! No puede ser leído**

**CASCOPY.** Los ficheros o grupos de datos del cassette, protegidos contra copiado, no pueden ser transmitidos al disquete con el programa CASCOPY.

### **No se ha podido encontrar el grupo de datos (File) en el disquete**

**CASCOPY.** El fichero o grupo de datos, que había que trasladar del disquete al cassette, no se consigue encontrar en el disquete. Quizás se ha equivocado al hacer sus anotaciones, o ha colocado un disquete equivocado.

### **¡Error! Se interrumpe el programa CASCOPY**

**CASCOPY.** Al efectuarse la transmisión se ha producido un error fatal.

### **Nombre de fichero o grupo de datos: no válido**

**CASCOPY.** No es válida la denominación del fichero del grupo de datos que se desea transmitir. Quizás es demasiado larga o contiene signos no permitidos.

**El <FICHERO> ya existe aquí. Borrar, sí/no ?**

**CASCOPY.** Ya existe la denominación del fichero del grupo de datos, que se desea copiar en el cassette o en el disquete. Entonces se puede borrar (“sí”) o se puede dejar (“no”).

**No se ha podido cargar el sector de parámetros**

Al realizar la carga del sistema operativo CP/M se ha descubierto que está defectuoso el sector de parámetros. Compruebe este sector e instale uno nuevo, si es necesario.

**Sector de parámetros equivocado**

El contenido del sector de parámetros es “incomprensible” para el sistema operativo CP/M.

**La memoria intermedia de ampliación o expansión está llena o bien hay una característica o señal no permitida**

Hay errores en la tabla de modificación de la ocupación del teclado. Quizás ha utilizado un signo o característica no permitido.

**No se ha podido enviar la cadena de la impresora**

La impresora probablemente no está conectada o no está enchufada o no está seleccionada. Corrija la causa y cargue de nuevo el sistema operativo CP/M.

**Loader: unidad de disco de 3" = unidad B**

Esta indicación adicional de carga señala que la unidad de disco de 3" en el sistema CP/M es siempre la unidad B. Este ajuste se puede dejar fijo con **INSTALL**.

**Desbordamiento de la capacidad del sector de parámetros. No es posible la instalación (modificación). Interrumpir, sí/no ?**

**INSTALL.** Usted desea meter datos con más de 512 Bytes en el sector de parámetros. Eso es demasiado.

**Se interrumpe el programa INSTALL, ya que el sector de parámetros no es legible**

**INSTALL.** Al poner en marcha el programa INSTALL se lee el sector de parámetros. Pero, algo va mal en este sector, puesto que no se puede leer o no se lee bien.

**Atención:** El sector de parámetros no se ha podido escribir correctamente.

**INSTALL.** Al dejar el programa INSTALL, se escribe en el disquete el nuevo sector de parámetros. Sin embargo, esta vez lamentablemente no ha funcionado. Se ha producido algún error al escribir.

**Disquete no colocado.- Repetir, sí/no ?**

**DISKTOOL.** Usted ha olvidado colocar un disquete de destino o bien no ha bloqueado correctamente el botón de la unidad de disco. Corrija los errores e introduzca la respuesta "sí".

**El disquete tiene protección contra escritura.- Repetir, sí/no ?**

**DISKTOOL.** El disquete de destino colocado tiene una marca de protección contra escritura. Quite la marca, si es que está permitido, e introduzca la respuesta "sí".

**No hay unidad de disco**

**DISKTOOL.** Usted está intentando trabajar con el Floppy de memoria RAM o con el disco duro con una de las funciones DISKTOOL. Eso no puede ser. Con el programa DISKTOOL sólo se pueden manejar unidades de discos flexibles o disquetes.

**Los disquetes tienen formatos diferentes**

**DISKTOOL.** Usted ha intentado copiar un disquete VORTEX completo en un disquete de 3" o bien transmitir a un disquete de 3" las pistas del sistema de un disquete VORTEX. Estas funciones sólo trabajan, si se dispone de unidades de disco con el mismo formato.



### **Unidad de disco no disponible**

**DISKTOOL.** Usted ha intentado seleccionar una unidad de disco, que no está disponible (por ejemplo, la unidad D).

### **El disquete no tiene pistas reservadas del sistema**

**DISKTOOL.** Supongamos, que desea transferir pistas del sistema de un disquete de 3" a otro disquete de 3". Si este disquete tiene el formato de "sólo datos" (Data-Only), recibirá el aviso anterior.

### **El programa DISKTOOL 1.0 necesita vectores activos del sistema con las tarjetas de memoria RAM del sistema operativo CP/M**

**DISKTOOL.** Si desea trabajar con DISKTOOL también con el sistema operativo CP/M de 62K de las tarjetas de memoria RAM VORTEX, deberá conectar los vectores del sistema al "corregir errores" o "codificar" (Patch).

### **Unidad de disco d: El disquete tiene protección contra escritura.- Repetir, sí/no ?**

**FILECOPY.** El disquete de destino tiene una señal de protección contra escritura. Si la quita, siempre que esté permitido, podrá poner en marcha la operación de copiado con "sí".

### **El <FICHERO> ya existe. Reescribir, sí/no ?**

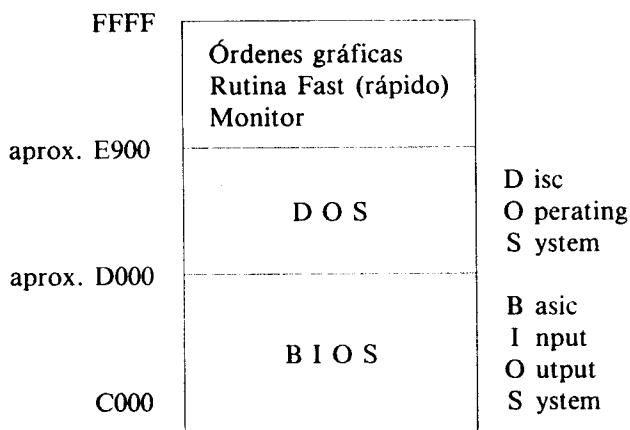
**FILECOPY.** el fichero o grupo de datos, que debe ser copiado precisamente en el disquete de destino bajo esta denominación de fichero, ya existe en dicho disquete. Si desea simplemente escribir encima, marque la respuesta "sí". En caso contrario, responde aquí con "no".

### **El programa FILECOPY 3.0 necesita vectores del sistema activos, con las tarjetas de memoria RAM del sistema operativo CP/M**

**FILECOPY.** Si desea trabajar con el programa FILECOPY también con el sistema operativo CP/M de 62K de las tarjetas de memoria RAM VORTEX, deberá conectar los vectores del sistema al "corregir errores" o "codificar" (Patch).

## Los sistemas operativos VDOS y CP/M internamente

El sistema operativo VDOS 2.0 ROM se puede subdividir, simplificando mucho, en las partes siguientes:



Los tres campos representan unidades más o menos independientes. Sobre todo el monitor y las órdenes gráficas se pueden delimitar claramente, mientras que entre los sistemas operativos DOS y BIOS existen relaciones muy estrechas.

### Descripción breve del funcionamiento:

## BIOS (Basic Input Output System, Sistema de Introducciones y Salidas BASIC)

Primero, para evitar confusiones, aquí la palabra "Basic" no tiene nada que ver con el lenguaje de programación BASIC. En este caso significa sólo "básico". El BIOS es en realidad la unión entre el controlador de Floppys (hardware, equipo) y su ordenador CPC (software, programas). Este sistema, a su vez se divide en dos bloques lógicos:

1. módulos para el mando de las unidades de disco;
2. módulos para el mando de la impresora, el teclado, la pantalla y, en general, para los aparatos periféricos.

Las subrutinas (módulos) del punto 2 no tienen propiamente nada que ver con las unidades de disco; de todos modos, se necesitan para la instalación de un sistema operativo CP/M. (Véase también más adelante el capítulo: "Examen detallado del CP/M").

El bloque 1 (mando de la unidad de disco) se puede dividir, por su parte, en otros 3 niveles jerárquicos consecutivos:

- 1.0 Rutinas de activación (de impulsión) de los discos flexibles Floppy.
- 1.1 Funciones básicas.
- 1.2 Gestión de la grabación (rutinas de bloqueo/desbloqueo).

Las rutinas de activación o de impulsión de los discos (Floppy) activan directamente el controlador de los discos a través de las correspondientes "entradas" (PORTS). Si se desea comprender su forma de funcionamiento y su estructura, habrá que ocuparse detalladamente de la construcción mecánica de los aparatos del controlador propiamente dicho y de las unidades de disco. Pero esto nos llevaría demasiado lejos en este manual, por lo que recomendamos, al que lo desee que lea los correspondientes folletos especiales y las hojas de datos.

Las funciones básicas son subrutinas, que utilizan, a su vez, las rutinas de activación de los discos flexibles y pueden ser utilizadas directamente por el usuario o por los programadores expertos en el lenguaje de la máquina. En el sistema BIOS del VDOS ROM están realizadas las siguientes funciones básicas:

- 1.10 Lee un sector físico
- 1.11 Escribe un sector físico
- 1.12 Formatea una pista completa
- 1.13 Posiciona el carro de cabeza
- 1.14 Comprueba el estado de la unidad de disco
- 1.15 Desconecta el motor de la unidad de disco.

Los valores a introducir se transmiten de acuerdo con los siguientes signos convencionales (las letras designan el registro interno del Z80):

- C = Número de sector
- D = Número de pista

E = Número de unidad de disco  
HL = Dirección de la memoria intermedia.

El señalizador de arrastre (Carry Flag) nos informa sobre el éxito o el fracaso de una operación de la unidad de disco:

- Arrastre realizado (Carry) significa que la operación se ha desarrollado con éxito;
- Arrastre = 0 (Carry = 0) nos indica que algo ha fallado.

El bloque mínimo de datos, que pueden gestionar tanto el sistema operativo CP/M como el DOS, no es un sector físico (512 Bytes), sino un registro denominado "Record" (128 Bytes). Cada sector físico puede comprender cuatro de estos registros o records. Un record es propiamente una denominación "histórica" de un valor y procede todavía de los primeros tiempos de los Floppys. Entonces, el valor o tamaño físico de un sector era también de 128 Bytes, es decir un "record". Con la creciente incorporación de circuitos integrados "chips" para los aparatos controladores, se consiguió rápidamente la posibilidad de leer y escribir también sectores de mayor tamaño (de 256, 512, ... Bytes), con lo que se pudieron reducir drásticamente los tiempos de acceso. Pero, para entonces, ya se habían popularizado determinados sistemas operativos (sobre todo el CP/M), por lo que el "record" era el formato de datos usual. Por ello, el sistema BIOS ha sido ampliado por rutinas de bloqueo/desbloqueo. Estas rutinas son unas subrutinas, que dividen el sector físico en forma controlada en "records" o bien lo componen a base de varios "records" y controlan adecuadamente el acceso. Esto se debe a que, con un tamaño físico de sector de 512 Bytes, nos vemos obligados, en el caso más favorable, a echar mano del disquete en realidad sólo cada cuatro "records".

Al conjunto de temas de la gestión de los "records" pertenecen las siguientes rutinas del sistema BIOS:

- 1.20 Coloca el número de pista
- 1.21 Coloca el número de record
- 1.22 Coloca la dirección (DMA) de la memoria intermedia de "record"
- 1.23 Lee el "record" en la memoria intermedia de "records"
- 1.24 Escribe el "record" en la memoria intermedia de "records"
- 1.25 Traduce el número de pista, en caso necesario.

Cada una de estas rutinas corresponde a una entrada en el denominado vector del sistema BIOS del CP/M. (Véase más adelante el capítulo “Examen detallado del CP/M”.)

El sistema operativo BIOS contiene todavía otras funciones adicionales, que detallaremos aquí sólo brevemente:

- SELDSK selecciona una determinada unidad de disco
- BOOT realiza una puesta en marcha en frío del CP/M
- WBOOT realiza una puesta en marcha en caliente del CP/M.

## **DOS (Disk Operating System, Sistema Operativo de Disco**

El “DOS” es el corazón, propiamente dicho, de la memoria ROM del sistema VDOS. Como ya hemos visto, el sistema BIOS sólo es capaz de gestionar datos del disquete en forma de “Records” o de sectores. Allí no se ha hablado para nada de ficheros o grupos de datos (Files). Por este motivo, se requieren determinados conocimientos básicos, para trabajar “sin destruir nada” con el sistema BIOS, mientras que, por el contrario, los accesos directos al sistema DOS se realizan generalmente “con los ojos cerrados”.

El sistema VDOS, exactamente igual que el CP/M, es un sistema operativo de disquetes orientado para ficheros o grupos de datos. En forma parecida a las tarjetas o fichas de un fichero, los datos no se gestionan como Bytes individuales (letras), sino como unidades lógicas superiores (grupos de datos). ¿Cómo vuelve a encontrar entonces el sistema DOS un determinado fichero o grupo de datos en el disquete? Si echamos todas las fichas en una caja y las revolvemos bien, es muy improbable que tengamos la suerte de encontrar rápidamente una ficha determinada. ¿Qué hacemos entonces? Ordenamos las fichas en un fichero y confeccionamos un índice de su contenido. Esto es exactamente lo que hace también el sistema DOS: Cada grupo de datos o fichero recibe un nombre para su señalización clara e inconfundible. Estos nombres se almacenan en un índice o directorio (el índice de contenido del disquete), junto con el lugar exacto del disquete, en el que se encuentra el grupo de datos. Esta explicación está,

desde luego, muy simplificada, pero corresponde exactamente a la realidad. Antes de tratar con más detalle sobre el sistema DOS, daremos todavía unas pocas explicaciones básicas sobre la estructura de los datos, que son indispensables para comprenderlo perfectamente.

Como usted seguramente ya sabe, o lo podrá comprobar rápidamente por sí mismo, en el sistema operativo CP/M se pueden utilizar también los mismo disquetes, que usted emplea en el BASIC. Esto no es una circunstancia natural, sino que se debe a que el sistema VDOS gestiona los grupos de datos del mismo modo que el CP/M. Esto le libra de la preocupación de separar cuidadosamente los "disquetes del sistema CP/M" y los disquetes del BASIC". Así no le puede salir nada mal. Las explicaciones siguientes sirven, por lo tanto, del mismo modo para los sistemas VDOS y CP/M:

**Record:** Bloque de datos con 128 Bytes. Es la cantidad mínima de datos utilizable en el intercambio de datos con el Floppy.

**Bloque:** El sistema DOS debe llevar, en cierto modo, una contabilidad de los datos existentes en un disquete. Esto se hace del modo siguiente: se distribuye el disquete en pistas, se cortan las pistas y se coloca una detrás de la otra. De este modo, se obtiene una cinta larga, en la que va un Byte detrás de otro; así obtenemos una configuración lineal del disquete. Ahora bien, una posibilidad de gestionar los datos sería el darle a cada Byte un número, de acuerdo con su situación en la cinta. En este caso, la gestión de los datos sería enormemente costosa. Con una capacidad de 708K, habría que registrar en el sistema DOS 724.992 números, lo cual no es realizable en la práctica. Por ello, se divide la cinta (el disquete) en grandes grupos, los bloques, y se les dan números a estos bloques. El tamaño de bloque en el Floppy VORTEX es de 4K (4096 Bytes). Como se ve fácilmente, en este caso (de división en bloques) se necesitan muchos menos números, concretamente unos 200. Independientemente del tamaño grande o pequeño de un grupo de datos, éste ocupa siempre al menos un bloque en el disquete. Lamentablemente, esto tiene un ligero inconveniente: se pierde forzosamente siempre algo de espacio de almacenamiento. Suponga que

tiene 20 ficheros o grupos de datos en el disquete, cada uno de los cuales contiene sólo 1K de datos. Sin embargo, cada grupo de datos ocupa un bloque, de modo que perdemos  $20 \times 3K = 60K$ , a pesar de que, en realidad, habríamos necesitado sólo  $20 \times 1K = 20K$ . Pero, hay que conformarse con ello. Una solución sería hacer más pequeños los bloques, pero esto tendría a su vez como consecuencia el que necesitaríamos más números para numerar los bloques. Por consiguiente, hay que aceptar algún compromiso.

Para poder comprender mejor la importancia del bloque, como estructura de datos en el sistema operativo DOS, estudiaremos ahora con mayor detalle el índice o directorio y su configuración:

El directorio (el índice de contenido del disquete) se divide en "entradas". Cada entrada (Entry) tiene exactamente una longitud de 32 Bytes y se configura en la forma siguiente:

- Byte 0 : Número del usuario
- Byte 1-8 : Nombre del fichero o grupo de datos
- Byte 9-11 : Extensión (ampliación)
- Byte 12 : Número de extensión
- Byte 13,14 : Variables de trabajo del sistema DOS
- Byte 15 : Record Count = cuenta de records (cantidad de "records" en la extensión que tenemos en este momento)
- Byte 16-31 : Tabla de ocupación de bloques.

Ahora nos fijamos en el Byte 16-31, la tabla de ocupación de bloques. Aquí tenemos por orden sucesivo los números de los bloques, que ocupa el grupo de datos. Supongamos que deseamos encontrar en el disquete un determinado grupo de datos. El sistema DOS actúa en la forma siguiente: busca un registro en el índice o directorio, que contenga el nombre del grupo de datos buscado. Si encuentra dicho registro, podrá saber inmediatamente, mediante la tabla de ocupación de bloques, dónde está este grupo de datos. Hasta aquí todo es muy sencillo. Como podemos calcular fácilmente, en la tabla de ocupación de bloques de un registro

del índice o directorio tenemos sitio para un máximo de 16 números de bloques. Con un tamaño de bloque de 4K sólo podemos colocar claramente en el disquete, por lo tanto, un máximo de un fichero o grupo de datos de una longitud de 64K. ¿Qué hacer entonces, si el grupo de datos tiene 65K o más? Aquí entra en juego entonces la última estructura de datos de nuestra explicación: la extensión (Extent).

**Extent:** La "Extent" (extensión o ampliación) caracteriza un bloque de datos del tamaño de 16K. ¿Por qué precisamente 16K? Esto se debe también a una cuestión "histórica": en las estaciones de disquetes de pequeña capacidad es conveniente elegir bloques pequeños, a fin de perder el mínimo espacio posible. El tamaño mínimo posible de bloque es 1K (todos los ordenadores CPC con unidad de disco de 3" "funcionan" con bloques de 1K). Con este tamaño de bloque podemos alojar exactamente 16 números de bloques en un registro del índice o directorio, es decir, el registro puede alojar en este caso exactamente datos de 16K. Si un grupo de datos es más grande, el sistema operativo DOS confecciona simplemente más registros (Extents = ampliaciones) con el mismo nombre del grupo de datos y anota en el Byte 12 el número de la extensión en la que se encuentran. Si el sistema DOS gestiona grupos de datos en varios bloques grandes, no cambia prácticamente nada en esta unidad, sino que únicamente entran más extensiones en un registro del directorio (bloques de 4K → 4 Extensiones).

Así pues, ahora ya sabemos lo que es un "record", un bloque y una extensión y hemos aprendido que los ficheros o grupos de datos se depositan por bloques en el disquete. Todavía daremos una corta explicación sobre otro registro muy importante en una entrada del índice o directorio: el "Record Count" (véase arriba en el Byte 15). Este registro tiene la misión de fijar el fin exacto de un grupo de datos y, por lo tanto, su tamaño (en "records") exacto. Un ejemplo: tamaño de bloque 4K; grupo de datos con una longitud de 8 records (1K). Este grupo de datos ocupa exactamente un bloque, pero en realidad sólo necesita la primera cuarta parte de dicho bloque. ¿Cómo puede saber entonces el sistema DOS dónde termina el grupo de datos dentro



del bloque? Precisamente para esto sirve el "Record Count": cuenta los "records" dentro de una extensión (por lo tanto, va desde 0 hasta 127, puesto que en una extensión entran exactamente 128 records) y de este modo permite determinar el fin absoluto de un grupo de datos, independientemente del tamaño del bloque. Además, el sistema DOS permite comprobar, por medio del "Record Count" si hay todavía otra extensión a continuación o no la hay (si la cuenta de records "Record Count" es 127, viene a continuación siempre otra extensión).

Ahora llegamos ya al objetivo propiamente dicho de este capítulo: la forma de funcionamiento del sistema operativo DOS.

El sistema DOS, que es más o menos una biblioteca de programas, se divide, en forma muy general, en dos grupos de módulos:

- 2.0 Rutinas para el índice o directorio orientado por ficheros y la manipulación de los datos;
- 2.1 Módulos de adaptación.

Ambos grupos están parcialmente relacionados entre sí en forma jerárquica. Las rutinas del segundo grupo (a las que hemos denominado módulos de adaptación, por el motivo que veremos pronto) se refieren a rutinas procedentes del primer grupo y éstas a su vez acceden al disquete a través del sistema operativo BIOS.

En primer lugar, veamos el primer grupo y su subdivisión:

- 2.01 Abre el fichero o grupo de datos
- 2.02 Cierra el fichero
- 2.03 Escribe secuencialmente el "record" en un fichero
- 2.04 Lee secuencialmente el "record" desde el fichero
- 2.05 Escribe directamente el "record" en el fichero
- 2.06 Lee directamente el "record" desde el fichero
- 2.07 Borra el fichero o grupo de datos
- 2.08 Pone un nuevo nombre al fichero
- 2.09 Reproduce en la pantalla el índice de contenido
- 2.10 Manipula los atributos del fichero o grupo de datos.

Vemos que el grupo 1 de las funciones del sistema DOS proporciona operaciones básicas para la manipulación del fichero/directorio.

El sistema VDOS, al contrario que el AMSDOS, permite dos clases de acceso al disquete: el acceso secuencial y el acceso directo. Con el acceso secuencial, sólo se puede leer/escribir un grupo de datos siempre por orden sucesivo. El acceso directo o también relativo permite leer y escribir por el orden sucesivo que se desee cualquier tipo de "records" del fichero.

El grupo 1 continúa estando en el centro del sistema DOS, es decir, no se han previsto entradas estandarizadas en estas rutinas. No obstante, si desea utilizar estas rutinas, sólo podrá alcanzarlas si entra directamente en la memoria ROM. Sin embargo, al hacerlo tenga en cuenta imprescindiblemente que las entradas no estandarizadas pueden variar de una revisión a otra de la memoria ROM. Por consiguiente, lo más prudente sería evitar tales entradas, sobre todo si se confeccionan programas de software comerciales.

El grupo 2 los módulos de adaptación poseen todos ellos entradas estandarizadas, al contrario de lo que ocurre en el grupo 1, y por lo tanto, se pueden alcanzar siempre a través de estas entradas, independientemente de las revisiones. Ahora veamos el concepto: módulo de adaptación. Para ello tenemos que repasar todavía algo:

El BASIC del ordenador CPC (1.0 y 1.1) mantiene una memoria externa de almacenamiento masivo, a través de un número propio denominado de "flujo = Stream" (véase el manual de BASIC). En el ordenador modelo 464 esto es la unidad de disco incorporada; en los modelos 664/6128 es la unidad de disco de 3" incorporada. El BASIC no se dirige directamente a esta memoria de almacenamiento masivo, sino que utiliza para ello el denominado "Main Jump Vector". Este vector es una regleta de "solicitudes" moduladas estandarizadas, que se encuentra en la memoria de almacenamiento a partir de la dirección BB00H, es independiente de la versión o modelo y, por lo tanto, es igual en todos los ordenadores CPC. Al decir aquí "igual" no queremos decir de ningún modo que las correspondientes regletas de salto o entrada tengan que coincidir entre sí Byte por Byte, sino que más bien significa que cada entrada tiene unos parámetros definidos de entrada y de salida y cumple una determinada función. Esto se denomina también "comportamiento Black Box". Un ejemplo: supongamos que desea reproducir la letra A en la pantalla. Un vista-

zo al manual de microprogramación en memoria muerta (Firmware) del ordenador CPC nos dice que para ello la función adecuada es la número 30, denominada TXT OUTPUT. Su entrada se encuentra en BB5AH. Al realizarse la entrada, en el acumulador (registro del Z80) debe estar el signo a reproducir (en este caso la "A", ASCII código 41H). No existe ninguna condición para la entrada. Por lo tanto, carece de importancia absolutamente, qué rutina se esconde concretamente detrás de la entrada BB5AH; lo importante es sólo el efecto que causa, con qué datos tiene que ser alimentada y qué es lo que da como respuesta (estado, condición de entrada). Esto se denomina también punto de interconexión (del programa o software). (No se debe confundir con los puntos de interconexión de los aparatos del Hardware, por ejemplo, Centronics, RS232C, ...)

Pondremos todavía otro ejemplo tomado de la vida diaria: el buzón para las cartas. Un buzón es, en cierto modo, también un punto de interconexión con unas condiciones bien definidas de entrada y de salida. Su objeto es transportar las cartas desde el lugar "x" hasta el lugar "y". Lo que ocurre luego concretamente con una carta, después de que usted la ha echado en el buzón, es cuestión secundaria (a no ser que se pierda la carta), hasta que llega a su destino. Así pues, para dar un significado concreto a estos conceptos: la carta es la magnitud de entrada (véase el signo en el acumulador en TXT OUTPUT); el transporte desde "x" hasta "y" es la tarea a realizar, es decir la función del punto de interconexión (véase "reproducir signos en la pantalla"). No existe condición de salida, mientras todo marche bien. Si la carta sufre algún daño durante su recorrido por los circuitos postales, la oficina de Correos se lo comunicará; ésta sería entonces la correspondiente condición de salida, por decirlo así, el "aviso de estado".

Después de esta pequeña excursión al campo del correo, volvamos otra vez a nuestra memoria de almacenamiento masivo en el ordenador CPC.

El BASIC se comunica con esta memoria de almacenamiento masivo a través de 13 entradas en el "Main Jump Vector", las funciones 125 a 137 (véase el manual de microprogramación en memoria muerta "Firmware" del ordenador CPC). Para disponer de la información

completa, enumeraremos a continuación estas funciones. Muy importante: los nombres de las funciones están tomados de acuerdo con la nomenclatura del citado manual y, por lo tanto, comienzan siempre con CAS . . . Esto no quiere decir que con dichas funciones nos refiramos sólo a la unidad de disco: nos podemos referir a cualquier medio de almacenamiento masivo de datos. Para ello, únicamente hay que sustituir las entradas por otras (doblar el vector), que nos conduzcan hasta las rutinas, con las mismas condiciones de entrada y de salida y con las mismas misiones, es decir, por ejemplo, una unidad de disco de Floppys.

Antes de entrar en el tema, un detalle más, molesto pero muy importante, que seguramente aumentará todavía más la confusión. En la descripción de las funciones se hablará frecuentemente de un "bloque". Los bloques, a los que se refieren las funciones siguientes, no tienen nada que ver con los bloques de la organización de los disquetes descritos anteriormente. Aquí se trata de porciones de datos del tamaño de 2K, debido a que el BASIC los necesita precisamente de esta forma.

#### **Función 125: Entrada: BC77H Nombre: CAS IN OPEN**

- Misión:** abre un fichero o grupo de datos en el canal de lectura y lee su primer bloque (atención, tenga cuidado de no confundirse).
- Entrada:** HL = indicador en el nombre del fichero  
B = longitud del nombre del fichero  
DE = indicador en la memoria intermedia del bloque de 2K.
- Salida:** – el fichero o grupo de datos se ha podido abrir correctamente:  
Carry=1 Z=0  
HL = indicador del File Header  
DE = dirección de destino de los datos  
BC = longitud del fichero  
a = tipo de fichero  
– el canal de lectura se estaba ya utilizando:  
Carry=0 Z=0

**Función 126: Entrada: BC7AH Nombre: CAS IN CLOSE**

Misión: cierra correctamente el grupo de datos en el canal de lectura.

Entrada: ninguna.

Salida: – se ha podido cerrar el grupo de datos:  
Carry=1  
– no se estaba utilizando ningún grupo de datos:  
Carry=0

**Función 127: Entrada: BC7DH Nombre: CAS IN ABANDON**

Misión: cierra inmediatamente el canal de lectura.

Entrada: ninguna.

Salida: ninguna.

**Función 128: Entrada: BC80H Nombre: CAS IN CHAR**

Misión: toma el siguiente signo del canal de lectura, procedente del grupo de datos que esté abierto allí en este momento.

Entrada: ninguna.

Salida: – se ha podido recoger correctamente el signo:  
Carry=1 Z=0 Signo en el acumulador  
– se ha detectado el final del fichero o grupo de datos:  
Carry=0 Z=0

**Función 129: Entrada: BC83H Nombre: CAS IN DIRECT**

Misión: lee el grupo de datos abierto en este momento en el canal de lectura, en una secuencia en el almacenador de datos. Las funciones 129 y 128 se excluyen mutuamente entre sí. Por lo tanto, no se pueden sacar primero individualmente los 5 primeros signos y luego recoger el resto de una vez.

Entrada: HL = dirección de destino, donde se deben depositar los datos.

- Salida:**
- el fichero se ha podido leer y depositar correctamente:  
Carry=1 Z=0 HL=entrada en el campo de datos (sólo es adecuada, en caso de que se trate de un programa de máquina)
  - el grupo de datos no estaba abierto:  
Carry=0 Z=0

**Función 130: Entrada: BC86H Nombre: CAS RETURN**

**Misión:** devuelve otra vez al canal de lectura el último signo leído con CAS IN CHAR.

**Entrada:** ninguna.

**Salida:** ninguna.

**Función 131: Entrada: BC89H Nombre: CAS TEST OEF**

**Misión:** comprueba si se ha alcanzado el final del grupo de datos.

**Entrada:** ninguna.

- Salida:**
- no se ha alcanzado todavía el final del grupo de datos:  
Carry=1 Z=0
  - se ha alcanzado el final del grupo de datos:  
Carry=0 Z=0

**Función 132: Entrada: BC8CH Nombre: CAS OUT OPEN**

**Misión:** abre un grupo de datos en el canal de escritura.

**Entrada:** HL = indicador en el nombre del grupo de datos  
B = longitud del nombre del grupo de datos  
DE = indicador en la memoria intermedia del bloque de 2K.

- Salida:**
- no se ha podido abrir correctamente el grupo de datos:  
Carry=1 Z=0 HL=indicador en File Header.
  - el canal de escritura se está utilizando ya:  
Carry=0 Z=0.

**Función 133: Entrada: BC8FH Nombre: CAS OUT CLOSE**

Misión: cierra correctamente el canal de escritura.

Entrada: ninguna.

Salida: – se ha podido cerrar correctamente el grupo de datos:  
Carry=1 Z=0  
– no se estaba utilizando ningún grupo de datos:  
Carry=0 Z=0.

**Función 134: Entrada: BC92H Nombre: CAS OUT ABANDON**

Misión: cierra inmediatamente el canal de escritura.

Entrada: ninguna.

Salida: ninguna.

**Función 135: Entrada: BC95H Nombre: CAS OUT CHAR**

Misión: envía un signo al canal de escritura.

Entrada: el acumulador contiene el signo.

Salida: – se ha podido entregar el signo:  
Carry=1 Z=0  
– no estaba abierto el grupo de datos de escritura:  
Carry=0 Z=0

**Función 136: Entrada: BC98H Nombre: CAS OUT DIRECT**

Misión: envía al canal de escritura un campo coherente de almacenamiento de datos. Las funciones 134 y 135 se excluyen mutuamente entre sí.

Entrada: HL = indicador en el campo del almacenador  
DE = longitud del campo del almacenador  
BC = dirección de entrada (sólo utilizable en programas de máquina)  
A = tipo de grupo de datos

Salida: – se ha podido escribir correctamente:  
Carry=1 Z=0  
– no estaba abierto para escritura ningún grupo de datos:  
Carry=1 Z=0

### **Función 137: Entrada: BC9BH Nombre: CAS CATALOG**

**Misión:** sitúa en la pantalla de imagen el índice de contenido. Esta función utiliza el canal de lectura y, por lo tanto, no se puede utilizar, si éste está ocupado ya en este momento para otra cosa.

**Entrada:** DE = indicador en la memoria intermedia de trabajo de 2K.

**Salida:**

- todo en orden:  
Carry=1 Z=0
- el canal de lectura se estaba ya utilizando:  
Carry=0 Z=0

Si, al conectar el ordenador CPC, se encuentra un Floppy en el sistema, se desvía hacia éste la parte del "Main Jump Vector", que guía en dirección al reproductor de datos el punto de interconexión de la memoria de almacenamiento masivo, si se está en el llamado ROM WALK (en este caso, el sistema operativo va buscando en todas las memorias ROM de fondo, de modo que comprueba qué aparatos periféricos están disponibles). Para ello, es suficiente con sustituir las entradas en las rutinas de la unidad de disco por las correspondientes entradas en las rutinas del Floppy (módulos de adaptación).

## **Variables de memoria RAM de los sistemas BIOS y DOS**

Tanto el sistema operativo BIOS, como el DOS necesitan para la gestión de datos también la memoria RAM, a pesar de que las rutinas se encuentran en la ROM. Esta RAM se divide en dos grupos:

### **3.0 Memoria RAM No reubicable**

#### **3.1 Memoria RAM reubicable**

Como ya lo dice el propio nombre, el campo de memoria RAM no reubicable no es desplazable, es decir, está dirigido absolutamente hacia una dirección por el sistema operativo DOS/BIOS. Por el contrario, el campo de la RAM reubicable está dirigido hacia una direc-



ción sólo de modo indicativo y, por lo tanto, se puede desplazar libremente. El motivo de que este campo de la RAM tenga que ser desplazable debe buscarse en la configuración de la microprogramación en memoria muerta (Firmware): toda ROM (de fondo) de la periferia puede reservar para sí memoria RAM durante su ROM WALK. Esto sólo no tendría demasiada importancia, porque toda ROM de fondo posee siempre un número de ROM. El recorrido ROM WALK se realiza desde el número 7 → 0 en el ordenador 464 y desde el 15 → 0 en los ordenadores 664/6128. El número de ROM es una cuestión de aparatos de hardware y está fijado por el equipo electrónico situado alrededor de la ROM. Así pues, la reserva de RAM se realiza de la forma siguiente (aquí nos referimos sólo al ordenador 464; para los ordenadores 664 y 6128 se procede en forma similar): primeramente el número 7 de la ROM se reserva su sitio en la RAM; a continuación, el número 6 de la ROM se puede reservar sitio directamente debajo; esto quiere decir, sin embargo, que una ROM, antes de su recorrido propiamente dicho ROM WALK, no puede saber dónde estará su zona de memoria RAM en el almacenador, puesto que no sabe cuánto sitio le reservan las memorias ROM situadas “encima” del mismo. Por ello, la ROM da, en forma indicativa, la referencia de todas sus variables en el campo desplazable de la RAM, recibiendo del núcleo 1 la dirección básica de su campo de memoria RAM (en el registro IY), y añadiéndose el correspondiente “Offset”. Si se escribe un programa, que deba ser “lanzado” más tarde a una ROM (de fondo), teniendo en cuenta rigurosamente la dirección indicada por las variables de la RAM, la ROM podrá ser metida más tarde en una versión opcional de ROM y trabajará correctamente a la primera. Todo este asunto tiene un sólo inconveniente, que generalmente no causa perturbaciones: la dirección indicada es más complicada que la absoluta y esto, a su vez, hace que el programa resulte algo “más lento” (generalmente no puede medirse) y algo más largo (esto ya suele producir molestias, dado que en la ROM sólo existe espacio limitado). Dado que toda la configuración del almacén de datos (en el BASIC) del ordenador CPC está orientado hacia esta reserva dinámica de espacio, sólo se dispone de forma muy limitada de RAM “absoluta”. (La mayor parte de la RAM “absoluta” resulta ocupada por el interpretador del BASIC y por el sistema operativo). Uno de los pocos campos de RAM “absolutos” grandes y coherentes se

encuentra en BE80H (hasta el "Stack") y es utilizado también por la microprogramación en memoria muerta (Firmware) del Floppy.

**Muy importante:** A continuación encontrará campos de memoria RAM ocupados por memoria ROM del sistema operativo VDOS 2.0, clasificados por su orden de importancia. Naturalmente, tenemos cuidado de no modificar estos campos durante las revisiones; pero a veces esto resulta inevitable. Por ello, tenga cuidado antes de "enredar" según su capricho y voluntad (el "inspeccionar" está naturalmente permitido).

**A) Campo de memoria RAM no reubicable, de la memoria ROM del sistema operativo VDOS 2.0:**

(Todas las direcciones son hexadecimales).

**Importante:** Un \* en la segunda columna significa que en el mismo lugar se puede encontrar la misma variable en el sistema AMSDOS. La cifra de la tercera columna indica la longitud de las variables en Bytes.

BE40 \* 2 Dirección DPH, unidad de disco A.

BE40 \* 2 Dirección DPB, unidad de disco B.

BE44 \* 2 Motor on counter (contador de marcha del motor): tiempo en unidades de 50 mseg, después de la conexión del motor, hasta el primer acceso. Debe ser mantenido, a fin de disponer de una velocidad constante de rotación del disquete.

Valor de defecto (omisión): 50

BE46 \* 2 Motor off counter (contador de parada del motor): tiempo en unidades de 50 mseg desde el último acceso hasta la desconexión del motor. Si se desconectara el motor inmediatamente después del acceso y tuviéramos que realizar un nuevo acceso un poco más tarde, habría que esperar otra vez a que el motor estuviera en fase, es decir, que se haría excesivamente lento el acceso.

Valor de defecto (omisión): 250

- BE49 \* 1 Tiempo de reposo de la cabeza, en unidades de 1 mseg. Después de que el carro de la cabeza entra en una pista, continua vibrando mecánicamente todavía durante cierto tiempo. Hay que esperar a que se "tranquilice" el carro, para evitar errores de escritura/lectura.  
Valor de defecto (omisión): 30
- BE4A \* 1 Índice combinado de pasos para unidad de disco de 5.25"/3.5" y 3". Como sabe, el sistema operativo BIOS de VORTEX puede activar unidades de disco de 5.25"/3.5", así como también la unidad de disco de 3" Amstrad. Aparte de su capacidad, estos tipos de unidades de disco se diferencian también por sus muy diferentes índices de paso. El carro de la cabeza no se mueve continuamente, sino en pasos individuales (Steps). Según cuál sea la unidad de disco activada, se debe conmutar, por lo tanto, el índice de paso. Los 4 bits superiores son el índice de paso de la unidad de disco de 3" en unidades de 1 mseg y los 4 bits inferiores son el índice de paso de la de 5.25"/3.5" en unidades de 1 mseg.  
Valor de defecto (omisión): 196
- BE4B 8 Memoria intermedia resultante para uPD 765 (véase la hoja de datos técnicos).
- BE53 \* 1 Seek Disk number = número de búsqueda de disco (número lógico de la unidad de disco; se pone por medio de la rutina SELDSK del sistema BIOS).
- BE54 \* 1 Seek Track number = número de búsqueda de pista (número lógico de pista; se pone mediante la rutina SELTRK del sistema BIOS).
- BE55 \* 1 Seek Sector number = número de búsqueda de sector (número lógico de sector = número de "Record"; se pone mediante la rutina SELSEK del sistema BIOS).
- BE56 \* 1 Host disk number = número primario de disco (número de la unidad de disco física; se obtiene en el sistema BIOS).
- BE57 \* 1 Host Track number = número primario de pista (número físico de pista; se obtiene en el sistema BIOS).

BE58	*	1	Host Sektor number = número primario de sector (número físico de sector; se obtiene en el sistema BIOS).
BE59		1	Seek/Host Flag = señalizador de búsqueda/primario: los diferentes Bits tienen el siguiente significado: Bit 0-3: tipo de escritura 0=con destino 1=directorio    2=sin destino Bit 4 : 0=escribir 1=leer Bit 5 : 1=pre-lectura Bit 6 : señalización primaria activa Bit 7 : señalización primaria de escritura Los Bits 5-7 se necesitan para el bloqueo/desbloqueo
BE5A		1	Contador de "record" sin destino
BE5B		1	Número de unidad lectora sin destino
BE5C		1	Número de pista sin destino
BE5D		1	Número de sector sin destino (lógico)

Observación a estas cuatro últimas variables: el sistema operativo BIOS tiene fundamentalmente 3 posibilidades de escribir un "record" o registro:

1. Escribir en el directorio o índice.
2. Escribir con destino.
3. Escribir sin destino.

Como ya sabemos, en un sistema BIOS "bloqueado" no es imprescindible realizar inmediatamente la operación física de escritura, con cada "Record Write". Sin embargo aquí hay una excepción muy importante, que son las operaciones de escritura en el índice o directorio. Naturalmente, el directorio debe estar constantemente actualizado con los últimos datos, por lo que todo registro de directorio a escribir debe ser pasado también inmediatamente al disquete (sin bloqueo).

Antes de que pueda ser leído o escrito un registro, se deben transmitir al sistema BIOS las "coordenadas del registro", por medio de las órdenes **SELDISK**, **SELTRK**, **SELSEK** y **SETDMA**; un acceso de este tipo se denomina también "sin destino", ya que el bloque en el que se debe escribir o del que se debe leer el

registro se define completamente. Pero, también existe otra posibilidad de acceso del registro, que es cuando se han traspasado ya al sistema BIOS (con destino) las coordenadas del bloque y entonces se deben leer/escribir todos los registros siguientes del bloque. Este acceso es naturalmente más rápido que el que se realiza en un nuevo bloque (sin destino).

- BE5E 1 Seek/Recalibrate/ID-Flag = señalizador de búsqueda/recalibrado/ID: aquí se almacena en la memoria intermedia el código de orden de mando uPD, al recibir una orden de mando de lectura de búsqueda, recalibrado o ID.
- BE5F 1 Almacenador intermedio para el índice de paso de la unidad de disco seleccionada en el momento.
- BE60 \* 2 Dirección de la memoria intermedia existente en el momento; 128 Bytes (puesta mediante la rutina SETDMA del sistema BIOS).
- BE62 \* 2 Dirección de la memoria intermedia primaria (512 Bytes). Memoria intermedia del sector para un sector físico.
- BE64 1 Retry Count = cuenta de los intentos realizados: cantidad de intentos en caso de error.  
Valor por defecto (omisión): 16
- BE65 2 Almacenador intermedio de datos "Stack" al entrar el sistema BIOS.
- BE67 \* 6 Tick Block para el contador de conexión/desconexión del motor.
- BE6D \* 7 Event Block para el contador de conexión/desconexión del motor.
- BE74 1 Almacenador intermedio para la orden de mando del uDP.
- BE75 2 Aquí se almacena la dirección básica del campo reubicable de la memoria RAM del sistema DOS.  
Valor por defecto (omisión): A700H, si la memoria ROM del sistema VDOS es el número 7.
- BE77 2 Si se quiere acceder a una rutina del sistema DOS en la memoria ROM, se puede depositar aquí su dirección.  
Valor de defecto (omisión): 0

- BE79 2 Cadena de código para la protección P (se pone mediante la orden de mando CODE del RSX).
- BE7B 1 Message Flag = señalizador de mensaje: 0 = se indican los avisos de error. FFH = se suprimen los avisos de error.
- BE7C 1 En el sistema BASIC: señalizador EOF: 0 = prueba con el marcador EOF. FFH = no hay prueba. Con el sistema CP/M: señalizador FAST para la reproducción en la pantalla: 0 = velocidad normal. FFH = reproducción acelerada.
- BE7D 2 No tiene ningún significado en el sistema BASIC. En el sistema CP/M: dirección básica del campo reubicable de la memoria RAM.
- BE7F 1 Sólo con la unidad de discos flexibles X en el sistema BASIC: Byte de configuración.  
0 = la unidad de disco de 5.25" es la unidad A.  
1 = la unidad de disco de 5.25" es la unidad B.  
(Se pone con la orden de mando !X de RSX.)

Las siguientes variables son muy diferentes en los sistemas BASIC y CP/M y, por ello, se tratan por separado:

### 1. BASIC:

- BE80 2 Almacenador intermedio para indicador de paquetes de datos "Stackpointer".
- BE82 1 Almacenador intermedio para código de error "Error Code"
- BE83-BE8E Sin significado.
- BE8F 3 Select Disk (seleccionar disco) indirección DOS (actúa la rutina SELDSK del sistema BIOS).
- BE92 3 Set Track (poner pista) indirección DOS (actúa la rutina SETTRK del sistema BIOS).
- BE95 3 Set Sector (poner sector) indirección DOS (actúa la rutina SETSEK del sistema BIOS).
- BE98 3 Set DMA indirección DOS (actúa la rutina SETDMA del sistema BIOS).
- BE9B 3 Read Record (leer registro) indirección DOS (actúa la rutina READ del sistema BIOS).

- BE9E        3    Write Record (escribir registro) indirección DOS (actúa la rutina WRITE del sistema BIOS).
- BEA1        3    Posición principal indirección DOS.  
Entrada: acumulador = número lógico de pista.  
Salida: acumulador = número físico de pista.
- BEA2-BEE9    Rutina para corrección de un error.
- BEBA        2    Dirección básica para gestión relativa de grupo de datos.
- BEBF        1    Número del primer sector con disco I/O.
- BEC0        1    Número del último sector con disco I/O.

Durante el recorrido ROM WALKS hasta el momento, en el que el sistema BASIC ha alcanzado su bucle READY (preparado), las siguientes variables tienen otro significado en el ordenador CPC 464:

- BE80        6    Tick Block = bloque de excitación para cargador Hello.
- BE86        7    Event Block = bloque de suceso para cargador Hello.

Ahora un par de palabras respecto al significado de las indirecciones DOS y sobre la forma de funcionamiento de la secuencia Hello.

Las indirecciones DOS representan un punto de interconexión bien definido entre los sistemas DOS y BIOS. El sistema DOS accede al sistema BIOS, es decir, a las unidades de disco, únicamente a través de estas indirecciones. La descripción de los parámetros de transmisión la encontrará en la parte relativa al sistema operativo CP/M, en las explicaciones sobre el vector de entrada del sistema BIOS.

La secuencia Hello, el "mecanismo de giro de llave y en marcha", del que sólo disfrutaban los poseedores del ordenador 464, se basa en la idea de detectar el BASIC después del recorrido ROM WALK, antes de que entre en el bucle READY (preparado). La corrección de preparado "Ready Patch" sólo es posible en el ordenador 464/BASIC 1.0). De todos modos, no tiene ningún sentido el corregir la indirección READY del BASIC ya durante el recorrido ROM WALK, puesto que el BASIC, una vez que ha realizado el recorrido por la memoria ROM (ROM WALK), comienza de nuevo la iniciación (C9H) de todas las indirecciones BASIC y, por lo tanto, borra-

ría la corrección. Por eso, la rutina ROM WALK en la memoria ROM del sistema VDOS coloca la indirección READY en un estado definido de comienzo (00) y produce luego un activador, que comprueba constantemente la iniciación C9H de todas las direcciones READY del BASIC, mientras que continúa en marcha el recorrido por la memoria ROM (ROM WALK). Si se encuentra un C9H, se realiza la corrección del READY y el cargador Hello lo copia en la memoria RAM a continuación de BF00H. Si el BASIC activa entonces la indirección READY, el programa se desvía hacia el cargador HELLO a continuación de BF00H. Esto hace que se borre el activador, coloca la indirección READY en C9H, comprueba si se ha pulsado una tecla e interrumpe el trabajo, si había ocurrido así. Si no había ocurrido ésto, comprueba el estado de la unidad A. Si no hay ningún disquete en la unidad, se interrumpe también. Si hay un disquete, el cargador busca el grupo de datos HELLO.BAS. Cuando lo ha encontrado, lo carga en la memoria del BASIC y se pone en marcha.

## 2. CP/M:

- BE80 \* 3 Message Flag = señalizador de mensaje: 0 = comienzo de avisos. 0FFH = fin de avisos.
- Entrada: acumulador = valor nuevo.
  - Salida: acumulador = valor antiguo.
- BE83 \* 3 Inicializar las constantes de tiempo.
- HL = indicador en el campo de parámetros.
- BE86 \* 3 - Entrada: acumulador = 0FFH. E = número de unidad de disco (0,1)
- Salida: HL = indicador en DPB Flag Byte.
  - Entrada: acumulador = 0FEH Pone o borra el señalizador FAST (rápido)
  - Entrada: acumulador = 1 unidad de disco A = 3"; B = 5.25"
  - acumulador = 2 unidad de disco A = 5.25"; B = 3"
  - acumulador = 0 unidad de disco A = B = 5.25"



BE89 \* 3 Lee un sector físico.

- Entrada: HL = indicador en la memoria intermedia del sector
  - C = número del primer sector a leer
  - B = número del último sector a leer, en caso de que esté puesto el señalizador de multisector
  - E = número de unidad de disco
  - D = número de pista
- Salida: Carry = 1/acumulador = 0, en caso de que salga bien.  
Carry = 0/acumulador = código de error, en caso de error.

BE8C \* 3 Escribe un sector físico.

- Entrada: HL = indicador en la memoria intermedia del sector
  - C = número del primer sector a leer
  - B = número del último sector a leer, en caso de que esté puesto el señalizador de multisector
  - E = número de unidad de disco
  - D = número de pista
- Salida: Carry = 1/acumulador = 0, en caso de que salga bien.  
Carry = 0/acumulador = código de error, en caso de error.

BE8F \* 3 Formatea una pista entera.

- Entrada: HL = indicador en la memoria intermedia del sector
  - E = número de unidad de disco
  - D = número de pista
- Salida: Carry = 1/acumulador = 0, en caso de que esté bien.  
Carry = 0/acumulador = código de error, en caso de error.

- BE92 \* 3 Posiciona el carro de la cabeza.  
 - Entrada: E = número de unidad de disco.  
           D = número de pista.  
 - Salida: Carry = 1/acumulador = 0, en caso de  
                   que esté bien.  
               Carry = 0/acumulador = código de  
                   error, en caso de error.
- BE95 \* 3 Comprueba el estado de la unidad de disco.  
 - Entrada: acumulador = número de unidad de disco.  
 - Salida: Carry = 1, luego acumulador = ST 3 (véase  
                   la hoja de datos del uPD)  
               Carry = 0, luego HL = indicador en la  
                   memoria intermedia del resultado  
                   del uPD.
- BE98 \* 3 Modifica la cuenta de nuevo intento "Retry Count":  
 - Entrada: acumulador = nuevo valor  
 - Salida: acumulador = valor antiguo
- BE9B \* 3 BIOS Entry. Asegura el paquete de datos y se encarga de  
 la configuración correcta de la memoria ROM.
- BE9E 3 Desconecta el motor inmediatamente.
- BEA1 3 Position Head Indirection
- BEA4 \* 3 Inicializa la memoria intermedia de la consola.  
 - Entrada: HL = indicador en la cadena  
               A = señalizador de borrado "Clear Flag"
- BEA7 3 Introducción del estado SIO 1
- BEAA 3 Introducción del estado SIO 2
- BEAD 3 Salida del estado SIO 1
- BEB0 3 Salida del estado SIO 2
- BEB3 3 Trae el signo del SIO 1
- BEB6 3 Trae el signo del SIO 2
- BEB9 3 Envía el signo del SIO 1
- BEBC 3 Envía el signo del SIO 2

Los impulsores o activadores SIO no están disponibles todavía en el sistema operativo BIOS, de modo que las rutinas BEA7-BEBC llevan a rutinas simuladas "Dummy" (para escribir) o proporcionar un marcador EOF; 1AH (para leer).

### **Variables del monitor de disquetes:**

- BEC1 1 Número actual de unidad de disco (según el mando L).
- BEC2 1 Número actual de pista (según el mando T).
- BEC3 1 Número actual de sector (según el mando S).
- BEC4 2 Dirección actual de la memoria intermedia (según el mando B).
- BEC6 1 Variable interna.
- BEC7 1 Variable interna.
- BEC8 1 Número de la memoria superior ROM (según el mando Y).
- BEC9 1 0 = microprogramación en memoria muerta (Firmware)  
ROM desconectada:  
FF = conectado (según el mando X).

### **Variables del monitor:**

- BECD 3 Dirección a distancia para indirección FAST (rápido).
- BED0 3 Dirección a distancia para el punto de interrupción.
- BED3 2 Dirección actual para el mando M.
- BED5 2 Dirección actual para el mando L.
- BED7 2 Dirección actual para el mando F.
- BED9 2 Dirección actual para el mando D.
- BEDD 1 Señalizador del CP/M 0 = VDOS activo FF = CP/M activo.
- BEDE 2 Indicador de apilado de registro en la entrada al monitor.
- BEE0 1 Variable interna.
- BEE1 1 Variable interna.
- BEE2 1 "Opcode" almacenador intermedio.
- BEE3 2 Almacenador intermedio para derivaciones del programa.
- BEE5 2 Almacenador intermedio para el indicador de apilado de registro.
- BEE7 2 Almacenador intermedio para HL.
- BEE9 2 Cantidad de pasos de rastreo.
- BEEB 2 Almacenador intermedio para AF.
- BEEC 2 Usuario AF.
- BEEE 2 Usuario BC.

BEF0	2	Usuario DE.
BEF2	2	Usuario HL.
BEF4	2	Usuario IX.
BEF6	2	Usuario IY.
BEF8	2	Usuario SP.
BEFA	1	Variable interna.
BEFB	2	Usuario PC.
BEFD	1	Distancia con órdenes relativas del Z80.
BEFE	1	Señalizador interno.
BEFF	1	Señalizador interno.
BF00	24	Punto de interrupción campo de trabajo.
BF80		Paquete del monitor.

**B) Campo reubicable de la memoria RAM de la memoria ROM del sistema VDOS 2.0:**

El campo reubicable de la memoria RAM de la memoria ROM del sistema VDOS 2.0 se diferencia mucho del campo de la ROM del sistema AMSDOS. Todas las direcciones indicadas son relativas y se refieren a la base del campo de memoria RAM. Si la ROM se encuentra en un marco con el número más alto posible de ROM de fondo, esta base es la dirección A700H.

0	1	Señalizador "Login" unidad de disco A 0 = unidad de disco todavía no anunciada. FFH= unidad de disco anunciada.
1	1	Señalizador "Login" unidad de disco B; véase arriba.
2	1	Variable interna.
3	2	Dirección DPH de la unidad de disco de defecto (omisión).
5	1	Número momentáneo del usuario.
6	1	Número momentáneo de la unidad de disco.
7	2	Indicador en el FCB momentáneo (FCB = File Control Block = bloque de control del fichero).
9	1	Contador de filas para el mando CAT/DIR.
A	2	Contador del índice o directorio.
C-12	7	Variables internas.
13	1	Cantidad de signos para comparación de entrada del directorio de FCB.

14	1	Número de unidad de disco, procedente del FCB actual.
15	1	Señalizador 2 del DOS: Bit 1 = Read/Write random 0 = read 1 = write
16-18	3	Variables internas.
19	1	Señalizador 1 del DOS: Bit 1: 0 = random no activo 1 = random activo Bit 2: 0 = write 1 = read Bit 3: 0 = random 1 = secuencial
1A,1B	2	Variables internas.
1C	2	Almacenador intermedio para dirección DMA.
1E	1	Almacenador para cantidad de entradas del CAT.
1F	2	Indicador en la memoria intermedia de CAT RAM.
21,22	2	Variables internas.
23	2	Almacenador para indicador de apilado de registro para el caso de error.
24-25	32	FCB (bloque de control de fichero) de trabajo.
45-68	36	Input FCB.
69-8C	36	Output FCB.
92-B2	33	Input Header (cabecera).
B3-B3	33	Output Header.
E0-15F	128	Record Buffer (memoria intermedia de grabación).
164-178	21	Vector CAS Input.
179-187	15	Vector CAS Output.
188	3	Vector CAS CAT.
18B	3	Dirección lejana para el manipulador de entrada del sistema DOS.
190	23	ALV 0
1A7	23	ALV 1
1BE	128	Memoria intermedia de directorio.
23E	512	Memoria intermedia primaria.
43E	16	DPH 0
44E	16	DPH 1
45E	39	CSV 0
485	39	CSV 1
4AC	25	DPB 0
4C5	25	DPB 1

Observaciones: Los bloques de control de fichero (FCB) de entrada y de salida tienen la estructura estándar del sistema CP/M. (Véase “Examen detallado del sistema CP/M ...”.)

Las cabeceras tienen la siguiente configuración (todas las direcciones se refieren al comienzo de la cabecera):

0	1	Byte de modo: 1 = modo CHAR 2 = modo DIRECTO.
1	2	Dirección de la memoria intermedia del bloque.
3	2	Indicador en el signo siguiente en la memoria intermedia del bloque.
5	1	Byte de usuario.
6	8	Nombre del fichero o grupo de datos.
E	3	Extensión (ampliación).
11	4	No utilizado.
15	1	Número del bloque actual.
16	1	00 = no es todavía el último bloque.
17	1	Tipo de fichero.
18	2	Longitud del grupo de datos.
1A	2	Dirección de ubicación de datos.
1C	1	00 = no es el primer bloque.
1D	2	Longitud.
1F	2	Dirección de puesta en marcha del programa de la máquina.
21	3	Contador de Byte para el sistema DOS.
24	2	Suma de comprobación.

### C. Gestión relativa del fichero o grupo de datos:

El sistema operativo VDOS 2.0 puede gestionar simultáneamente 16 ficheros o grupos de datos relativos (16 canales). La longitud de campo en cada canal es opcional y está sólo limitada por el almacenador disponible. La manipulación del HIMEM del sistema BASIC se realiza por medio de la orden de mando FILES del RSX.

La memoria RAM del BASIC tiene la siguiente configuración en la gestión relativa activa de grupos de datos:

HIMEM' = HIMEM antes de la primera solicitud de la orden de mando FILES (ficheros). Están activos  $N(<16)$  ficheros relativos.

HIMEN'

Campo de trabajo  
del canal N ...

·  
·

Campo de trabajo  
del primer canal

---

Memoria interme-  
dia variable de  
305 Bytes

HIMEN

Los diferentes campos de trabajo están relacionados entre sí mediante direcciones de enlace; por ello, el sistema operativo DOS tiene que recordar sólo la dirección básica del campo general total. Aquí el campo de trabajo de un canal tiene la siguiente estructura. Todas las direcciones se refieren a la dirección de base del campo de trabajo del canal:

- |    |    |  |
|----|----|--|
| 0  | 1  | Número de canal;<br>los números posibles están entre 0 y 127, ambos inclusive, Si se pone el Bit 7, el canal está activo en ese momento. |
| 1  | 2  | Dirección de enlace con el siguiente campo de canal.   |
| 3  | 2  | Longitud máxima de campo; ha sido definida en FILES.   |
| 5  | 2  | Longitud actual del campo; ha sido definida en OPEN.   |
| 7  | 2  | Número actual de grabación (record).   |
| 9  | 2  | Indicador en la memoria intermedia de grabación.   |
| B  | 2  | Sin utilizar.  |
| D  | 36 | FCB (bloque de control del fichero).   |
| 31 | 32 | Máscaras del campo (máximo 16).  |
| 51 | 9  | Sin utilizar.  |
| 5A | ?  | Memoria intermedia de grabación (longitud definida en FILES).  |

### **Estructura del campo de las constantes de tiempo:**

0	2	Contador de motor en marcha.
2	2	Contador de motor desconectado.
4	1	No utilizado.
5	1	Tiempo de reposo (tranquilización) de la cabeza.
6	1	Índice combinado de paso.
7	1	HUT Byte (véase la hoja de datos del uPD).
8	1	HLT Byte (véase la hoja de datos del uPD).

### **Examen detallado del CP/M: el sistema operativo CP/M**

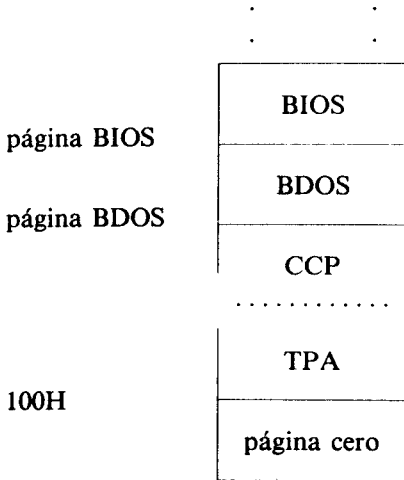
El sistema CP/M, igual que el BASIC del ordenador CPC, es un sistema operativo autónomo. Consta de tres componentes:

- el procesador de mandos de consola  
Console Command Processor = CPC
- el sistema operativo básico de disco  
Basic Disk Operating System = BDOS
- el sistema básico de entradas y salidas  
Basic Input Output System = BIOS.

El sistema operativo CP/M fue desarrollado por la empresa DRI (Digital Research Inc.) y solamente dicha empresa tiene el derecho exclusivo para conceder licencias. Por lo tanto, si reproduce uno de sus propios programas en un disquete, deberá tener cuidado de que no existan en el mismo pistas del sistema CP/M; de lo contrario cometerá una infracción o lesión del derecho de reproducción "Copyright".



Antes de entrar en detalle sobre las diversas partes del sistema CP/M, indicaremos aquí en primer lugar la distribución de la memoria de almacenamiento:



**Página cero:**

La página cero (Zero page), dirección 00-FFH, es necesaria como campo de datos y de trabajo para el sistema operativo CP/M. Prácticamente cada Byte tiene un significado, por lo que una manipulación atolondrada produce generalmente el derrumbamiento del sistema CP/M.

Dirección	Significado
00H-02H	Puesta en marcha en caliente; aquí hay un salto o entrada después de la página BIOS + 3.
03H	I/O Byte.
04H	Unidad de disco de defecto (omisión).
05H-07H	Entrada en el BDOS; aquí hay una entrada después de la página BDOS + 6.
08H-27H	Repetición de la puesta en marcha 1-5.
30H-37H	Repetición de la puesta en marcha 6.
38H-3AH	Repetición de la puesta en marcha 7.

- 3BH-5BH De momento no utilizado; reservado.
- 5CH-7CH Defecto (omisión) FCB (bloque de control de fichero).
- 7DH-7FH Número aleatorio "ramdom" (opcional).
- 80H-FFH Defecto (omisión) memoria intermedia de grabación.

Para el usuario es aquí especialmente interesante, por ejemplo, el "defecto/omisión FCB" en la dirección 5CH. Supongamos que solicita un fichero de mando (extensión COM), introduciendo la orden:

**GRUPO DE DATOS** nombre de fichero.ext 

entonces el procesador CCP coloca automáticamente un bloque de control de ficheros FCB para el grupo de datos "nombre de fichero.extensión". Esto le quita a usted el trabajo.

**TPA:**

La TPA (Transient Program Area = área de programa transitorio) es la zona "útil" propiamente dicha del sistema operativo CP/M. En la TPA se desarrolla, por ejemplo, su aparato WordStar. La TPA comienza exactamente en la dirección 100H y va como máximo hasta la página BDOS, es decir, el procesador CCP puede ser cargado desde un programa del usuario, sin perder el entorno del sistema CP/M. El procesador CPC (y el sistema operativo BDOS) se vuelve a cargar en la siguiente puesta en marcha en caliente. Un programa del usuario puede cargar también, en determinadas circunstancias, el sistema operativo BDOS, pero no debe acceder entonces a ninguna otra función del BDOS (en caso contrario se desintegra el sistema). Sin embargo, esto parece ser más bien un caso especial, que no se presentará prácticamente nunca.

**CCP:**

El CCP (Console Command Processor = procesador de mando de consola) es, por decirlo así, su "interlocutor", cuando trabaja con el sistema operativo CP/M. El CCP puede realizar órdenes de mando, que introduce usted manteniendo una sintaxis muy rígida. Además de la cualidad más importante, la de cargar programas en la zona de programa transitorio TPA y poderlos poner en marcha allí. el CCP contiene también las denominadas órdenes residentes, como por ejemplo **DIR**, **SAVE**, **TYPE** o bien **REN**. Precisamente esta rígida sintaxis de mando, es decir, el poco confort de manejo del mismo, ha hecho que el sistema operativo CP/M fuera durante mucho tiem-

po sólo algo para “especialistas” y no haya alcanzado la popularidad de los sistemas operativos BASIC. Pero comprobará también que el sistema CP/M, una vez que se alcanza un dominio mínimo del mismo, es una “herramienta” extremadamente elegante y polivalente.

#### **BDOS:**

El BDOS (Basic Disk Operating System = sistema básico operativo de disco) es el corazón propiamente dicho del sistema CP/M. Como el propio nombre indica, por su función equivale al sistema operativo DOS en el VDOS 2.0 ROM. Las diversas funciones en el sistema BDOS se exponen más abajo en forma de tabla. Aquí recomendamos que se lea abundante bibliografía sobre este tema; existen entre tanto libros muy buenos sobre el sistema CP/M (también para el principiante).

El procesador CCP y el sistema operativo BDOS se encuentran en las llamadas pistas del sistema de un disquete del CP/M y se cargan cada vez de nuevo en el almacenador, durante la puesta en marcha en caliente.

#### **BIOS:**

El BIOS (Basic Input Output System = sistema básico de carga y de salida) es el elemento de unión entre los aparatos del hardware y el sistema operativo BDOS. En el caso de una implementación estándar del sistema CP/M, el BIOS se encuentra en el disquete, igual que el BDOS y el procesador CCP, y es cargado una vez al efectuarse la puesta en marcha en frío. (Rutina de carga). En las implementaciones del sistema CP/M en el ordenador CPC se utiliza el sistema BIOS en una memoria ROM (VDOS o bien AMSDOS), debido a su mayor eficiencia, puesto que sólo de este modo tan sencillo puede acceder al Floppy el sistema DOS. El sistema BIOS es también una biblioteca de programas, que se divide, de forma muy general, en dos grupos lógicos:

- Input Output con no disco
- Input Output con disco.

El primer grupo permite al sistema operativo CPO/M comunicarse con el mundo exterior ( en este caso es usted), mientras que el segundo grupo gestiona sus datos.

## La memoria RAM en el sistema operativo CP/M:

Puesto que el campo no reubicable de la memoria RAM es aprovechado en gran parte por el sistema BIOS, también se aprovecha del mismo modo con el sistema CP/M. El sistema CP/M utiliza adicionalmente las siguientes variables:

(todas las direcciones son hexadecimales).

ACF8	23	ALV 0
AD0F	23	ALV 1
AD26	128	Memoria intermedia de índice
ADA6	512	Memoria intermedia primaria
AFA6	16	DPH 0
AFB6	16	DPH 1
AFC6	39	CVS 0
AFED	39	CVS 1
B014	25	DPB 0
B02D	25	DPB 1
B046	3	Manipulador de entrada del BIOS
B049	2	Entrada interrupción original
B04B	2	Estado de "sistema de puerta"
B04D	1	Señalizador de cursor
B04E	1	Señalizador de memoria intermedia de consola
B04F	2	Indicador de memoria intermedia de consola
B051	1	Contador de memoria intermedia de consola
B052	128	Memoria intermedia de consola
B0D2	2	Almacenador intermedio para paquetes de datos.

## Estructura del bloque de parámetros de discos (DPB):

El bloque de parámetros de discos de un Floppy VORTEX consta de dos partes: la parte estándar del sistema CP/M y una ampliación específica del sistema VDOS. Significado de los diversos registros: (estructura de la tabla: valor de defecto u omisión, cantidad de Bytes, denominación abreviada, explicación).

(Todos los valores son decimales).

Parte estándar:

36	2	SPT	Cantidad de grabaciones por pista.
5	1	BSH	Factor de cambio de bloque.
31	1	BLM	Máscara de bloque.
3	1	EXM	Máscara de extensión.
176	2	DSM	Cantidad total de bloques 1.
127	2	DRM	Cantidad total de registros de directorio 1.
128	2	AL0,1	Dos primeros Bytes del ALV.
32	2	CKS	Longitud del CSV.
2	2	OFF	Cantidad de pistas del sistema.

Ampliación específica VORTEX

1	1	MASK	Máscara del sector = número del primer sector.
9	1	ST	Cantidad de sectores físicos por pista.
42	1	GP0	Longitud del GAP0 (escribir/leer).
82	1	GP1	Longitud del GAP1 (formatear).
229	1	FILL	Byte de llenado para mando de formato.
2	1	N	Bytes/sector (→ 512).
4	1	BLK	Factor de bloqueo; 4 grabaciones por sector.
79	1	ATRK	Número físico de pista actual.
79	1	MTRK	Cantidad máxima de pistas por página -1.
1	1	FLAG	Byte de señalizador.

Significado de los diversos Bits en el Byte de señalizador:

**Bit Función**

– sólo el señalizador de la unidad A:

- 1 Señalizador de carga: 0 = mensaje de carga conectado.  
1 = mensaje de carga desconectado.
- 2 No utilizado  
(en el sistema VDOS 1.0: señalizador de mensaje).
- 3 Señalizador de motor: 0 = motor desconectado  
1 = motor conectado.

– ambas unidades de discos:

- 0 Cantidad de cabezas 1.

- 4 Señalizador de multisección: 0 = un sector I/O  
1 = multisección I/O.
- 5 Unidad de disco tipo: 0 = 5.25"/3.5"  
1 = 3".
- 6 Señalizador de recalibrado:  
0 = recalibrar el carro de la cabeza antes del siguiente posicionado  
1 = no hay recalibrado.

### **Estructura de la cabecera de parámetros de disco, Disk Parameter Headers (DPH):**

Disposición de la tabla: valor de defecto (omisión), cantidad de Bytes, denominación breve, explicación.

Todos los valores son decimales.

- 0            2    Variable de trabajo (dirección de la transmisión del sector).
- 0            2    Variable de trabajo (dirección de la transmisión del sector).
- 0            2    Variable de trabajo (dirección de la transmisión del sector).
- 0            2    Variable de trabajo (dirección de la transmisión del sector).
- DIRBUFF    2    Dirección de la memoria intermedia de índice o directorio.
- DPB            2    Dirección del bloque de parámetros de disco.
- CSV            2    Dirección del "Check Size Vektors" = comprobación tamaño vector.
- ALV            2    Dirección del "Allocation Vektors" = ubicación del vector.

## Disposición de un bloque de control de ficheros, File Control Block (FCB):

El sistema operativo CP/M, o bien respectivamente su sistema BDOS, sólo pueden acceder a ficheros o grupos de datos, si a éstos se les ha asignado previamente un bloque de control de ficheros (FCB). El bloque FCB contiene todos los datos sobre un fichero, que necesita el sistema BDOS para poder acceder claramente al grupo de datos. El bloque FCB estándar tiene una longitud de 33 Bytes, mientras que el bloque FCB aleatorio (random) tiene una longitud de 36 Bytes. A continuación exponemos el significado de los diversos Bytes por orden sucesivo (primera columna = abreviatura; segunda columna = longitud en Bytes/decimal; tercera columna = explicación).

DR	1	Código de la unidad de disco. 0 = falta el aparato 1-16 = selección automática del aparato A-P.
FN	8	Nombre del fichero o grupo de datos.
EXT	3	Extensión o ampliación.
EX	1	Número de la extensión actual.
S1,S2	2	Variable interna.
RC	1	Record Counter = contador de grabación dentro de la extensión actual.
ALLT	16	Tabla de ubicación de bloques (es cargada por el sistema BDOS).
CR	1	Número de grabación momentáneo para una operación de escritura/lectura.
R0,R1,R2	3	Número de grabación aleatorio (Random), opcional.

## El vector de ubicación

El vector de ubicación (ALV = Allocation Vektor) o tabla de ocupación, desempeña un papel central en la gestión de disquetes de sistemas operativos similares al CP/M. Su disposición es muy sencilla:

	Byte 0	Byte 1	Byte 2	.....
Bit:	76543210	76543210	76543210	

Cada Bit equivale a un bloque en el disquete. Se empieza a contar con el 0. Si un bloque ya está ocupado, se coloca el correspondiente Bit. El sistema operativo DOS debe mantener constantemente actualizado el vector de ubicación ALV (tabla de ocupación), puesto que de lo contrario se registran mal los datos en el disquete. Si se coloca un nuevo disquete y éste avisa de su presencia con ^C, el sistema operativo DOS se encarga de que se vuelva a colocar el ALV. El sistema operativo DOS, con ayuda del denominado CSV comprueba si se ha cambiado un disquete.

La longitud del vector de ubicación ALV es esencialmente la cantidad total de bloques, dividida por 8.

### **El vector de comprobación de tamaño, Check Size Vektor**

El CSV, (Check Size Vektor, vector de comprobación de tamaño), tiene la única misión de decirle al sistema operativo DOS si se ha cambiado un disquete. Esto funciona en la forma siguiente:

El sistema operativo DOS calcula una suma de comprobación de un Byte en cada grabación de índice o directorio (128 Bytes, 4 registros de 32 Bytes cada uno) y deposita ordenadamente en el CSV estos Bytes de comprobación. El CSV es actualizado constantemente durante el trabajo con el disquete. Un signo ^C obliga a la nueva incorporación del CSV. Si ahora se cambia un disquete, sin haberlo anunciado previamente (^C), el sistema operativo DOS lo reconoce en el 99,99% de los casos por una incompatibilidad de CSV. Sin embargo, también podría ocurrir perfectamente que dos grabaciones de índice o directorio tengan la misma suma de comprobación para diferentes contenidos. Hay que resignarse a correr este riesgo.

La longitud del CSV se obtiene dividiendo por 4 la cantidad total de registros de directorio.



## El vector de salto (Jump) del sistema operativo BIOS

El vector de salto o desviación (Jump) del sistema operativo BIOS tiene la siguiente estructura. Usted puede encontrar el vector del sistema BIOS con el sistema CP/M en la forma siguiente: ponga en marcha un desparasitador, por ejemplo el DDT y compruebe qué dirección hay en el punto 1 del almacenador. Allí encuentra la dirección wboot de la rutina de puesta en marcha en caliente en el vector de salto.

Las entradas al vector del sistema BIOS en aparatos de I/O sin disco (por ejemplo, consola, lista, . . .), son normalmente inofensivas y lo más que puede ocurrir es que derrumben el sistema, si hacen alguna maniobra equivocada. En cambio, las entradas en las rutinas de disco ya pueden producir perjuicios notablemente más importantes, por lo tanto, primeramente hay que sacar el disquete y meter otro "sin valor".

wboot:	JP BOOT	; arranque en frío sistema CP/M
	JP WBOOT	; arranque en caliente del sistema CP/M
	JP CONST	; comprueba el estado de la consola
	JP CONIN	; trae un signo de la consola
	JP CONOUT	; envía un signo a la consola
	JP LIST	; envía un signo a la impresora
	JP PUNCH	; envía un signo al perforador
	JP READER	; recoge el signo de la unidad de disco
	JP HOME	; coloca el carro de la cabeza en la pista 0
	JP SELDSK	; selecciona unidad de disco
	JP SETTRK	; coloca el número de pista
	JP SETSEC	; coloca número de grabación
	JP SETDMA	; coloca la dirección DMA (memoria intermedia de grabación)
	JP READ	; lee una grabación
	JP WRITE	; escribe una grabación
	JP LISTST	; comprueba el estado de la impresora
	JP SECTRAN	; traduce el número de grabación.

<b>Rutina</b>	<b>Entrada</b>	<b>Salida</b>
CONS		A=FF signo preparado A=00 no hay signo preparado
CONIN		A=signo introducido
CONOUT	C=signo	
LIST	C=signo	
PUNCH	C=signo	
READER		A=signo (1AH si es fin de fichero EOF)
HOME		
SELDSK	C=unidad de disco	HL=dirección DPH HL=0 no hay unidad de disco
SETTRK	BC=número de pista	
SETSEC	BC=número de grabación	
SETDMA	BC=dirección DMA	
READ		A=0 todo está en orden; en caso contrario error
WRITE	C=modo de escritura	A=0 todo está en orden; en caso contrario error
LISTST		A=0 impresora no preparada A=FF impresora preparada
SECTRAN	BC=número de grabación DE=Transtable	HL=transmite número de grabación

## **Entradas del sistema operativo BDOS**

Como ya sabemos, el sistema operativo BDOS es una parte esencial del sistema CP/M. El BDOS es sólo una biblioteca de programa. Si desea hacer programas de máquina con el sistema CP/M, puede utilizar también las funciones del BDOS y, lo que es mejor, la solicitud de una función del BDOS es independiente del sistema; es decir, si ahora escribe un programa del sistema CP/M en su ordenador CPC, que trabaje rigurosamente con funciones BDOS, esto funcionará

también, por ejemplo, inmediatamente en un KAYPRO, sin ninguna limitación (exceptuando quizás una adaptación de video). La entrada en 5 del sistema BDOS es precisamente un punto de interconexión estandarizado.

El número de la función del sistema BDOS debe pasarse al registro C, al solicitarlo.

## Funciones del sistema operativo BDOS

Número	Función	Entrada	Salida
0	Reajuste de sistema		
1	Entrada a consola		A = carácter
2	Salida de consola	E = carácter	
3	Entrada al lector		A = carácter
4	Salida perforadora	E = carácter	
5	Salida por listado	E = carácter	
6	E/S a consola	E = FF (entrada) E = FE (estado) E = carácter (salida)	
7	Obtención del byte E/S		A = byte de E/S
8	Ajuste del byte E/S	E = byte de E/S	
9	Impresión	DE = puntero	
10	Lectura del tampón de consola	DE = puntero	caracteres en tampón
11	Obtención del estado de la consola		A = 00
12	Versión retorno		HL = versión
13	Ajuste del sistema de disco		
14	Selección de disco	E = núm. de disco	
15	Apertura de fichero	DE = fcb	A = FF (no encontrado)
16	Cierre de fichero	DE = fcb	A = FF (no encontrado)
17	Búsqueda del primero	DE = fcb	A = código DIR
18	Búsqueda del siguiente	DE = fcb	A = código DIR
19	Borrado de fichero	DE = fcb	A = FF (no encontrado)

Número	Función	Entrada	Salida
20	Lectura secuencial	DE = fcb	A = código de error
21	Escritura secuencial	DE = fcb	A = código de error
22	Creación de fichero	DE = fcb	A = FF (no hay espacio)
23	Cambio de nombre de un fichero	DE = fcb	A = FF (no encontrado)
24	Retorno al vector de registro		HL = vector
25	Vuelta al disco actual		A = número de disco
26	Ajuste de dirección del DMA	DE = dirección del DMA	
27	Obtención de la dirección ALLC		HL = dirección
28	Disco protegido contra escritura		
29	Obtención del vector R/O		HL = vector
30	Ajuste de los atributos de fichero	DE = fcb	A = FF (no encontrado)
31	Obtención de la dirección del DPB		HL = dirección
32	Ajuste/Obtención del código de usuario	E = FF (obtención) E = 0...F (ajuste)	A = código de usuario
33	Lectura aleatoria	DE = fcb	A = código de error
34	Escritura aleatoria	DE = fcb	A = código de error
35	Registro de tamaño de fichero	DE = fcb	r0, r1, r2
36	Ajuste de registro aleatorio	DE = fcb	r0, r1, r2
37	Reajuste de la unidad de disco	DE = vector de la unidad	A = 0
38	Acceso a la unidad de disco	no permitido	
39	Unidad de disco libre	no permitido	
40	Escritura aleatoria + llenado	DE = fcb	A = código de error

## **Para finalizar comentaremos algo sobre el Hardware**

### **Unidades X:**

El equipamiento o hardware de las unidades X consta únicamente de un módulo de memoria ROM con un punto de interconexión (opcional) RS232C. El número de ROM se puede ajustar libremente (regleta de clavijas de dos filas de 16 polos). El soporte de la ROM puede alojar memorias ROM tanto de 16K como de 32K. Estas últimas se gestionan como si fueran dos memorias ROM de 16K cada una y reciben su propio número de ROM. Las entradas de maniobra del punto de interconexión en serie son:

FADC datos SIO  
FADD control SIO  
FBDC contador 0  
FBDD contador 1  
FDBF palabra de modo

### **Unidades de disco + controlador:**

Las unidades de disco con controlador propio ocupan en su versión estándar las entradas:

#### **Ordenador CPC 464:**

FB7E control del uPD  
FB7F datos del uPD  
FA7E motor

#### **Ordenadores CPC 664/6128**

FBF6 control del uPD  
FBF7 datos del uPD  
FA7E motor

para el mando de la unidad de discos. En el controlador con módulo Eprom de memoria integrada ROM se pueden ajustar las direcciones de las entradas libremente por medio de un decodificador de ROM.

## Ocupación de las clavijas de los conectores

**Conectores de los puntos de interconexión del disco flexible en el controlador:**

### 1. Conector de 5.25":

6	/DS3
8	/INDEX
10	/DS0
12	/DS1
14	/DS2
16	/MOTOR
18	/DIRECTION
20	/STEP
22	/WRDATA
24	/WRGATE
26	/TRACK 00
28	/WRPROT
30	/RDATA
32	/SIDE
34	/READY
1	+12V (Input)
3,5,7	+5V (Input)

Todos los demás con base impar (masa).

### 2. Conector de 3":

27	/INDEX
25	/DS0
23	/DS1
19	/MOTOR
17	/DIRECTION
15	/STEP
13	/WRDATA
11	/WRGATE
9	/TRACK 00
7	/WRPROT
5	/RDATA

3        /SIDE  
 1        /READY  
 21,29,31,33 +5V (Input)

Todos con base par (masa).

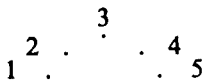
### 3. Conector RS232C:

Fundamentalmente, todos los puntos de interconexión RS232C de VORTEX poseen un conector DTE "macho" de 25 polos.

2        /TXD  
 3        /RXD  
 4        RTS  
 5        CTS  
 7        Masa  
 8        DCD  
 20      DTR  
 22      R. IN

### Clavija hembra de enchufe DIN en la unidad X de disco flexible:

(Clavija vista por detrás)



1, 2 +5V (en la clavija de 3 polos, sólo 1)  
 4, 5 +12V (en la clavija de 3 polos, sólo 5)  
 3 Tierra

Tanto el controlador con punto de interconexión RS, como la platina X/RS poseen también una clavija DIN, cuya ocupación de bornes es exactamente la contraria:

1, 2 +12V  
 4, 5 +5V  
 3 Tierra

## Alimentación de corriente al controlador de floppys VORTEX (no a la unidad X de disco flexible):

El controlador de floppys recibe corriente de +5V a través de la regleta de bornes B3 en la caja de la unidad de disco (es decir, a través de un cable de cinta plana de 34 polos).

2	1
4	3
6	5
8	7
10	9
12	11
14	13
16	15
	18
	17
	20
	19
	22
	21
	24
	23
	26
	25
	28
	27
	30
	29
	32
	31
	34
	33

3, 5, 7 +5V

Todos los demás masa impar.

Los números grabados se entienden mirando hacia la abertura de la regleta de bornes.

La regleta de bornes está situada a unos 12 cm. de distancia de los dos (1) enchufes de las fichas o tarjetas, que se enchufan en las unidades de disco.

hacia los enchufes de las tarjetas.



## **Unidades de disco flexible**

La unidad de disco conectada al controlador VORTEX cumple las siguientes especificaciones:

Capacidad (sin formatear)	1MB
Índice de transmisión de datos	250K Bit/seg.
Tiempo de cambio de pista	mínimo 4 mseg
Tiempo de reposo de la cabeza	máximo 20 mseg
Tiempo de aceleración del motor	máximo 1 seg.
Número de revoluciones	300 rpm.
Cantidad de cabezas	2
Pistas/Superficie	mínimo 80
Modulación	FM/MFM
Densidad de pista	96 tpi

Nos reservamos el derecho a efectuar todo tipo de modificaciones.



## APÉNDICE 1

### El problema EOF

EOF (End of file, fin de grupo de datos).

El sistema operativo VDOS 2.0 tiene, igual que el VDOS 1.0, un "punto débil", que no debe achacarse a ningún error de programación o falta de atención, sino que se debe principalmente a la estructura del sistema BASIC del ordenador CPC y al intento de hacerlo compatible con los elementos de 3". Como quizás ya sepa, el sistema operativo AMSDOS tiene, entre otros, el "error MERGE" o error de fusión. Este error consiste en que usted puede encontrarse repentinamente con un error "EOF met" al "fusionar" un programa BASIC, sin haberse alcanzado en realidad el final del grupo de datos. A pesar de que es tan sencillo encontrar el diagnóstico del error, en cambio es muy difícil corregirlo. Ciertamente existe, por ejemplo, la corrección de errores "DATA Becker Patch", que aparentemente suprime este problema, pero, en su lugar, produce otros errores. Para no limitarnos a simples explicaciones, damos a continuación algunas informaciones de fondo:

El fin de un grupo de datos ASCII se marca, de acuerdo con las normas prefijadas, mediante `CTRL Z` (1A hexadecimal). Esto es importante, siempre que no sea suficiente para poder reconocer el final del módulo 128, es decir, por registros o grabaciones (esto es el denominado "Hard end" de un grupo de datos), sino que además se tenga interés en volver a encontrar otra vez este final con la exactitud de un Byte. Por este motivo se utiliza también en el sistema AMSDOS y, por motivos de compatibilidad, también en el sistema VDOS, el 1A como señalizador del final de fichero o grupo de datos EOF. Esto está bien si se quiere asegurar el que el 1A aparezca sólo al final y no en cualquier parte del propio grupo de datos. Pero, aquí comien-

zan ya las dificultades: la microprogramación en memoria muerta (Firmware) en el ordenador CPC prevé dos posibilidades estandarizadas para la introducción por lectura de datos desde la periferia (cassette/disquete):

1) función 128 (BC80h): **CAS/DISC IN CHAR**  
introduce signos individuales.

2) Función 129 (BC83h): **CAS/DISC IN DIRECT**  
introduce bloques de memoria enteros.

Los problemas los ocasiona la función 128. Aquí tenemos las siguientes condiciones de retorno (Carry y Zero son señalizadores del procesador del Z80):

Carry true (arrastre verdadero)  
Zero false (cero falso)

si se ha podido introducir un signo desde la unidad periférica, y

Carry false (arrastre falso)  
Zero false (cero falso)

si esto no ha sido posible, es decir, si se ha alcanzado el final del grupo de datos.

Pero, de todos modos, sólo existen dos criterios para el final de un grupo de datos:

Criterio 1: se ha detectado **CTRL Z** (IAH) (Soft end)

Criterio 2: se ha alcanzado el final físico, es decir, éste ha sido el último registro o grabación (Hard end)

El sistema BASIC del ordenador CPC comprueba ahora exactamente el final del grupo de datos, a través de la función 128, siendo indiferente el que se cargue un grupo de datos neto ASCII o bien un grupo de datos binario (por ejemplo, un programa BASIC).

Por este motivo, puede obtener un "EOF met" en el medio de una "intercalación" de un programa BASIC, por ejemplo, si aparece un número de línea 26 (IAH). El criterio 1 del fin del grupo de datos EOF remacha implacablemente esta situación. La corrección de datos "DATA Becker Patch", corrige este inconveniente, pero, por otro

lado, se encarga de que no se pueda ya encontrar correctamente el final de un grupo de datos ASCII. (El "Patch" elimina el criterio 1).

El dilema consiste en que mediante la misma rutina, a saber la función 128 del "Firmware", no se pueden cargar el ASCII y los grupos de datos binarios con el mismo señalizador EOF; esto conduce inevitablemente a una contradicción, que sólo se puede solucionar con un "o bien . . . o bien . . .". Lamentablemente, esto no se tuvo en cuenta en el sistema BASIC del ordenador CPC y/o en el sistema AMSDOS.

En cuanto al sistema operativo VDOS: en la versión 1.0 se tuvo en cuenta sólo el criterio 1, por lo que también era posible "intercalar" a voluntad. De todos modos, ya había dificultades para encontrar el fin del grupo de datos de un fichero ASCII secuencial; según los casos, se podían introducir hasta 127 signos de más.

El VDOS 2.0 soluciona este nudo gordiano en la forma siguiente: el usuario decide cuál de los criterios EOF necesita. Para ello se utiliza la función **DERROR** del RSX.

!**DERROR,2** **ENTER** activa tanto la detección neta del "Hard end" como la comprobación del señalizador 1AH (Soft end) del EOF y ha sido proyectado preferentemente para poder reconocer de forma adecuada el final de los grupos de datos secuenciales del texto. Naturalmente, se produce aquí de manera implacable el problema "EOF met" al "intercalar".

Por eso, existe también la orden !**DERROR,3** **ENTER**: esto es, por lo demás, el estado de defecto después de la conexión o después de un RESET y equivale al VDOS 1.0; aquí se comprueba solamente el final físico del grupo de datos (Hard end), es decir, no se produce ningún problema "EOF met" al "intercalar", pero, en cambio, aparecen dificultades en los grupos de datos secuenciales del texto. Por ello, deberá decidir usted mismo en su programa, qué es lo que menos le molesta.