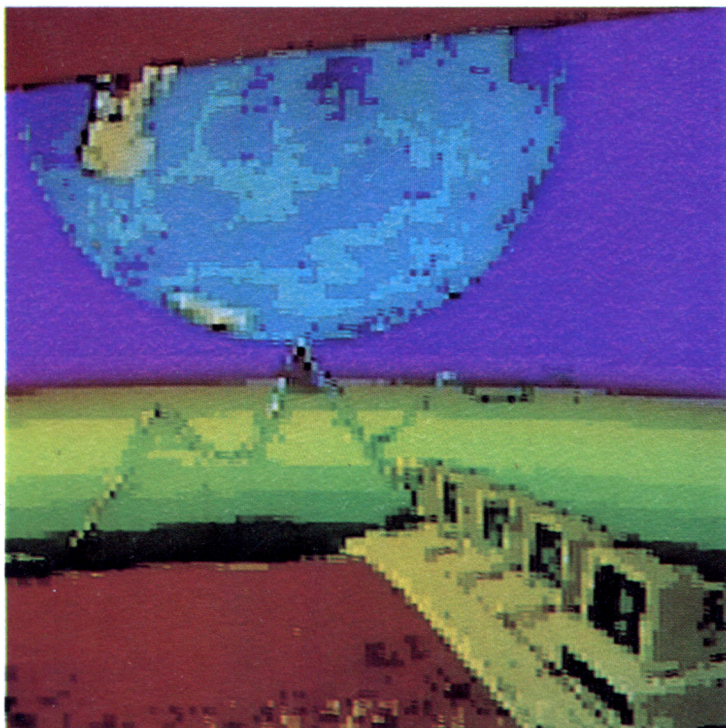


GRAN BIBLIOTECA AMSTRAD



SISTEMA OPERATIVO CP/M

UNA PUERTA ABIERTA A CIENTOS DE APLICACIONES

GRAN BIBLIOTECA
AMSTRAD

2

SISTEMA OPERATIVO CP/M

Director editor:

Antonio M. Ferrer Abelló

Director de producción:

Vicente Robles

Director de la obra:

Fernando López Martínez

Redactor técnico:

Carlos de la Ossa Villacañas

Colaboradores:

Data-3 Informática, S.A.

Pilar Manzanera Amaro

Amelia Polo García

Diseño:

Bravo/Lofish

Maquetación:

Carlos González Amezúa

Dibujos:

José Ochoa

© Ediciones Ingelek, S. A.

Todos los derechos reservados. Este libro no puede ser, en parte o totalmente, reproducido, memorizado en sistemas de archivo, o transmitido en cualquier forma o medio, electrónico, mecánico, fotocopia o cualquier otro sin la previa autorización del editor.

ISBN del tomo: 84-7708-006-2

ISBN de la obra: 84-7708-004-6

Fotocomposición: Andueza, S. A.

Imprime: Héroes, S. A.

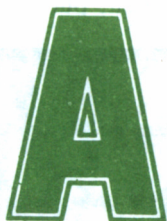
Depósito Legal: M-31329-1986

Precio en Canarias, Ceuta y Melilla: 435 ptas.

SISTEMA OPERATIVO CP/M

<u>Introducción.</u>	5
<u>¿Qué es un sistema operativo?</u>	9
<u>Signos y convenios.</u>	17
<u>Cuestión de sintaxis.</u>	25
<u>Anatomía del CP/M.</u>	37
<u>A fondo.</u>	45
<u>Antes de seguir.</u>	53
<u>Los comandos residentes.</u>	67
<u>ED: el editor de CP/M.</u>	79
<u>Directorios para todos los gustos.</u>	95
<u>PIP: genio de las transferencias.</u>	105
<u>SET: bajo la protección de CP/M.</u>	117
<u>Como complemento de SET.</u>	131

INTRODUCCION



principio de los años setenta, Gary Kildall, uno de los genios legendarios de Silicon Valley, comenzó a comercializar una versión de lo que él mismo denominó CP/M, siglas de *Control Program for Microprocessors* o programa de control para microprocesadores. El éxito alcanzado por aquella versión, la llevó a fundar la Digital Research, hoy en día una de las más importantes firmas a nivel mundial de software de base y aplicaciones.

Desde entonces, el CP/M ha contado con un sin fin de modificaciones y versiones, de las cuales la 2.2, lo llevó al primer lugar entre los sistemas operativos para micros de 8 bits. Desarrollado en principio para el microprocesador 8080 de Intel, las últimas versiones completamente adaptadas al Z80 han ejercido el empuje necesario para su gran éxito, al ser esta unidad central la más utilizada en micros de 8 bits.

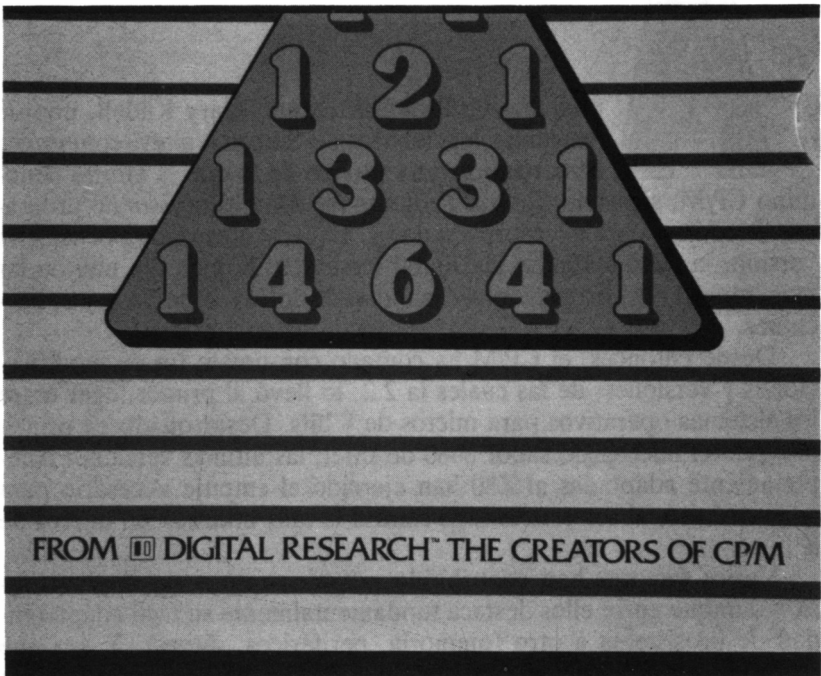
Varios factores han contribuido notablemente a su amplia difusión, aunque entre ellos destaca fundamentalmente su fácil adaptabilidad de un sistema a otro (memoria, periféricos, discos). Y eso que según la mayoría de los expertos, el CP/M es un sistema operativo nada sencillo de aprender por los usuarios inexpertos, con una nefasta

gestión de errores por la irreconocible interpretación de sus mensajes y un sistema de acceso al disco bastante lento. Sin embargo, la amplia biblioteca de programas a su disposición lo convierten en la estrella de los sistemas operativos para casi todas las firmas fabricantes de microordenadores.

La última versión de CP/M es la 3.0, conocida comúnmente como CP/M Plus, preparada para gestionar volúmenes de memoria acordes con los disponibles hoy en día y con otras muchas sofisticaciones típicas de los sistemas operativos modernos. En este libro nos centraremos en ella, por ser la adoptada por AMSTRAD en sus ordenadores CPC 6128 y PCW 8256/8512.

Cuando decidimos escribir este libro, la primera idea que vino a nuestra cabeza fue la limitación de espacio que suponía intentar condensar en un único volumen todas las características y particularidades de un sistema operativo como CP/M.

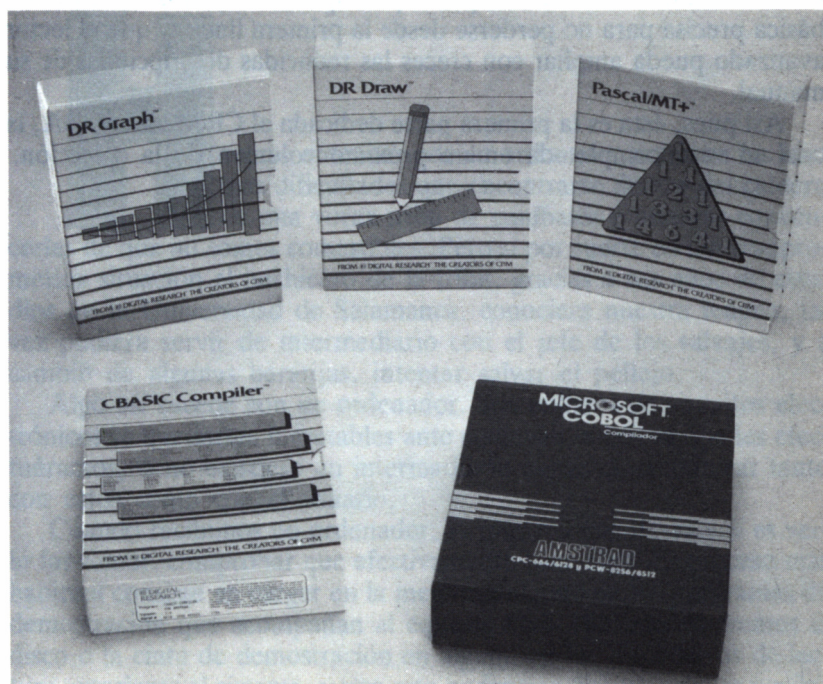
Al principio, se pensó en realizar una guía en la que sin entrar en mayores profundizaciones, se comentaran los principales comandos del sistema operativo. Pero pronto desechamos la idea, puesto que



2.0.1. Digital Research: «LOS CREADORES DE CP/M».

HIDROPROCESADOR	MONOUSUARIO		MULTIUSUARIO	
	MONOTAREA	MULTITAREA	MONOTAREA	MULTITAREA
8080 Z-80	CP/M PLUS	MP/M II	MP/M II	CP/NET
8086	CP/M-86	CP/M CONCURR.	MP/M-86	CP/NET 86
6800	CP/M 68 K			
Z8000	CP/M 78 K			

2.0.2. La familia CP/M: multiusuario y monousuario.



2.0.3. Son muchas las aplicaciones soportadas por el sistema operativo CP/M.

nos sentimos plenamente identificados con todos aquellos usuarios de AMSTRAD que ansiosos de conocer y profundizar en las posibilidades de su equipo, deciden adentrarse en el intrincado mundo del CP/M y, rápidamente se sienten decepcionados, o mejor dicho, faltos de información, cuando comprueban las sucintas explicaciones que en los manuales se le ofrecen sobre este sistema operativo.

Además, de nuestra propia experiencia contestando a las consultas que sobre temas informáticos realizan los lectores de las publicaciones periódicas de INGELEK, S. A. (a veces nos han planteado cuestiones como ¿puedo programar en «lenguaje» CP/M?, o ¿este programa, puede procesar textos?), comprendimos que el usuario poco avanzado desconoce qué es un sistema operativo, para qué sirve, o qué es lo que se debe esperar de él.

Por ello, resolvimos escribir el original con la mayor rigurosidad posible, aun dentro de las limitaciones de espacio comentadas (piense el lector que algunos comandos de CP/M tienen docenas de parámetros u opciones que modifican sustancialmente su forma de actuar). De esta manera intentamos que el principiante reciba la información básica precisa para no perderse desde la primera línea, y que el lector avanzado pueda ampliar con creces las reducidas descripciones de su manual.

Así pues, ésta es la primera parte dedicada al CP/M en la GBA, la cual se verá completada en un próximo volumen de la colección.

¿QUE ES UN SISTEMA OPERATIVO?



acerse entender por un ordenador presenta las mismas dificultades que encontrarse dentro del caldero de una voraz tribu de caníbales e intentar convencerles de que no somos comestibles. Pero si por suerte en tan comprometida situación el hechicero de la tribu, gracias a sus años de estudios en la Universidad de Salamanca, conociera nuestra lengua, tal vez pudiera servir de intermediario con el jefe de los salvajes, y a cambio de algunas baratijas, intentar salvar el pellejo.

Algo así sucede con un ordenador. Sus millares de circuitos electrónicos se quedarían inmutables ante cualquier solicitud que les efectuaráramos, si no existiese un intermediario capaz de comunicar tanto con estos como con el usuario.

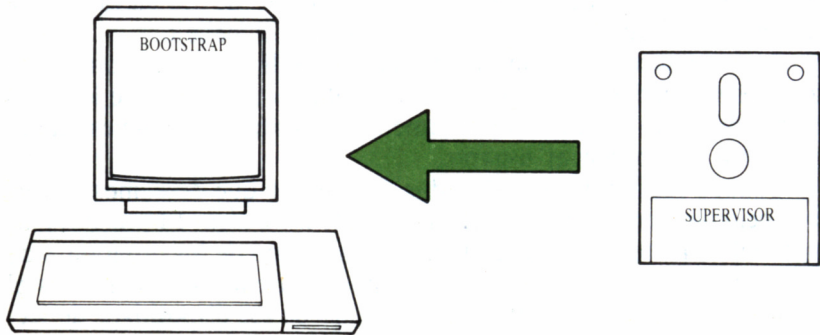
Cuando recibimos un ordenador, lo primero que deseamos es verlo funcionar, comprobar que efectivamente hace algo. El proceso más habitual consiste en cargar en la memoria alguno de los programas de demostración que acompañan al equipo. Para ello introduciremos el disco o la cinta de demostración en alguno de los dispositivos de lectura exteriores al sistema, y tras esto podremos disfrutar de las posibilidades de nuestro ordenador.

Obviamente, para efectuar esta transferencia de información del

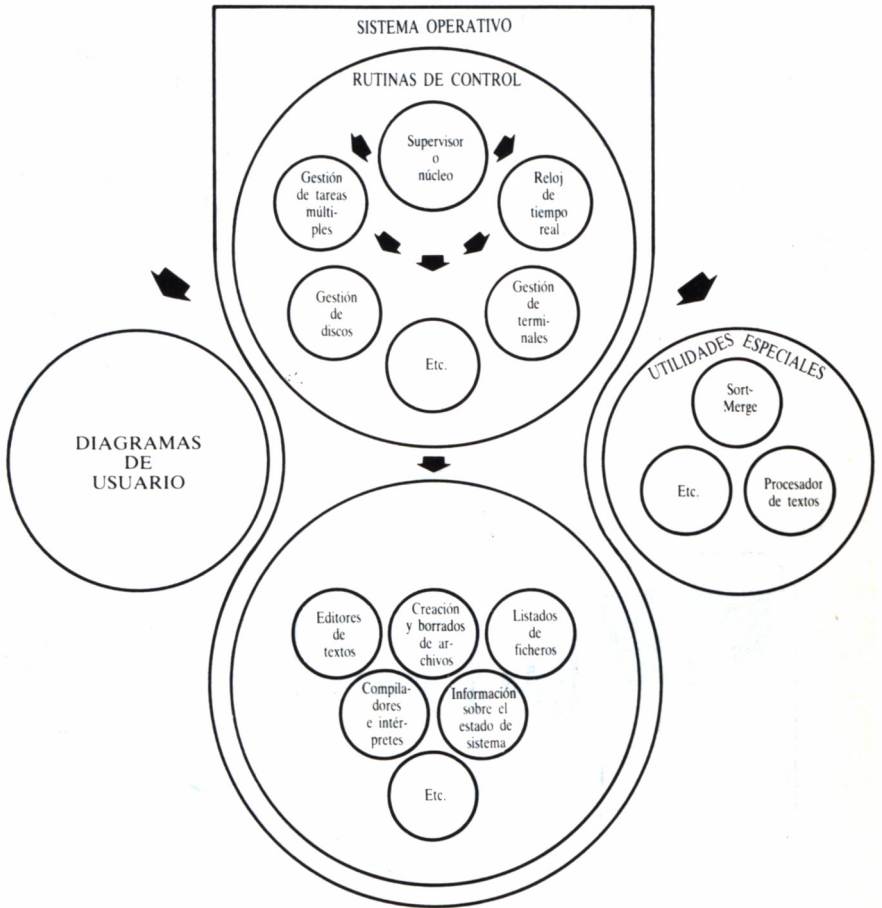
disco o cinta hasta la memoria, debe existir otro programa que se encargue de gestionarla, pues de otra manera el ordenador no tendría ni idea de cómo efectuarlo. Pero no es sólo esto lo que ha de saber el mencionado programa; además, debe interpretar las señales y a la par emitir las adecuadas para que la información pueda ser representada en la pantalla.

Todo el conjunto de programas que permiten obtener de forma cómoda y rápida el máximo rendimiento del ordenador es lo que se conoce como sistema operativo. Más precisamente, un sistema operativo es un conjunto de procedimientos manuales y automáticos que permiten al usuario efectuar todo tipo de operaciones con su ordenador.

A nadie se le escapará en este momento que son muchas las tareas que debe realizar el sistema operativo, por lo que en general la memoria que éste puede llegar a ocupar desbordará la capacidad del



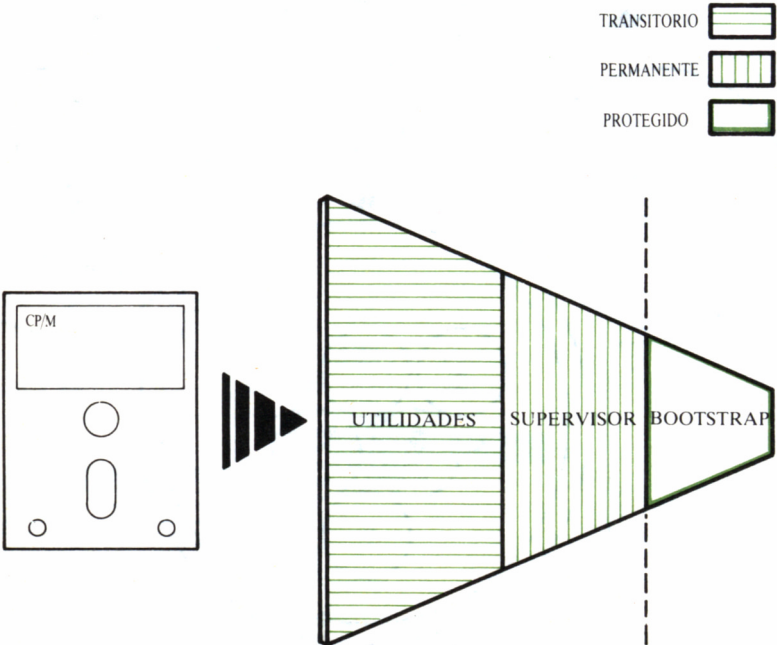
2.1.1. Carga del supervisor.



2.1.2. Esquema general de un sistema operativo.

equipo. Para evitar este inconveniente el sistema operativo está compuesto por muchos programas, habitualmente almacenados en disco, que únicamente son transferidos a la memoria central cuando se necesita; sin embargo, existe un programa que permanece constantemente en la memoria pues debe encargarse de controlar el buen funcionamiento del sistema y de gestionar todas las transferencias de información entre la memoria y los periféricos.

Este programa recibe el nombre de supervisor o núcleo, y normalmente es el encargado de llevar desde el disco a la memoria cualquier programa secundario del sistema operativo que le solicitemos. Muy bien, pero ¿quién se encarga de ubicar el supervisor en la memoria? Naturalmente, otro programa aunque de pequeño tamaño pues su única misión consiste en leer el supervisor del disco y guardarlo en la memoria ROM por dos motivos principales: en principio no se puede perder su contenido cada vez que desconectemos el ordenador de la red y, por otra parte, debe estar protegido contra cualquier accidente provocado porque el usuario, sin querer, escriba sobre él alterando su contenido y convirtiéndolo en algo inútil para realizar su misión.

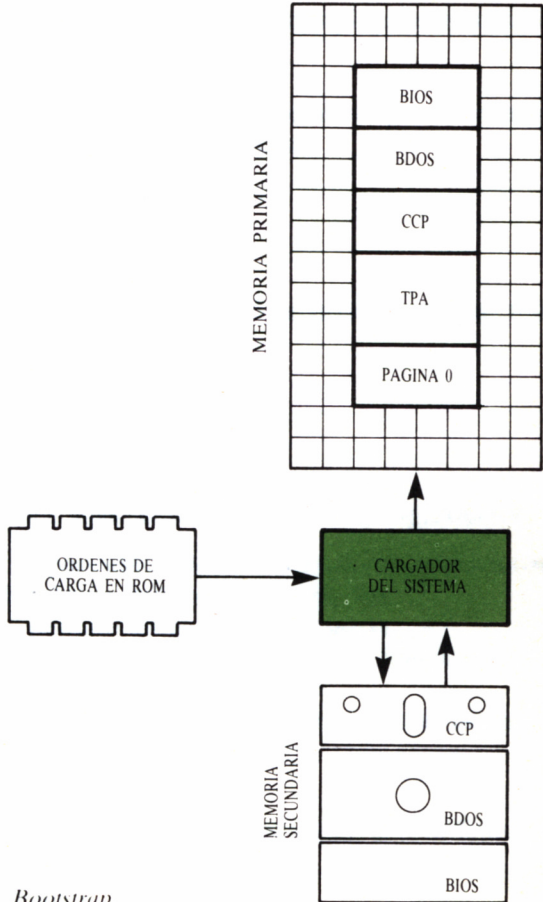


2.1.3. Distribución en memoria de Bootstrap, supervisor y utilidades.

FUNCIONES DEL SISTEMA OPERATIVO

Como comentamos en el párrafo anterior, el sistema operativo abarca una serie de subrutinas y subprogramas que se ocupan de realizar tareas de vital importancia para el funcionamiento del sistema. Entre ellas cabe destacar las siguientes:

— Comandos o programas de utilidad para visualizar y modificar la información contenida en las memorias de almacenamiento exterior. La información referente a cada fichero almacenado en disco o a un conjunto de éstos (nombre, tamaño, espacio disponible, etc.) es muy interesante poder observarla bajo diferentes formatos con el fin de seleccionar el más adecuado en cada momento.



2.1.4. *Proceso de Bootstrap.*

— Facilidades de intercambio de información entre los dispositivos de almacenamiento. El sistema operativo debe contener programas de utilidad para efectuar el proceso de formateado de disketes, la copia de estos, etc.

— Protección contra accidentes. Entre las labores del sistema operativo debe encontrarse la solicitud de confirmación cuando por ejemplo tratamos de eliminar un fichero del disco, dándonos la oportunidad de rectificar por si acaso hemos cometido un error en la manipulación de las teclas.

EJERCICIOS

1. ¿Qué es un sistema operativo?
 - A) Un lenguaje de programación.
 - B) El bisturí de un cirujano.
 - C) El conjunto de programas que permiten obtener el máximo rendimiento del ordenador.

2. ¿Reside el sistema operativo en la memoria central del Amstrad?
 - A) Sí, por completo. Normalmente se almacena en la ROM.
 - B) No enteramente.
 - C) No, porque cuando llegó estaba todo completamente ocupado.

3. ¿Cómo se llama la parte del sistema operativo que se encarga de dar paso a las diferentes secciones de éste?
 - A) Nucleo y Sicleo (que dice el chino).
 - B) Núcleo o Revisor.
 - C) Núcleo o Supervisor.

4. ¿Con qué procedimiento se carga el supervisor en la memoria?
 - A) Bootstrap.
 - B) Lo carga el usuario.
 - C) Método de carretilla y pala.

5. ¿Facilita el sistema operativo las acciones de «piratería»?
 - A) Sí, poniendo a disposición de todos los usuarios un barco corsario, con 100 cañones por banda y bla, bla...
 - B) Sí, porque nos permite cargar los programas.
 - C) Sí, lamentablemente, aunque no es ésta su finalidad.

6. ¿Quién se encarga de coordinar las transferencias de datos entre la memoria central y los periféricos?
 - A) El coordinador de información.
 - B) La Dirección General de Tráfico (Recuerda: si coordinas, no codutzzcas).
 - C) El sistema operativo.

7. ¿Incluye el sistema operativo ayudas a la programación?

A) No. Eso es tarea del intérprete.

B) Sí.

C) Sí, principalmente el ADA (Ayuda Del Automovilista).

SIGNOS Y CONVENIOS



lo largo de este volumen se siguen una serie de convenios para intentar evitar, al máximo de nuestras posibilidades, ambigüedades en la definición de la sintaxis en las diferentes órdenes que son tratadas. Para ello, se han utilizado signos de los que a continuación pretendemos aclarar su significado.

Una línea de órdenes de CP/M está formada por un COMANDO, una COLA de comando opcional y un retorno de carro (*carriage return*). El COMANDO es el nombre del fichero que contiene el programa a ejecutar, mientras que la COLA puede consistir en una especificación de unidad de disco, una o más especificaciones de fichero y una serie de opciones o parámetros, siguiéndose esta estructura:

A>COMANDO {COLA de comando} <cr>

Se han utilizado los siguientes signos especiales en las líneas de sintaxis general de las diferentes órdenes:

- { } determinan un dato opcional dentro de la orden.
- | actúa como separador entre datos o parámetros alternativos.

<cr>	define un retorno de carro (CR, Carriage Return), habitualmente para especificar que se ha de pulsar la tecla RETURN para enviar la orden a CP/M.
↑	indica la pulsación de la tecla CONTROL (ALT, en los PCW), cuya abreviatura es CTRL.
n	sustituye un dato numérico.
s	señala una cadena de caracteres alfanuméricos.
o	indica una opción dentro de una determinada orden.
[]	los corchetes se emplean en CP/M para encerrar las siglas que definen una opción.
RW	representa el atributo del sistema RW (Read/Write, lectura/escritura). Es opuesto a RO.
RO	representa el atributo del sistema RO (Read Only, sólo lectura), opuesto a RW.
SYS	indica el atributo del sistema SYStem, contrario a DIR. Los ficheros con esta característica no se visualizan al efectuar un catálogo del directorio.
DIR	es el atributo del sistema DIR, en oposición a SYS.
...	indica que el elemento que los precede dentro de la sintaxis de una orden puede repetirse el número de veces requerido por el comando.
*	símbolo comodín: sustituye a parte, o al nombre completo de un fichero o su extensión.
?	símbolo comodín: sustituye a un único carácter que ocupe la misma posición que este signo dentro del nombre del fichero o en su extensión.

CARACTERES DE CONTROL

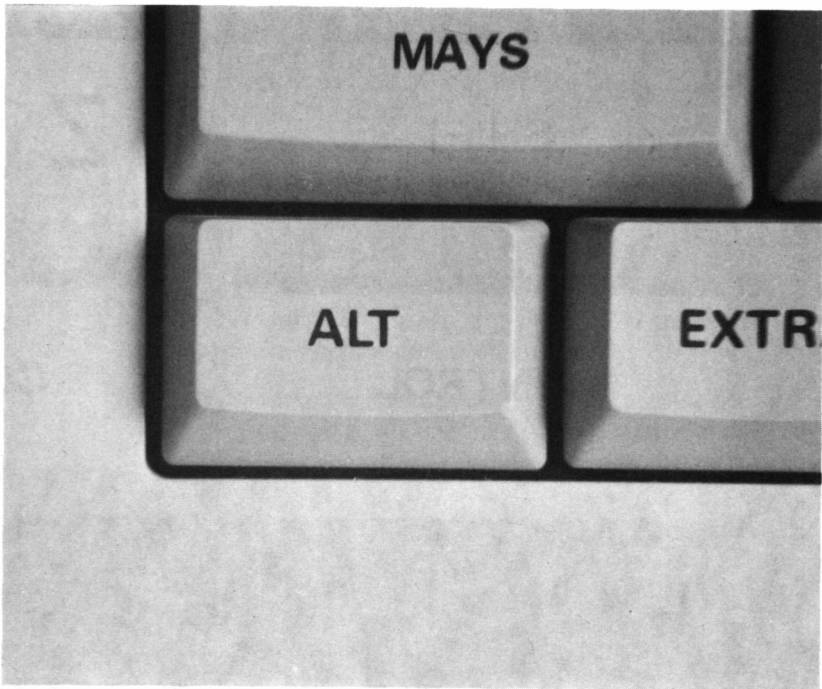
Normalmente, en todas las operaciones que requieran la intervención del usuario, se emplean determinadas teclas para indicar al sistema operativo que tome una determinada acción. A continuación se muestra una lista completa de los caracteres de control que obedece CP/M Plus:

- CTRL-A desplaza el cursor un carácter a la izquierda.
- CTRL-B sitúa el cursor al principio o al final de una línea de comandos alternativamente, sin modificarla.
- CTRL-C detiene la ejecución de un programa, incluso en el caso de que hubiera sido interrumpido temporalmente mediante CONTROL+S (ALT+S).
- CTRL-E fuerza un retorno de carro automático, pero no envía la línea actual a CP/M. Si el cursor no estaba situado al final de ésta, los caracteres a partir de la posición del cursor en adelante, saltan a la siguiente línea, pero una vez enviada, se interpreta correctamente.
- CTRL-F desplaza el cursor una posición hacia la derecha sobre la línea de órdenes.
- CTRL-G borra el carácter situado bajo el cursor en la línea de órdenes.



— 2.2.1. La tecla de CONTROL tiene su homólogo en la ALT de los PCW.

- CTRL-I actúa como tabulador (idéntico resultado que al pulsar la tecla TAB).
- CTRL-H borra el carácter situado a la izquierda de la posición actual del cursor.
- CTRL-J provoca un retorno de carro (CR) más un salto de línea (LF), siendo enviadas las órdenes actuales a CP/M para su inmediata ejecución.
- CTRL-K borra los caracteres situados bajo el cursor y a la derecha de éste, hasta el final de la línea de órdenes.
- CTRL-M provoca el mismo efecto que un retorno de carro (CR).
- CTRL-P bascula la salida de información hacia la impresora o la pantalla.
- CTRL-Q continúa con la ejecución del listado de un programa interrumpido con CONTROL+S (ALT+S).



■ 2.2.2. *La tecla de ALT, como CONTROL en los CPC, es de gran importancia en el manejo del sistema CP/M.*

- CTRL-R reescribe los caracteres situados a la izquierda del cursor en una nueva línea y la actualiza en el buffer de órdenes.
- CTRL-S detiene el listado de un programa.
- CTRL-U ignora todos los caracteres de la línea de órdenes y sitúa el cursor en la siguiente.
- CTRL-W recupera el contenido de la línea de órdenes anterior, siempre y cuando la actual esté vacía. En caso contrario, desplaza el cursor al final de ésta.
- CTRL-X borra todos los caracteres situados a la izquierda del cursor.

EJERCICIOS

8. ¿Qué componentes forman una línea de órdenes CP/M?
- A) El portero y tres defensas.
 - B) Bootstrap, supervisor y utilidades.
 - C) Comando, cola y retorno de carro.
9. ¿Qué significa CP/M?
- A) Son las siglas de Programa de control para microprocesadores (*Control Program for Microprocessors*).
 - B) Convención de Programadores Mundiales.
 - C) Son las siglas de Gestión de la unidad central (*Central Process Management*).
10. ¿Cómo se representa la pulsación de la tecla RETURN en la sintaxis de una orden de CP/M?
- A) <cr>.
 - B) [RET].
 - C) Con un dedo pulsando la tecla.
11. ¿Cómo se representa la pulsación de CONTROL (ALT, en los PCW) en la sintaxis de una orden de CP/M?
- A) Con dos dedos pulsando la tecla.
 - B) [CTRL].
 - C) ↑.
12. ¿Cómo se representa la pulsación de ESCAPE en la sintaxis de una orden de CP/M?
- A) No se representa.
 - B) [ESC].
 - C) ?.
13. Las opciones de un determinado comando se encierran entre...
- A) Corchetes.
 - B) Rejas.
 - C) Llaves.

14. Dentro del nombre de un fichero ¿Cuántos caracteres sustituye el signo asterisco (*)?

- A) Todos los situados desde la posición ocupada por éste.
- B) El que ocupa en el nombre.
- C) No lo sabe ni él.

15. Dentro del nombre de un fichero ¿Cuántos caracteres sustituye el signo de interrogación (?)

- A) Se interroga precisamente porque no lo sabe.
- B) Todos los situados desde la posición ocupada por éste.
- C) El que ocupa en el nombre.

16. ¿Cómo se detiene definitivamente la ejecución de un programa?

- A) CONTROL-S (ALT-S, en los PCW).
- B) Llegando a tiempo con el indulto.
- C) CONTROL-C (ALT-C, en los PCW).

17. ¿Qué caracter tiene el mismo efecto que DEL (←BORR) en CP/M?

- A) CONTROL-H (ALT-H, en los PCW).
- B) CONTROL-G (ALT-G, en los PCW).
- C) El colérico.

18. ¿Cómo se detiene temporalmente la ejecución de un programa?

- A) Engañando al ejecutor con el grito: «¡Un momento, que ahí llega el indulto!».
- B) CONTROL-C (ALT-C, en los PCW).
- C) CONTROL-S (ALT-S, en los PCW).

19. ¿Y cómo se reanuda?

- A) CONTROL-S (ALT-S, en los PCW).
- B) Comprobando que efectivamente no ha llegado el indulto.
- C) CONTROL-Q (ALT-Q, en los PCW).

20. ¿Cómo se recupera la última línea de órdenes?

- A) CONTROL-W (ALT-W, en los PCW).
- B) CONTROL-R (ALT-R, en los PCW).
- C) Con suerte y buena memoria.

CUESTION DE SINTAXIS

Entre las funciones básicas de CP/M podemos encontrar órdenes destinadas a crear, modificar, borrar, combinar, imprimir y transferir datos desde la memoria al disco o viceversa. Como el lector ya debe conocer, los datos están almacenados en el ordenador según determinadas secuencias de bytes, los cuales responden a un código aceptado internacionalmente: el código ASCII.

Ahora bien, sobre el disco, los datos no pueden estar distribuidos de cualquier forma, puesto que de ser así se convertirían en algo irreconocible e ininterpretable. Por ello, se organizan según estructuras homogéneas que reciben el nombre de registros; éstos se asocian para formar ficheros, constituyendo la gestión de estos bloques el objetivo principal de CP/M, puesto que cuando le indicamos que ejecute una orden determinada, estamos señalando el proceso particular al que deben ser sometidos los bytes almacenados en el fichero o ficheros especificados.

Un registro lógico abarca 128 bytes (equivalente a 2 elevado a 7) y el número de registros que es posible alcanzar en un fichero es de 65536 (2 elevado a 16). Por tanto, la dimensión máxima de un fichero creado por CP/M será de $65.536 \times 128 = 8.388.608$ bytes (2 elevado a 23).

NOMBRES DE FICHEROS

Tanto los programas como los datos se graban en ficheros en los discos, y por este motivo, antes de poder ejecutar un programa es necesario comunicar al CP/M cuál es su nombre, y en algunos casos también el nombre de los ficheros de datos que utilizará.

El identificador de un fichero creado por CP/M debe cumplir unas sencillas reglas sintácticas y podemos considerarlo formado por dos partes: la primera debe ser una combinación cualquiera de hasta ocho caracteres, a excepción de los siguientes:

<> = ! | - + \ ; : . () [] \$ & ? *

y la segunda otra cadena de 3 caracteres como máximo que se denomina extensión o tipo del fichero. Ambas deben ir separadas por un punto (.).

Normalmente al primer grupo de 8 caracteres se le atribuye la tarea de especificar el nombre del fichero, dándole alguno que haga referencia a su contenido, y al segundo el tipo de información contenida en el mismo. Esta última indicación, como veremos más adelante, se convierte en obligatoria para algunos tipos de ficheros sobre los que sólo pueden llevarse a cabo funciones particulares si la extensión es una determinada (por ejemplo, en los ficheros .SUB).

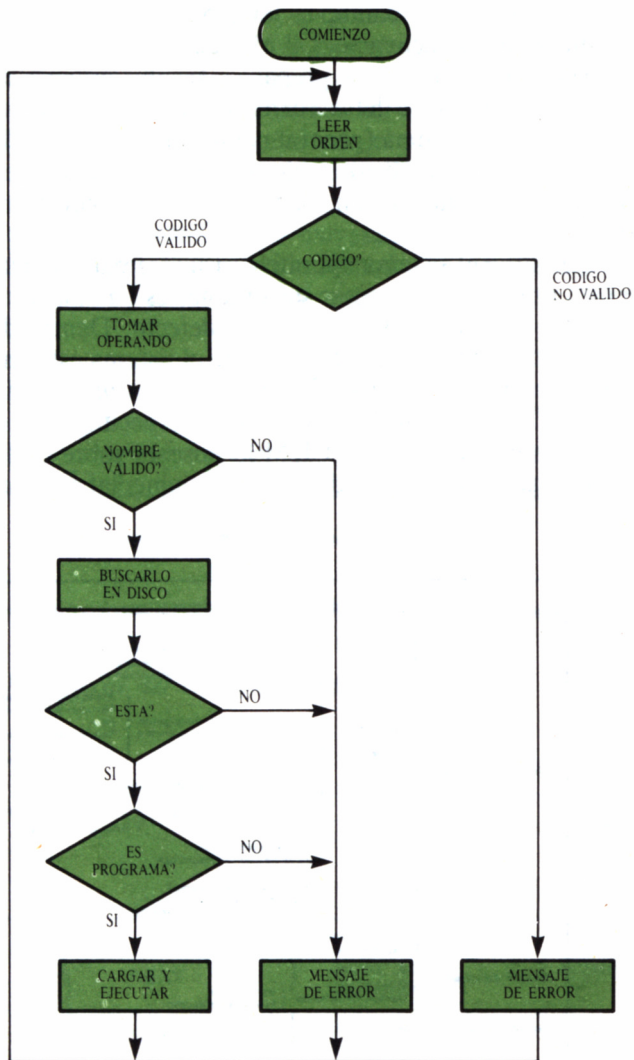
SIMBOLOS COMODIN

Para el caso en que se tenga que someter a una misma operación más de un fichero, CP/M dispone de una función particular que podríamos calificar de «comodín», consistente en utilizar el asterisco (*) o el signo de interrogación (?) dentro de cualquiera de las dos zonas que definen el nombre de un fichero con los siguientes fines:

1) Si se emplea el signo de interrogación, éste sustituye a cualquier carácter en su misma posición. Se pueden utilizar hasta 11 de estos signos, 8 en la zona reservada al nombre y 3 en la de la extensión.

2) Si se emplea el asterisco, sustituye a cualquier grupo de caracteres, y quedarán afectados por la operación todos los ficheros en los que el nombre y la extensión presenten el mismo grupo de caracteres indicado en el nombre tras el comando, antes o después del asterisco.

El CP/M es un sistema operativo utilizable por una única consola (unidad de entrada de comandos) puesto que se trata de un sistema monousuario, pero con poder para comunicarse con varias unidades

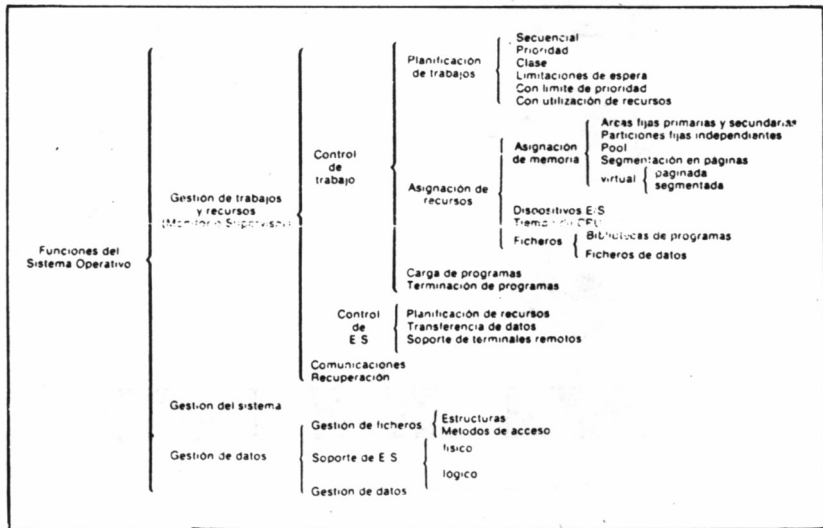


2.3.1. Diagrama de flujo de un programa monitor.

de disco. La conexión con éstas se define mediante una letra mayúscula (de la A a la P) seguida por dos puntos (:). De esta manera, en los equipos AMSTRAD de la serie CPC podemos referirnos desde CP/M a las unidades A: o B:, mientras que en los ordenadores PCW 8256/8512, además se puede gestionar una tercera unidad (el disco virtual), referenciada mediante la letra M:

En el momento de activarse, CP/M muestra en la pantalla el copyright y la versión instalada y se prepara a aceptar órdenes del usuario con la visualización de la letra correspondiente a la unidad que ha sido activada por el sistema (generalmente A). Esta recibe la denominación de unidad «por defecto» y hacia ella o desde ella se establecerá todo el flujo de comunicación hasta que expresamente indiquemos otra cosa.

El símbolo que el sistema operativo visualiza para señalar al usuario que está preparado a aceptar sus órdenes es un «mayor que» (>), precedido del nombre de la unidad por defecto. Dentro de la jerga informática este símbolo indicativo de que el sistema está en condiciones de aceptar órdenes recibe el nombre de *prompt*. Cada vez que una orden se refiere a un fichero y no se especifica la unidad de disco donde éste se encuentra, CP/M lo buscará sobre el disco situado en la unidad activa, procediendo a su inmediata ejecución si lo

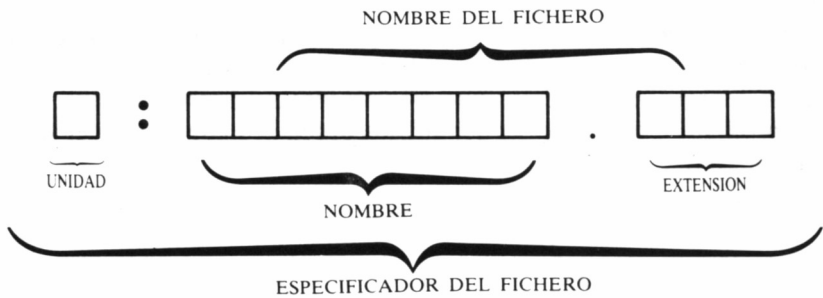


2.3.2. Tabla de funciones del sistema operativo.

encontrara, o emitiendo el correspondiente mensaje de error (de distinta naturaleza según el tipo de comando introducido).

Para acceder a ficheros almacenados en otras unidades diferentes a la actual, será preciso anteponer al nombre de éste la letra correspondiente a la unidad deseada seguida de dos puntos. Si se quiere utilizar como unidad por defecto, se escribirá el nombre de esta seguida de dos puntos, tras lo cual en la pantalla aparecerá el prompt precedido del nombre de esta unidad.

Escribiendo en el teclado podemos ver cómo en la pantalla se visualizan los caracteres pulsados, y cómo el ordenador no toma ninguna acción hasta que no se pulse la tecla RETURN o INTRO. Además, una vez aceptada la orden las letras minúsculas son automáticamente transformadas por CP/M a mayúsculas. Sólo entonces el sistema operativo examina la corrección formal y la consistencia de lo que se ha escrito, y lo lleva a cabo si todo es correcto.



■ 2.3.3. *Sintaxis del especificador de fichero.*

**	{	*.EXT	}	NOM*.EXT NOMBRE.EXT	{ NOMBRE.EXT
	}	NOMBRE.*	}	NOMBRE.E* NOMBRE.EXT	{ NOMBRE.EXT

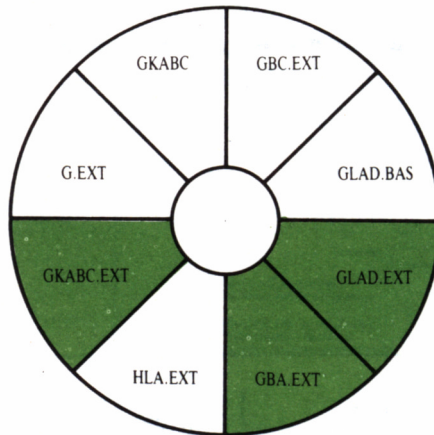
■ 2.3.4. *División por símbolos comodín.*

COMANDOS RESIDENTES Y NO RESIDENTES



Afortunadamente para el usuario, cuando ejecuta un programa no necesita estar preocupándose constantemente de las transferencias de información entre la memoria y los dispositivos periféricos; de todo el trabajo «sucio» se encarga el sistema operativo. Imaginemos la enorme complejidad que supondría llevar a buen fin cualquier tarea que le encomendáramos al ordenador, si a cada instante necesitará controlar uno a uno los diferentes contenidos de cada celda de memoria.

Para evitarnos todo este molesto trabajo, el funcionamiento del ordenador está controlado por una serie de programas, por supuesto formados por secuencias de instrucciones en lenguaje máquina, los cuales van interviniendo en el momento oportuno para gestionar la labor específica que tienen encomendada.

Sobre el disco, los programas se almacenan en forma de ficheros y una vez cargados se alojan en secciones de la memoria reservadas por el sistema operativo, sin que la integridad de éste corra peligro de sobrescribirse o destruirse. Puesto que la misión principal de CP/M consiste en facilitar la ejecución de los programas del usuario y ayudarle a organizar los datos que éstos manejan, dispone de una serie de programas especiales denominados órdenes o comandos del sistema operativo.



A> ERASE G?A*.EXT

 BORRADOS
 NO BORRADOS

2.3.5. Borrado de un disco.

En virtud del sistema con el que la orden se ejecuta, podemos distinguir dos tipos de comandos: residentes y no residentes (también denominados órdenes transitorias, ayudas o utilidades).

Los comandos residentes son parte integrante del CP/M y se hallan en la sección del sistema operativo inicialmente cargada de memoria. Siempre están disponibles, y por tanto, podemos hacer uso de ellos en cualquier momento que los necesitemos.

Los no residentes están en los discos del sistema como ficheros de extensión **.COM** y cuando son requeridos se cargan en una zona de memoria llamada TPA (*Transient Program Area*, Área de programas transitorios), sección de la RAM reservada para los programas ejecutados bajo el control de CP/M.

Los comandos no residentes son verdaderos programas, y para ejecutarlos es suficiente especificar tras el prompt de CP/M, su nombre. Si no están presentes sobre el disco es imposible su ejecución; además, la llamada a uno de ellos elimina previamente, el que pudiera estar antes que él en la TPA.

Para terminar hay que indicar que si en la línea de órdenes introducimos un comando no existente al haber tecleado su nombre erró-

■ 2.3.6. Pantalla inicial
CP/M CPC.

```
CP/M Plus Amstrad Consumer Electronics plc
v 1.2, 61K TPA, 1 disco, 112K disco M:
A>█
```

■ 2.3.7. Pantalla inicial
CP/M PCW.

```
CP/M Plus Amstrad Consumer Electronics plc
v 1.0, 61K TPA, 1 disc drive
A>█
```

■ 2.3.8. Proceso de carga
de un programa.

```
CP/M Plus Amstrad Consumer Electronics plc
v 1.2, 61K TPA, 1 disco, 112K disco M:
A>basic
BASIC?
A>basic
Mallard-80 BASIC with Jetsam Version 1.29
(c) Copyright 1984 Locomotive Software Ltd
All rights reserved
31597 free bytes
Ok
█
```

neamente, CP/M devuelve su nombre en mayúsculas finalizado con el signo de interrogación (?). También podría suceder que el comando no estuviera en el disco situado en la unidad por defecto, y que no le hubiéramos indicado su busca en otra diferente. En este caso el sistema operativo, según el tipo de comando, contesta con la indicación anterior o señalando que para ejecutarlo, necesita encontrar en el disco un determinado programa.

FICHEROS: SI, ¿PERO DONDE?

Otra información importante que necesita CP/M para encontrar un fichero es conocer dónde lo debe buscar. Para ello, deberemos indicarle qué unidad alberga el diskette que lo almacena, el nombre del fichero, y la extensión de éste. Estos tres elementos, la letra identificadora de la unidad, el nombre del fichero y su extensión, reciben la denominación de especificadores del fichero.

La letra de identificación de la unidad y los dos puntos que la deben acompañar se llaman especificador de la unidad. El nombre del archivo y la extensión deben seguir inmediatamente al especificador de la unidad, no estando permitidos los espacios en blanco entre los tres elementos.

Supongamos, por ejemplo, que tenemos en la unidad B el disco del sistema (el que contiene los principales ficheros de CP/M). Si en la boca A está un diskette cuyos ficheros queremos borrar, el comando correcto sería:

```
A>b:erase a:*.*
```

la cual señala a CP/M que debe buscar el fichero ERASE en la unidad B, mientras que lo que pretendemos realizar es eliminar todos los ficheros del diskette alojado en la A.

EXTENSIONES DE LOS FICHEROS

Como describíamos anteriormente, el nombre de cualquier fichero puede ir seguido (y separados ambos por un punto) por otra cadena corta denominada extensión. Con precisión: la extensión comienza por un punto, y continúa con uno, dos o tres caracteres, los cuales deben seguir inmediatamente al nombre del fichero.

Cuando el nombre del fichero va seguido por una extensión es imprescindible, salvo cuando éste sea un programa directamente eje-

cutable por CP/M (en seguida veremos qué ficheros cumplen esta condición), hacer referencia a ambas partes del nombre para que el sistema operativo lo pueda reconocer. Por ejemplo, supongamos que en el disco tenemos almacenado un fichero denominado XYZ.ABC y le ordenamos al sistema operativo que lo elimine mediante la orden ERASE; si tecleamos

ERASE xyz

el sistema operativo responde *No file*, señalando que no lo consiguió encontrar. Sin embargo, con ERA xyz.abc, automáticamente desaparecerá del directorio.

Las extensiones normalmente atribuidas a los ficheros durante la utilización de CP/M, dado que solamente algunas funciones están permitidas si estos tres caracteres responden a unas especificaciones concretas, son entre otras las siguientes:

COM: es la extensión que debe tener un fichero para considerarse directamente ejecutable por CP/M. Es el caso, por ejemplo, de DISCKIT3 o DISCKIT, utilidades para el formateado, copia y verificación de diskettes.

SUB: es la adecuada en aquellos ficheros creados por CP/M para ser ejecutados en proceso batch por SUBMIT.

BAS: extensión de los programas creados mediante órdenes de BASIC.

ASM: los programas fuente que han de ser sometidos posteriormente al ensamblado mediante las utilidades de CP/M, deben tener esta extensión.

\$\$\$: es la extensión de los ficheros temporales creados por comandos como ED o SUBMIT.

```
A>dir
A: C10CPM3 EMS : BANKMAN BAS : PROFILE ENG : SUBMIT COM : SETKEYS COM
A: KEYS CCP : LANGUAGE COM : SET24X80 COM : PALETTE COM : SETSIO COM
A: SETLST COM : DISCKIT3 COM : DATE COM : DEVICE COM : DIR COM
A: ED COM : ERASE COM : GET COM : PIP COM : PUT COM
A: RENAME COM : SHOW COM : TYPE COM : SET COM : SETDEF COM
A: AMSDOS COM : BANKMAN BIN : KEYS WP
```

2.3.9. Directorio CPC.

```
dir
A: J12SCPM3 EMS : BASIC COM : DIR COM : ED COM : ERASE COM
A: KEYS WP : LANGUAGE COM : PALETTE COM : PAPER COM : PIP COM
A: PROFILE ENG : RENAME COM : SET COM : SET24X80 COM : SETDEF COM
A: SETKEYS COM : SETLST COM : SETSIO COM : SHOW COM : SUBMIT COM
A: TYPE COM : RPED BAS : RPED SUB : DISCKIT COM
```

2.3.10. Directorio PCW.

EJERCICIOS

21. ¿Cuál es la estructura básica de un fichero?
 - A) El diskette.
 - B) Los registros.
 - C) El DNA.
22. ¿Cuál es el tamaño de un registro lógico?
 - A) Dos palmos.
 - B) 128 bytes.
 - C) 128 Kbytes.
23. ¿Qué dos zonas comprende la denominación de un fichero?
 - A) Nombre y longitud.
 - B) Fichero del Norte y Fichero del Sur.
 - C) Nombre y extensión.
24. ¿Qué longitud puede tener el nombre de un fichero?
 - A) Infinita.
 - B) 11 caracteres.
 - C) 8 caracteres.
25. ¿Y su extensión?
 - A) Ingordita.
 - B) 3 caracteres.
 - C) 8 caracteres.
26. ¿Es necesario especificar una extensión determinada en un fichero de datos?
 - A) Sí.
 - B) Normalmente, no.
 - C) Los ficheros de datos sólo se definen a ratos.
27. ¿Qué función desempeñan los símbolos comodín?
 - A) Conseguir Repoker cuando se tienen cuatro cartas iguales.
 - B) Sustituir a uno o más caracteres dentro del nombre o extensión de un fichero.
 - C) Sustituir uno o varios datos almacenados en un determinado fichero.

28. ¿Cuál de las siguientes afirmaciones es cierta?
- A) El CP/M es un sistema operativo monousuario.
 - B) El CP/M es un sistema operativo multipuesto.
 - C) Reagan y Gorbachov juegan a los símbolos comodín.
29. ¿Cómo se identifica una unidad de disco?
- A) Por su aspecto.
 - B) Mediante una letra seguida de dos puntos (:).
 - C) Mediante un número seguido por dos puntos (:).
30. ¿Qué es el *prompt* de CP/M?
- A) Es el signo que muestra el sistema operativo indicando que está dispuesto para recibir órdenes.
 - B) Es demasiado «pronto» para saberlo.
 - C) Un programa de utilidad para clasificar datos.
31. ¿A qué se denomina unidad por defecto?
- A) A la unidad A.
 - B) Aquella que se encuentra averiada.
 - C) Aquella a la que se dirige todo el flujo de información si no se especifica otra unidad.
32. ¿Cómo se accede a un fichero situado en una unidad diferente a la de defecto?
- A) Al ser un sistema monousuario, en CP/M no se puede efectuar esta operación.
 - B) Cambiando el disco de unidad.
 - C) Anteponiendo al nombre de éste el identificativo de la unidad seguido de dos puntos (:).
33. ¿Qué tipos de comandos son ejecutables por CP/M?
- A) Comando suicida y Comando-G.
 - B) BASIC y CP/M.
 - C) Residentes y transitorios.

34. ¿Qué extensión deben tener los programas directamente ejecutables por CP/M?

- A) Corta.
- B) BAS.
- C) COM.

35. ¿En qué zona se cargan los comandos transitorios?

- A) En la ATP (Asociación de Tenistas Profesionales).
- B) En la TPA (*Transient Program Area*).
- C) No pasan a la memoria y son directamente ejecutados sobre el disco.

ANATOMIA DE CP/M

E

l CP/M opera dividiendo la memoria central del ordenador en una serie de zonas encargadas de albergar los parámetros y datos necesarios para la correcta ejecución de un programa. Analicemos las áreas principales:

— La página 0: comprende de la posición 0 a la FFh y entre ambas están contenidos unos parámetros fijos a los que el sistema hace referencia para el control de las interconexiones entre los diferentes módulos que componen el CP/M.

— TPA (Transient Program Area, área de programas transitorios): Es el área RAM en que son cargados los programas a ejecutar, tanto las utilidades del sistema operativo, como los del usuario, y su dimensión varía según la versión instalada. Parte de 100h (TBASE) y se extiende hasta CBASE. En el caso de CP/M Plus son 61 Kbytes los disponibles para alojar los programas de ayuda y utilidades que el usuario precise.

— CCP (Console Command Processor, procesador de comandos de la consola): Se extiende de CBASE a FBASE (ver figura). En esta zona están contenidos los comandos que interactúan con el usuario. El CCP es responsable del diálogo y de las comunicaciones entre usuario y sistema operativo, e incluye las órdenes residentes DIR, ERA, REN, SAVE, TYPE o USER.

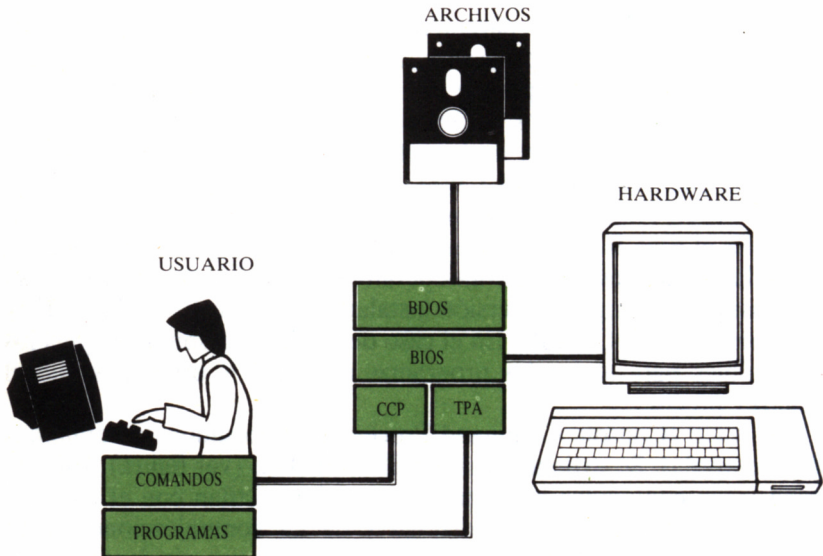
— **FDOS** (File Disk Operating System, sistema operativo de ficheros en disco): Se extiende desde FBASE hasta el final de la memoria (FFFFh). Contiene las rutinas que el ordenador necesita para ejecutar todas aquellas tareas que impliquen transferencias de información desde o hacia el disco. Está compuesto por dos zonas diferentes:

— **BDOS** (Basic Disk Operating System, sistema operativo básico de disco): Constituye el conjunto de las rutinas que se ocupan por completo del control de la unidad de disco.

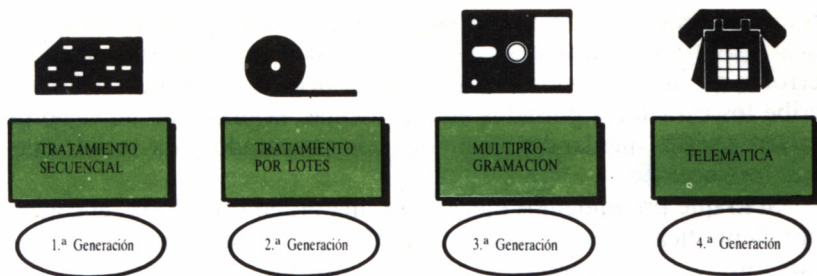
— **BIOS** (Basic Input Output System, sistema básico de entrada y salida): Formado por el conjunto de subrutinas que realizan la conexión entre las funciones del CP/M y las características físicas del ordenador al servicio del CCP, controlando los periféricos enlazados excepto la unidad de disco.

¿COMO FUNCIONA CP/M?

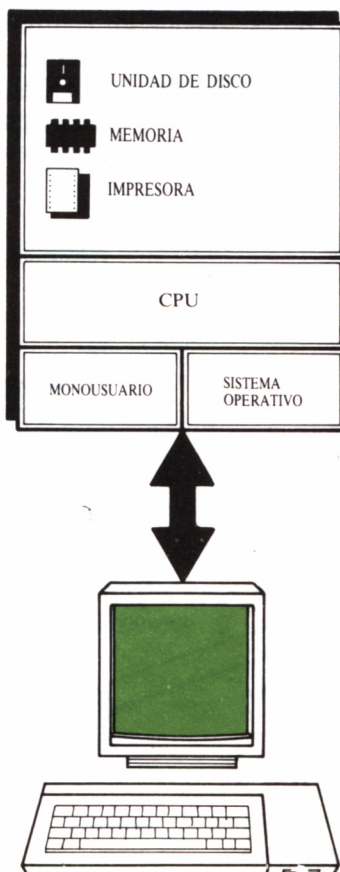
Cuando aparece el prompt de CP/M indicando que el sistema operativo está listo para aceptar órdenes, el CCP es el módulo activo.



2.4.1. Los módulos de CP/M.



2.4.2. Evolución de los sistemas operativos.



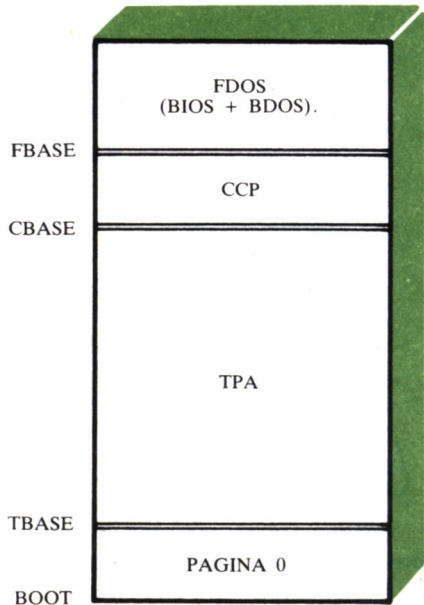
2.4.3. El sistema operativo monousuario.

Examina todo lo que el usuario teclea y ejecuta los comandos requeridos. Como ya hemos descrito, si una orden no es reconocida (por errores de sintaxis o por inexistencia del fichero .COM) el CCP reescribe los caracteres pulsados en mayúsculas, seguidos de un signo de interrogación, quedando a continuación preparado para recibir nuevas instrucciones.

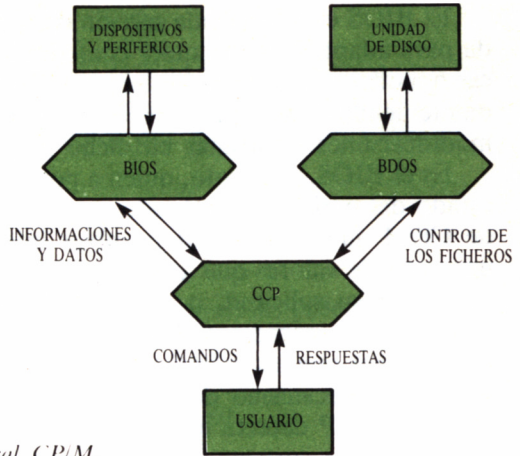
Si lo que pretendemos ejecutar es una orden transitoria, el correspondiente fichero es cargado en la TPA (área de memoria destinada a este fin) y seguidamente, si todo es correcto, es procesada.

Durante el proceso de cualquier orden que implique el manejo de un fichero en disco, el CCP se sirve de un bloque consistente en una secuencia de 36 caracteres contenidos en la página 0, los cuales reúnen información de diverso tipo sobre el fichero afectado (nombre, extensión, atributos, etc.). Esta zona recibe el nombre de FCB (File Control Block, bloque de control de fichero), y para su gestión, el CCP requiere la intervención del BDOS.

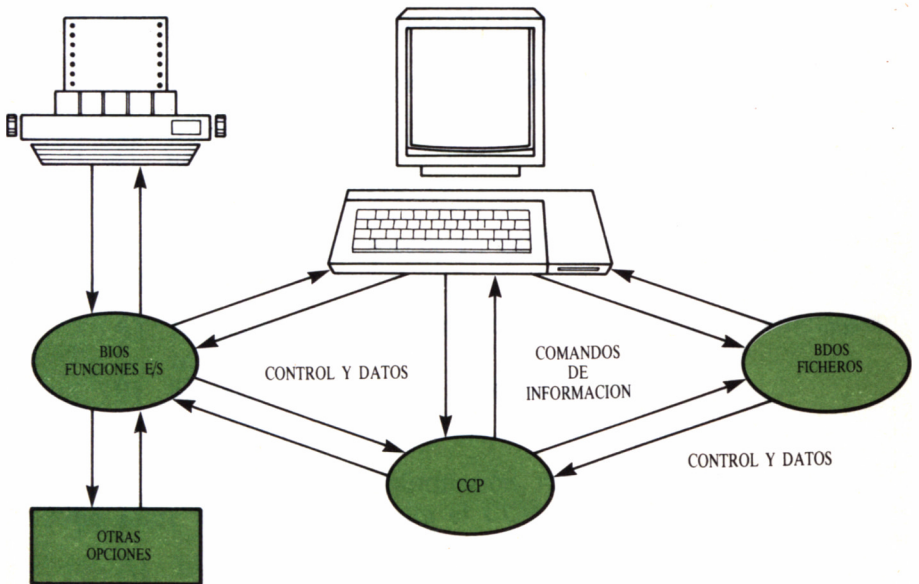
Tanto el CCP como el BDOS son módulos independientes de la estructura física del ordenador, puesto que los parámetros que controlan se sirven del BIOS para determinar la configuración hardware con que trabajan.



2.4.4. Mapa de memoria CP/M.



2.4.5. Estructura general CP/M.



2.4.6. Las partes de CP/M.

El BIOS es la conocida sección del CP/M que puede ser modificada para adaptarla a cualquier configuración entre las diferentes marcas o modelos de ordenador que trabajan bajo este sistema operativo, manteniéndose de esta forma la compatibilidad con sólo realizar las modificaciones oportunas en dicho módulo.

Es el BIOS, por tanto, quien a petición del CCP, informa sobre el estado del periférico implicado en determinada operación al módulo de CP/M que necesite intercambiar información con él, enviándole una serie de valores que le eviten tener que plantearse cómo reaccionará ante su solicitud, o cuál es su forma de trabajo.

Asimismo, representa la estructura que lleva consigo todas las informaciones que los módulos del CP/M intercambian entre sí. En definitiva el mecanismo de ejecución de una orden es el siguiente:

- El CCP detecta que el usuario pretende ejecutar una orden.
- Si no es correcta la rechaza, quedando nuevamente a la espera de recibir otra.
- Si es un comando residente lo ejecuta.
- Si es un comando no residente carga el fichero .COM en la TPA.
- Solicita la intervención del BDOS para el control de los ficheros implicados.
- El BIOS interviene para efectuar las oportunas conexiones con los periféricos.

EJERCICIOS

36. ¿En qué zona de memoria se almacenan los parámetros fijos a los que el sistema hace referencia para el control de las interconexiones entre los diferentes módulos que componen el CP/M?

- A) El BIOS.
- B) La página cero.
- C) El subconsciente.

37. ¿En qué módulo de CP/M se incluyen las órdenes residentes DIR o ERA?

- A) CCP (*Console Command Processor*).
- B) BIOS.
- C) En el módulo lunar.

38. ¿Varía la longitud de TPA según la versión de CP/M instalada?

- A) No.
- B) A medias.
- C) Sí.

39. ¿Qué zonas comprende el FDOS?

- A) BDOS y BIOS.
- B) BDOS y BEOS.
- C) DDOS y LERDOS.

40. ¿Qué es lo que sucede en CP/M si introducimos una orden no reconocida?

- A) Se escribe la orden en mayúsculas, seguida de un interrogante.
- B) Simplemente vuelve a aparecer el *prompt*.
- C) Se indigna.

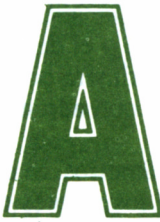
41. ¿Qué bloque de información, reuniendo datos de diverso tipo sobre un fichero, necesita el CCP para la carga del mismo?

- A) El FCB (*File Control Block*).
- B) El de Política Internacional.
- C) EL BDOS.

42. ¿Qué módulo de CP/M se altera para posibilitar la adaptación del sistema a distintas configuraciones hardware?

- A) El CCP.
- B) El BIOS.
- C) El que se indignaba, porque como se ha podido comprobar es propenso a alterarse.

A FONDO



unque no es el objetivo principal del libro, durante este capítulo analizaremos algunas de las rutinas principales que componen los diferentes módulos de CP/M, para que todo aquel lector que pretenda mediante el empleo de código máquina mejorar la operatividad del sistema, disponga de un punto de partida desde el que profundizar en el sistema operativo de su AMSTRAD. Los lectores sólo interesados en conocer las órdenes de CP/M, pueden saltarse el capítulo, sin temor a perder la continuidad con el resto del libro.

EL BIOS

El BIOS, sistema operativo básico de disco, está compuesto por una serie de subrutinas que pueden ser requeridas por varias órdenes del procesador de comandos de la consola (CCP), y que proporcionan como respuesta valores relacionados con el estado de los periféricos asociados al proceso en cuestión. Estas subrutinas utilizan unas tablas que describen las características del hardware de cada modelo de ordenador. Por tanto, con sólo modificar dichas tablas podremos obte-

ner una nueva versión del CP/M apta para otro diseño hardware. Es aquí donde reside la excelente compatibilidad entre todo el software escrito para ordenadores que trabajan bajo CP/M.

Podemos considerar este módulo dividido en cuatro grandes sectores:

- Interface para BDOS y los programas escritos bajo CP/M.
- Interface para los medios de almacenamiento exteriores al sistema.
- Entrada y salida de información a través de los periféricos controlados por BDOS.
- Tampón para los datos, los cuales una vez almacenados en esta zona de memoria, son puestos a disposición del sistema en el momento oportuno.

La llamada de rutinas del BIOS se efectúa a través de una tabla de saltos con un orden preestablecido. Esta puede desplazarse en la memoria, pero la dirección 0000h de la página 0 contiene un salto al vector de arranque en caliente del BIOS (el segundo de la tabla). Por ello, es fácil determinar en todo momento la dirección inicial.

Empleando el desensamblador de CP/M Plus (SID) o el de CP/M 2.2 (DDT), averiguaremos la dirección de salto de la primera posición de la página básica; descontando tres a esta dirección, obtendremos la de inicio de la tabla de saltos del BIOS.

Las rutinas del BIOS pueden enviar y aceptar parámetros. Si éstos son suministrados por el usuario, deben depositarse los valores de 8 bits en el registro C, mientras que si se trata de cantidades de 16 bits, se empleará el par BC. Los valores devueltos son recogidos en el acumulador en caso de datos de 8 bits, o en el registro doble HL cuando se trata de 16 bits.

LA PAGINA 0, LA TPA Y EL CCP

En las direcciones de la página 0, o página básica del sistema están contenidas funciones y datos encargados de realizar y servir de soporte para tareas específicas. Veamos las de mayor interés.

— En la posición 0 está almacenada una instrucción que efectúa un salto absoluto a la rutina del BIOS que realiza la puesta en marcha del sistema operativo. Cualquier programa que incorpore entre sus instrucciones una llamada a la dirección 0000h accederá a esta operación del CP/M.

— La localización 4h almacena la indicación de disco seleccionado

por defecto. Mantiene el número de unidad menos uno; por ejemplo, cuando la actual es la A, contiene un 0, si es la B, un 1, y cuando se trata del disco virtual del PCW 8256/8512 (M), el valor almacenado es un 12.

— La dirección 5Ch (y siguientes hasta un total de 36 posiciones) contiene el bloque de control del fichero (FCB) que maneja el CCP.

— La localización 80h (y siguientes hasta un total de 128 posiciones) constituye un área denominada memoria de acceso directo (DMA o *Direct Memory Access*) empleada por el sistema operativo como buffer de tránsito para los movimientos de datos disco-memoria.

El procesador de comandos de la consola, CCP, se carga inmediatamente por encima del área destinada a conservar los programas a ejecutar (TPA). Pero una vez en marcha un programa, su misión finaliza.

El CP/M admite que un programa de grandes dimensiones ocupe además de la TPA, parte del área del CCP, siendo posible introducir en un programa un salto a la rutina de arranque del sistema (0000h) que efectúe la reinicialización de éste y permita al usuario dialogar de nuevo con el CCP.

EL BDOS

La forma en la cual las informaciones se graban sobre el disco requiere un eficaz control de los ficheros, puesto que cada uno está distribuido en pistas concéntricas subdivididas a su vez en sectores contiguos. Para ello, el sistema operativo necesita conocer los enlaces entre pistas y sectores para determinar cuáles pertenecen o no a un programa determinado.

Para ello, a cada fichero se le asocia un mapa que defina únicamente la posición física de los datos sobre el disco. Además, dispone de otra zona informativa llamada FCB (File Control Block) que se conserva en las primeras pistas del disco inmediatamente después de las destinadas al sistema operativo.

Cuando éste necesita acceder a un fichero, reserva en la TPA el espacio suficiente para almacenar el correspondiente bloque de control del fichero (FCB), con todos los datos que le serán precisos. Al finalizar todas las operaciones, la versión actualizada del FCB almacenada en la TPA se duplica sobre el disco, siendo ésta, a partir de ese momento, el nuevo bloque informativo al cual efectuará referencia el sistema operativo cuando necesite acceder de nuevo al fichero.

El BDOS está compuesto de una serie de subrutinas que posibilitan la gestión del disco y de otros dispositivos periféricos. Para efectuar una llamada a cualquiera de ellas se siguen siempre las siguientes normas:

— En el registro C se almacena el número o código de operación seleccionado.

— Si la rutina precisa de datos de entrada, se le comunican en el registro E, o si se trata de valores de 16 bits, en el par DE.

— A continuación se efectúa un salto a la dirección 0005h de la página 0, la cual posibilita la entrada a BDOS, el cual calcula la dirección de la rutina seleccionada mediante el valor comunicado en el registro C.

— Si BDOS retorna datos hacia el programa principal, son recogidos los de 8 bits en el acumulador, mientras que los de 16 se almacenan en el par HL.

En las tablas que acompañan al capítulo están recogidas las principales entradas de la tabla de saltos de BIOS, así como las rutinas de mayor utilización de BDOS.

TABLA BDOS

Cod.	Valor en C (hexadec)	Objetivo de la función	Datos necesarios	Datos proporcionados
0	0	Puesta en marcha del sistema.	—	—
1	01	Lectura de datos desde la consola. Lee desde CON:	—	A=Car.ASCII
2	02	Escritura de datos sobre la consola. Escribe sobre CON:	E=Car.ASCII	—
3	03	Lectura desde el lector. Lee desde RDR:	—	A=Car.ASCII
4	04	Escritura sobre la perforadora. Escribe sobre PUN:	E=Car.ASCII	—
5	05	Escritura sobre la impresora. Escribe sobre LST:	E=Car.ASCII	—
6	06	Operaciones de I/O sobre la consola (lee un carácter y el estado de la consola o envía un carácter).	E=FFH E=FFH E=Car.ASCII	A=Car.ASCII A=Estado.

Cod.	Valor en C (hexadec)	Objetivo de la función	Datos necesarios	Datos proporcionados
7	07	Lee el estado de I/O.	—	A=byte I/O
8	08	Asigna el estado de I/O.	E=byt I/O	—
9	09	Imprime un buffer desde una dirección = hasta un \$.	DE=Dirección	—
10	0A	Llena un buffer (a partir de la dirección) con caracteres desde CON:	DE=Dirección	Buffer lleno
11	0B	Lee el estado de CON: si está listo un carácter devuelve FFH, si no 0H.	—	A=FFH/0H
12	0C	Número de la versión	—	HL=Número versión.
13	0D	Inicialización del Disco del Sistema (permite el cambio de los discos).		
14	0E	Selección de un disco.	E=Núm. Disco	—
15	0F	Apertura de un fichero.	DE=Dirección del FCB.	A=Código del directorio.
16	10	Cierre de un fichero.	DE=Dirección del FCB.	A=Código del directorio.
17	11	Búsqueda de un fichero.	DE=Dirección del FCB.	A=Código del directorio.
18	12	Búsqueda de la siguiente ocurrencia de un fichero.	—	A=Código del directorio.
19	13	Borrado de un fichero.	DE=Dirección del FCB.	A=Código del directorio.
20	14	Lectura secuencial de un fichero.	DE=Dirección del FCB.	A=Código de error. DMA=Registro.
21	15	Escritura secuencial en un fichero.	DE=Dirección del FCB.	A=Código de error. DMA=Registro.
22	16	Creación de un fichero.	DE=Dirección del FCB.	A=Código de error.
23	17	Cambio de nombre de un fichero.	DE=Dirección del FCB.	A=Código de error.
24	18	Vector de las unidades de disco activos.	—	HL=Vector.
25	19	Unidad seleccionada.	—	A=Unidad sel.
26	1A	Modificación del DMA.	DE=Nuevo DMA.	—
27	1B	Vector de posición del espacio ocupado.	—	HL=Dirección vector.
28	1C	Protección del disco sel. contra la escritura.	—	—

Cod.	Valor en C (hexadec)	Objetivo de la función	Datos necesarios	Datos proporcionados
29	1D	Vector de los discos protegidos contra la escritura.	—	HL=Vector.
30	1E	Modificación de los atributos de un fichero.	DE=Dirección del FCB.	—
31	1F	Parámetros del disco.	—	HL=Dirección del BPD.
32	20	Control del Código de Usuario	E=FFH E=Nuevo Cód. Usuario.	A=Cód. Usuario. —
33	21	Lectura en un fichero de acceso directo.	DE=Dirección del FCB.	A=Cód. de error. DMA=Registro.
34	22	Escritura en un fichero de acceso directo.	DE=Dirección del FCB.	A=Cód. de error. DMA=Registro.
35	23	Cálculo de la dimensión virtual de un fichero.	DE=Dirección del FCB.	Bytes 34-36 del FCB=Dirección registro que sigue a EOF (final del fichero).
36	24	Número del registro en un fichero de acceso secuencial.	DE=Dirección=del FCB.	Bytes 34-36 del FCB=Posición.
37	25	Inicialización de las unidades.	DE=Vector de las unidades.	A=OOH.
40	28	Escritura en un fichero de acceso directo (llena el registro con ceros antes de escribir)	DE=Dirección del FCB.	A=Código de error. DMA=registro.

FILE CONTROL BLOCK

N.º orden	N.º bytes	Significado
1	1	Código de la unidad (A-P).
2-9	8	Nombre del fichero (en letras mayúsculas).
10-12	3	Extensión del fichero (en letras mayúsculas): Si el primer bit del 10 es 1, el fichero es de tipo R/O; si el primer bit del 11 es 1, el fichero es de tipo SYS.
13	1	Valor de Ext; actualizado durante el acceso al fichero.
14	1	Reservado para uso interno del sistema.
15	1	Reservado para uso interno del sistema. Se pone a 0 en las subrutinas de apertura de ficheros.
16	1	Número de registros del actual extent (va de 0 a 127).
17-32	16	Utilizados por el CP/M para el control interno.
33	1	Número del registro a controlar en la siguiente operación de escritura o lectura.
34-35	2	Número de registros en caso de acceso directo (de 0 a 65.535).
36	1	Señal de overflow en caso de acceso directo (o aleatorio).

TABLA DE SALTOS DEL BIOS

00+desp.:	JMP BOOT	;arranque en frío
03+desp.:	JMP WBOOT	;arranque en caliente
06+desp.:	JMP CONST	;consulta estado consola
09+desp.:	JMP CONIN	;entrada de consola
0C+desp.:	JMP CONOUT	;salida por consola
0F+desp.:	JMP LIST	;salida por impresora
12+desp.:	JMP PUNCH	;perforador de cinta
15+desp.:	JMP READER	;lector de cinta perforada
18+desp.:	JMP HOME	;cabezal a pista 0
1B+desp.:	JMP SELDSK	;seleccionar unidad
1E+desp.:	JMP SETTRK	;fijar pista
21+desp.:	JMP SETSEC	;seleccionar sector de grabación
24+desp.:	JMP SETDMA	;seleccionar tampón de datos
27+desp.:	JMP READ	;leer sector de grabación
24+desp.:	JMP WRITE	;escribir sector de grabación
2D+desp.:	JMP LISTS	;consulta estado impresora
30+desp.:	JMP SECTRAN	;traducir número de la grabación

ANTES DE SEGUIR



Cualquier disco virgen, antes de poder ser utilizado por el sistema para servir de medio de almacenamiento a cualquier tipo de información, datos, programas, utilidades, etc., debe pasar una operación previa que distribuya el espacio disponible uniformemente, de forma tal que el ordenador pueda encontrar un fichero determinado.

Este proceso, conocido bajo la denominación de formateado, se encarga de dividir el disco en una serie de pistas concéntricas y, cada una de ellas, en un número fijo de sectores. El sector es la unidad mínima de información que puede extraerse o grabarse sobre el disco, y cuando el bloque de información no lo cubre completamente, el sistema operativo es el encargado de rellenar los bytes restantes hasta completarlo. Por el contrario, cuando es mayor, el mismo CP/M es quien se ocupa de gestionar el número de sectores necesarios para su almacenamiento.

Las unidades de disco de los ordenadores AMSTRAD admiten diversas capacidades de almacenamiento una vez formateadas, y según el tipo de formato seleccionado. Mientras que los ordenadores PCW soportan un único tipo, los de la serie CPC están capacitados para formatearlos según tres configuraciones. Para trabajar bajo

CP/M los discos deben encontrarse en formato «SISTEMA» y a él nos referiremos en el resto del libro.

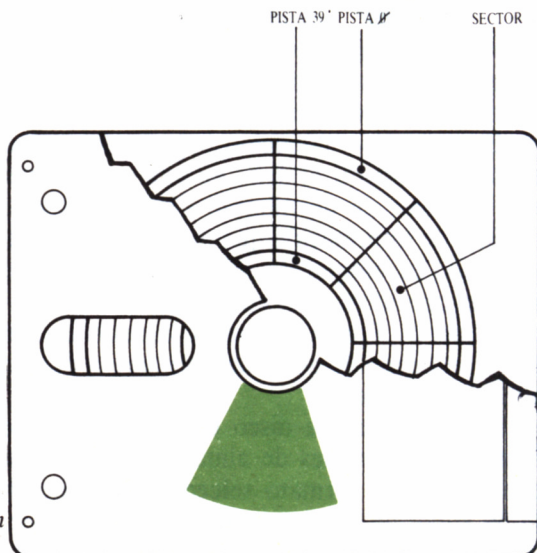
Entre las utilidades de CP/M Plus, DISCKIT3 (CPC) o DISCKIT (PCW), es el encargado de formatear, copiar y verificar discos. Para activarlo, una vez cargado CP/M desde BASIC mediante la orden externa |CPM en los CPC, o simplemente introduciendo el disco del sistema en la unidad A de los PCW al comenzar una sesión de trabajo, se tecldea DISCKIT3 (CPC) o DISCKIT en los PCW.

Seguidamente, aparecen en la pantalla una serie de menús, descritos en las figuras que acompañan al texto, con todas las opciones que pueden tomarse.

Ambos formatos, tanto el de los ordenadores CPC, como el de los PCW, mantienen algunas características comunes:

- 40 pistas por cada cara del disco.
- 8 sectores por pista.
- 512 bytes, longitud del sector.
- 128 bytes longitud del registro lógico.
- 64 reseñas máximo para el directorio.

En el CPC, las dos primeras pistas están reservadas al sistema operativo, y a alojar el directorio del disco, mientras que en los PCW solamente una pista está reservada. De ahí que la capacidad libre para el usuario, una vez formateado el disco, varíe de 169 K en los CPC, a 173 K en los PCW.



2.6.1. Organización de un disco.

2.6.2. Procesos accesibles desde Diskit.

DISC KIT v 1.1
PC8256 y CP/M Plus
© 1985 Amstrad Consumer Electronics plc y Loconotive Software Ltd.

Una unidad instalada

	f 6	Copiar	
	f 5		
	f 4	Inicializar	
	f 3		
Salir del programa	SAL	f 2	Verificar
		f 1	

S	Copiar disco CF2
Cualquier otra tecla para volver al menú	

S	Inicializar disco CF2
Cualquier otra tecla para volver al menú	

S	Verificar disco CF2
Cualquier otra tecla para volver al menú	

DISC KIT v 1.2
PC48256 y CP/M Plus
© 1985 Amstrad Consumer Electronics plc y Locomotive Software Ltd..

Dos unidades instaladas
Extraiga el disco de A:
Pulse cualquier tecla para continuar

Volver al menú principal

SAL

f 4

f 3

Inicializar disco CF2 en A:

f 2

f 1

Inicializar disco CF2DD en B:

S

Para inicializar un disco CF2,
insértelo en A, con la cara
correcta hacia la izquierda, y pulse S

Para cancelar y volver al menú principal,
pulse cualquier otra tecla

S

Para inicializar un
disco CF2DD, inserte en B:
el disco y pulse S

Para cancelar y volver al menú principal,
pulse cualquier otra tecla

Volver al menú principal

SAL

f 4
f 3
f 2
f 1

Leer disco CF2 en A:
Leer disco en B:

S

Para copiar un disco CF2, inserte en A: el disco para LEER, con la cara correcta hacia la izquierda, y pulse S

Para cancelar y volver al menú principal, pulse cualquier otra tecla

Volver al menú principal

SAL

f 4
f 3
f 2
f 1

Escribir sobre disco CF2 en A:
Escribir sobre disco CF2DD en B:

S

Para copiar un disco CF2, inserte en B: el disco para LEER, inserte en A: el disco para ESCRIBIR, y pulse S

Para cancelar y volver al menú principal, pulse cualquier otra tecla

S

Para copiar un disco
CF2DD, inserte en B: el
disco para LEER y pulse S

Para cancelar y volver al menú principal,
pulse cualquier otra tecla

f 4
f 3

Verificar disco CF2 en A:

Volver al menú principal

SAL

f 2
f 1

Verificar disco en B:

S

Para verificar un disco CF2,
insértelo en A: con la cara
correcta hacia la izquierda, y pulse S

Para cancelar y volver al menú principal,
pulse cualquier otra tecla

S

Para verificar un
disco, inserte en B:
el disco y pulse S

Para cancelar y volver al menú principal,
pulse cualquier otra tecla

DISC KIT 1.0
CPC6128 & CP/M Plus
(c) 1985 Amstrad Consumer Electronics plc
and Locomotive Software Ltd.

One drive found ← UN DRIVE

Copy	7		← COPIA
Format	4		← FORMATEADOS
Verify	1		← VERIFICACION
Exit from progra	0		← SALIDA DE PROGRAMA

DISC KIT 1.0
CPC6128 & CP/M Plus
(c) 1985 Amstrad Consumer Electronics plc
and Locomotive Software Ltd.

One drive found ← UN DRIVE

System format		9	← FORMATO SISTEMA
Data format		6	← FORMATO DATA
Vendor format		3	← FORMATO VENDOR
Exit menu		.	← SALIDA A MENU

About to read reserved track
Insert a System disc
Press any key to continue

← PARA LEER LAS PISTAS RESERVADAS
← INSERTE UN DISCO EN SISTEMA
← PULSE UNA TECLA PARA CONTINUAR

Remove disc
Press any key to continue

← QUITE EL DISCO

Y Format as System
Any other key to exit menu

← Y PARA FORMATEADO
TIPO SISTEMA

← CUALQUIER OTRA TECLA
PARA REGRESAR AL MENU

Insert disc to format
Press any key to continue

← INSERTE EL DISCO
A FORMATEAR

Y Copy
Any other key to exit menu

← Y PARA COPIA

Insert disc to READ
Press any key to continue

← INSERTE EL DISCO
PARA LEER

Disc is System format
Insert disc to WRITE
Press any key to continue

← EL DISCO ESTA EN
FORMATO SISTEMA
← INSERTE EL DISCO
PARA ESCRIBIR

Y Verify
Any other key to exit menu

← Y PARA VERIFICACION

Insert disc to verify
Press any key to continue

← INSERTE EL DISCO
PARA VERIFICAR

Insert disc to verify
Press any key to continue

← INSERTE EL DISCO
PARA VERIFICAR

DISC KIT 1.0
CPC6128 & CP/M Plus
(c) 1985 Amstrad Consumer Electronics plc
and Locomotive Software Ltd.

Two drives found

← DOS DRIVES

Copy

7	
---	--

Format

4	
---	--

Verify

1	
---	--

Exit from program

0	
---	--

About to read reserved tracks
Insert a System disc into A:
Press any key to continue

← PARA LEER LAS PISTAS
RESERVADAS
← INSERTE UN DISCO
EN SISTEMA EN A:

Format A:

	8	
--	---	--

Format B:

	5	
--	---	--

Exit menu

	2	
--	---	--

← FORMATEAR A:

← FORMATEAR B:

Y Format A: as System
Any other key to exit menu

← Y PARA FORMATEAR
A: COMO SISTEMA

Y Format B: as System
Any other key to exit menu

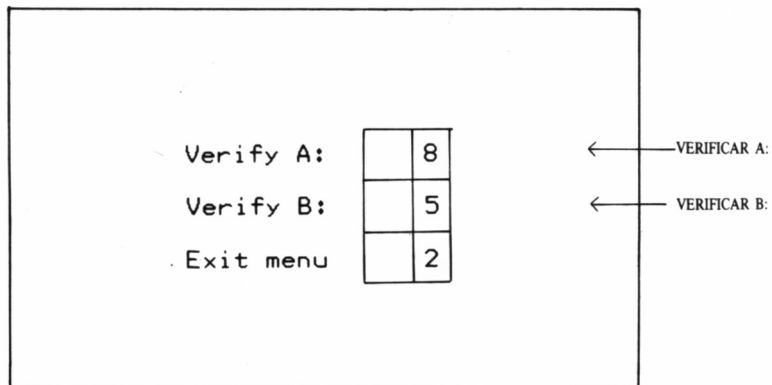
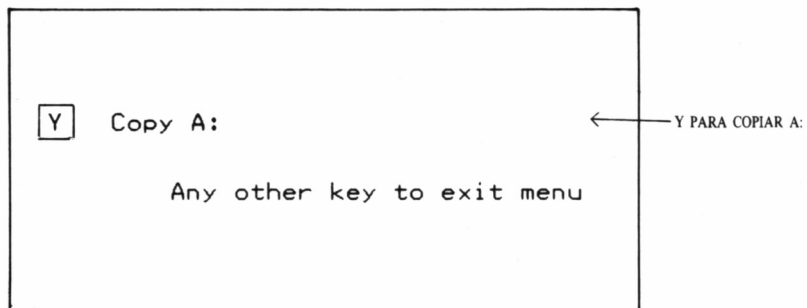
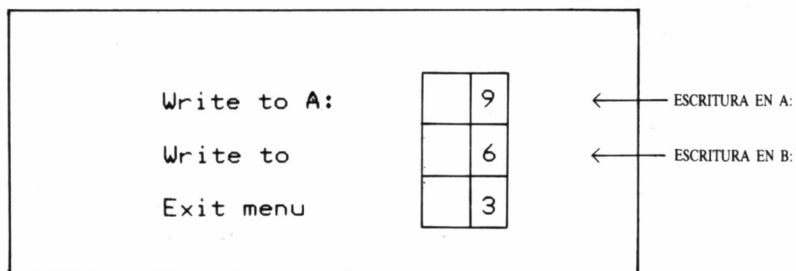
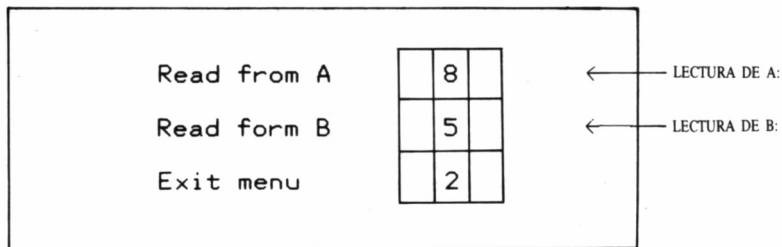
← Y PARA FORMATEAR
B: COMO SISTEMA

Insert disc to format into A:
Press any key to continue

← INSERTE EL DISCO A
FORMATEAR EN A:

Insert disc to format into B:
Press any key to continue

← INSERTE EL DISCO A
FORMATEAR EN B:



Y

Verify A:

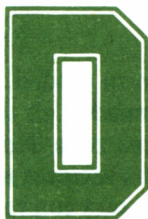
← Y PARA VERIFICAR A:

Any other key to exit menu

Insert disc to verify into A:
Press any key to continue

← INSERTE EL DISCO A
VERIFICAR EN A:

LOS COMANDOS RESIDENTES



entro de las utilidades de CP/M Plus existen una serie de órdenes de empleo directo por el usuario, sin necesidad de recurrir a los ficheros .COM almacenados en los discos del sistema. Reciben la denominación de comandos u órdenes residentes, puesto que una vez completada la inicialización, el sistema operativo las pone sin más a nuestra entera disposición. La razón de esta división de las órdenes estriba en que en las pistas del disco reservadas al sistema operativo, no existe espacio suficiente para albergar las muchas utilidades de que dispone CP/M.

Antes de introducirnos en la descripción tanto de los comandos residentes como de los transitorios u órdenes externas, debemos tener claros algunos conceptos que manejaremos en cada explicación:

— Comando o instrucción es toda aquella sentencia que invita al sistema operativo a llevar alguna opción.

— Parámetro, en general, es el nombre del fichero sobre el cual obligamos al comando que actúe.

— Opción, situada habitualmente entre corchetes ([]), modifica la acción del comando afectado en función de su significado, y como tal opción, puede eliminarse si lo deseamos. Normalmente, cuando la introducción de un parámetro es obligatoria, CP/M se encarga de soliciárselo al usuario.

Al emplear comandos y parámetros, deben separarse mediante un espacio en blanco. Este es el único lugar dentro de una orden completa en el que puede figurar un blanco; ni en el nombre del comando, ni

en la opción, ni en los posibles parámetros, el sistema operativo admite espacios en blanco.

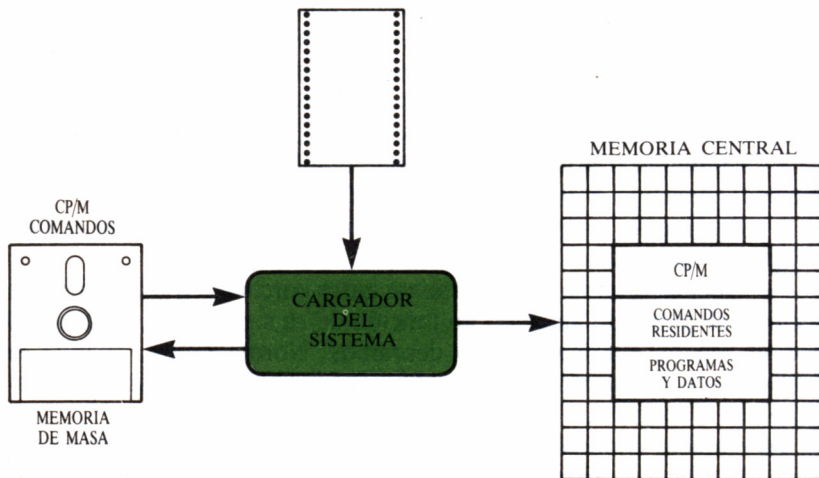
En CP/M 3.0 se dispone de seis comandos residentes: DIR, DIRSYS, ERASE, RENAME, TYPE y USER. El sistema operativo también los interpreta si le indicamos el nombre abreviado (formado por las tres primeras letras de cada uno, a excepción de DIRSYS del cual deben señalarse las cuatro primeras).

A estos comandos podemos añadir un séptimo: el encargado de modificar la unidad de discos utilizada por defecto. Recordemos que para ello basta con señalar tras el prompt de CP/M el nombre de la unidad seleccionada seguido de dos puntos (:).

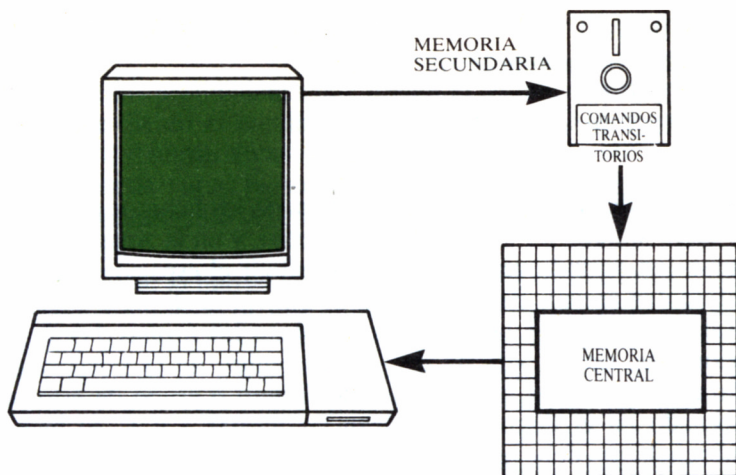
PREGUNTESELO A DIR

Cuando introducimos por primera vez un disco en su unidad, probablemente necesitemos conocer los ficheros almacenados en él. El nombre y extensión de todos ellos está anotado en una zona denominada directorio del disco, situada a continuación de los sectores reservados para el sistema operativo, y para examinarlo bastará con teclear tras el prompt de CP/M la orden DIR.

Supongamos que hemos introducido en la unidad A el disco del sistema operativo. Según la versión de ordenador habremos obtenido en la pantalla una de las dos listas de ficheros siguientes:



2.7.1. Los comandos residentes.



2.7.2. Los comandos transitorios.

Los usuarios de ordenadores de la serie PCW, si efectúan un directorio de la cara 1 de los discos del sistema (el que contiene el procesador de textos Locoscript) comprobarán que al final aparece la indicación SYSTEM FILE (S) EXIST. Esto significa que ciertos ficheros contienen el atributo SYS. Es importante señalar que los ficheros marcados con el atributo SYS, no son visibles al realizar un directorio del disco mediante DIR.

Para averiguar cuáles son éstos, podemos ejecutar la orden DIR-SYS, o abreviadamente, DIRS. Tras ello, el sistema escribirá sobre su monitor la siguiente lista de ficheros:

El comando DIR, por tanto, visualiza los nombres de los ficheros, aunque como más adelante veremos, manejando el fichero DIR.COM puede obtenerse bastante información adicional sobre las características de estos. Admite tres modalidades de ejecutarlo:

DIR

DIRS

DIR «con opciones»

Mientras que DIR y DIRS son órdenes residentes, para manejar DIR «con opciones» será preciso cargar antes en la memoria el programa de utilidad DIR.COM desde el disco. Cuando se trata como comando residente deben respetarse las siguientes reglas sintácticas:

- 1) DIR {d:}
- 2) DIR {nombrefichero}
- 3) DIRS {d:}
- 4) DIRS {nombrefichero}

DIR y DIRSYS muestran los nombres de los ficheros catalogados en el disco situado en la unidad por defecto, y en la zona de usuario en curso (en seguida veremos cómo distribuir el espacio de un disco en diferentes áreas). Para visualizar el directorio de otra unidad sin necesidad de cambiar la actual, se emplean las formas 1 y 3 antes mencionadas, donde «d» ha de ser el nombre de la seleccionada.

Para comprobar si un determinado fichero o grupo de éstos está almacenado en el disco, se emplean las formas 3 y 4. Al hacerlo, «nombrefichero» debe estar formado por las dos zonas identificadoras de éste, nombre y extensión. En estos casos, DIR y DIRS admiten la inclusión de los símbolos comodín (* y ?) dentro del nombre del fichero. Consideremos algunos ejemplos:

A>DIR

Visualiza una lista de todos los ficheros con el atributo DIR dentro de la zona de usuario número 0.

A>DIR b:

Muestra todos los ficheros de atributo DIR en la zona de usuario 0, del disco situado en la unidad B.

5A>DIR M:EJEMPLO.GBA

Contestará con EJEMPLO.GBA si es que un fichero con este nombre está en la unidad M, y en la zona de usuario número 5. En caso contrario, se visualiza No File.

2A>DIR *.BAS

Mostrará una lista de todos los ficheros con extensión .BAS, dentro del área de usuario número 2.

B9>DIR AB*.C?D

Este es un típico ejemplo de manejo de símbolos comodín. Serán impresos todos aquellos ficheros almacenados en el disco situado en la unidad B, y dentro de la zona de usuario 9, cuyas dos primeras letras sean AB, y la extensión comience por C y acabe en D, siendo indiferente el segundo carácter de ésta.

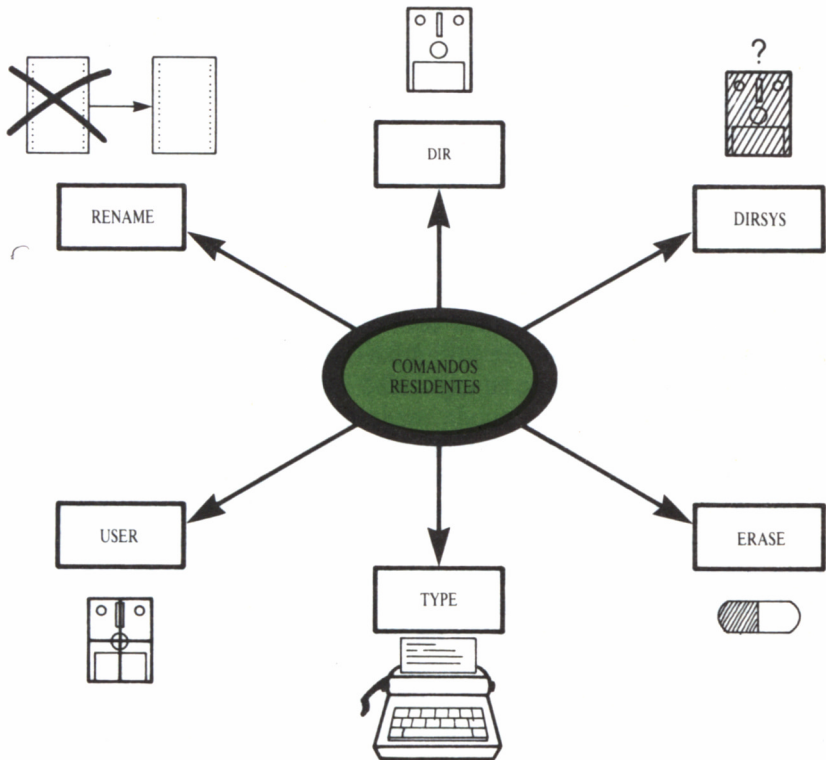
A>DIRS *.COM

Visualiza todos los ficheros de nombre cualquiera, con extensión .COM, situados en la unidad A, zona de usuario 0, pero con el atributo del sistema (SYS).

LAS ZONAS DE USUARIO

Otra de las posibilidades que brinda el sistema operativo al usuario es la de dividir el medio de almacenamiento externo con el cual trabaja en diferentes áreas. Esta distribución puede ser muy útil, en especial cuando varias personas trabajan con los mismos discos y podrían accidentalmente deteriorar trabajos previamente realizados por otro usuario.

En realidad, la mayor utilidad del sistema sólo se consigue cuando el equipo está dotado de una unidad de disco rígido, dado que la gran capacidad de almacenamiento que admite, aconseja una partición en diferentes zonas de trabajo. No obstante, distribuir el espacio de un diskette en diferentes zonas ofrece algunas ventajas: por ejemplo, se puede asignar una zona a los ficheros .BAS creados a partir de BASIC, otra a los textos generados por un procesador (Locoscript o cual-



2.7.3. Utilización de los comandos residentes.

quier otro), pero con la ventaja adicional de poder mantener simultáneamente, en dos zonas de usuario distintas, ficheros con el mismo nombre y extensión.

La orden de CP/M encargada de realizar esta partición es USER y es tratada como comando residente tan solo, ya que por mucho que nos empeñemos en encontrar un fichero USER.COM en los discos del sistema, no lo conseguiremos. Las reglas sintácticas a respetar son de lo más sencillo:

```
..USER {número}
```

donde «número» ha de estar comprendido entre 0 y 15.

Si tecleamos únicamente USER el sistema operativo contesta con la indicación *Enter user number* (introduzca número de usuario). Como hemos puntualizado, deberá ser una cantidad entre 0 y 15. Por ejemplo:

```
A>USER
Enter User#:9 (contestamos pulsando 9+RETURN)
9A>
```

el número de usuario actual pasa a ser el 9 de la unidad A, o directamente

```
A>USER 9
9A>
```

BORRANDO FICHEROS

Cuando manejamos un disco como medio de almacenamiento masivo tanto de datos como de programas, llegará un momento en que éste se llenará, puesto que su capacidad está limitada, y el número de reseñas que el directorio puede mantener no llega más allá de 64. Entonces no quedará otro remedio que eliminar algún fichero que no sea ya de utilidad, liberando el espacio del disco que ocupaba.

ERASE o ERA, abreviadamente, elimina del directorio del disco uno o más ficheros. Para borrar un fichero de la unidad por defecto basta con indicar a continuación del comando su nombre (como siempre formado por nombre más extensión). Estos admiten los símbolos comodín. Aunque su contenido no desaparece físicamente del disco, se debe obrar con precaución puesto que el espacio que ocupaba queda disponible para cualquier otro programa que lo reclame, y a no ser que dispongamos de los suficientes conocimientos de código máquina o poseamos alguna utilidad para recuperarlos los habremos perdido para siempre.

Para eliminar un fichero en otra unidad diferente de la actual, como siempre, no es preciso cambiar a ésta, sino que con señalar en cuál debe buscar CP/M, antes del nombre del archivo, será borrado a continuación.

Existe una tercera posibilidad antes de borrar un fichero consistente en pedirle al sistema operativo la confirmación pertinente. Esta opción sólo está disponible como comando transitorio en el fichero ERASE.COM. Su sintaxis es la siguiente:

```
ERASE {nombrefichero}[CONFIRM]
```

Al seleccionar este formato, el sistema operativo pide que se confirme, uno a uno, cada uno de los ficheros seleccionados para borrar en la orden ERASE. Consideremos el efecto de este mandato con algunos ejemplos.

```
A>ERASE EJEMPLO.BAS
```

Elimina del disco el fichero EJEMPLO.BAS si es que está almacenado en la unidad A.

```
A>ERA *.COM
```

Tras ello, el sistema operativo muestra el mensaje ERASE *.COM (Y/N)? Si contestamos Y, todos los ficheros con extensión .COM del disco situado en la unidad A, serán eliminados. En caso contrario, se aborta la operación.

```
A>ERA B:AB*.* [CONFIRM] o ERA B:AB*.*[C]
```

Cada fichero que el sistema operativo encuentre en la unidad B, con el nombre comenzando por AB y de cualquier extensión será mostrado con la indicación (Y/N)? pidiendo confirmación. Si se contesta afirmativamente (Y) es eliminado, mientras que en caso contrario, el sistema muestra el siguiente que cumpla con la plantilla designada.

```
A>ERA B:*.*
```

El sistema nos avisa con el mensaje Confirm (Y/N)?, para asegurarse que efectivamente el usuario desea llevar a cabo esta operación. Si se contesta afirmativamente, todos los ficheros almacenados en la unidad B, son borrados.

RENOMBRANDO FICHEROS

La orden RENAME permite cambiar el nombre asignado a cualquier fichero almacenado sobre el disco. Puede manejarse en su forma abreviada, REN, y admite los símbolos comodín. Su sintaxis correcta es la siguiente:

RENAME {nuevo-nombrefichero = antiguo-nombrefichero}

Comprobemos su forma de trabajar con algunos ejemplos:

A>RENAME NUEVO.BAS=VIEJO.BAS

El fichero de nombre VIEJO.BAS pasa a llamarse NUEVO.BAS.

A>RENAME

Tras esta orden el sistema visualiza los siguientes mensajes:

Enter New Name: contestamos, por ejemplo, NUEVO.BAS

Enter Old Name: respondemos, por ejemplo, VIEJO.BAS

A>

Acto seguido, como siempre, el fichero VIEJO.BAS pasa a llamarse NUEVO.BAS, aunque esta orden no es tratada como comando residente y será preciso que el fichero RENAME.COM esté sobre la unidad A.

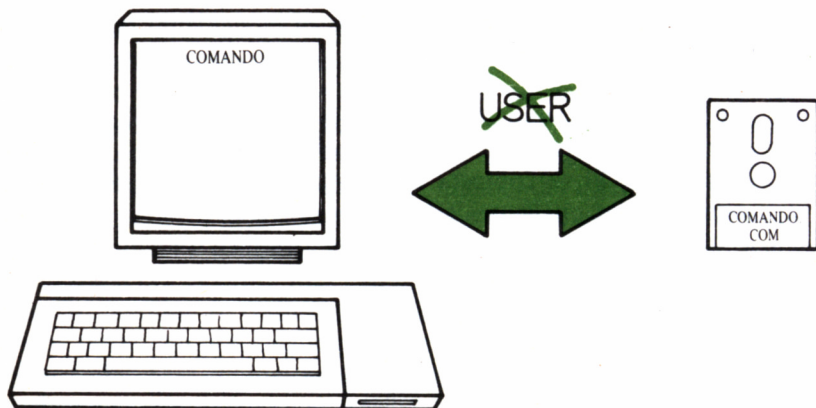
B>REN A:NUEVO.BAS=VIEJO.BAS

Sin necesidad de modificar la unidad por defecto, el fichero almacenado en A como VIEJO.BAS pasa a llamarse NUEVO.BAS. En este caso se trata como orden residente.

A>RENAME J*.XYZ=H*.XYZ

La presente orden modifica los nombres de todos los ficheros almacenados en la unidad A que comiencen por H, a otros nuevos cuya primera letra será J.

A>REN B:NUEVO=B:VIEJO



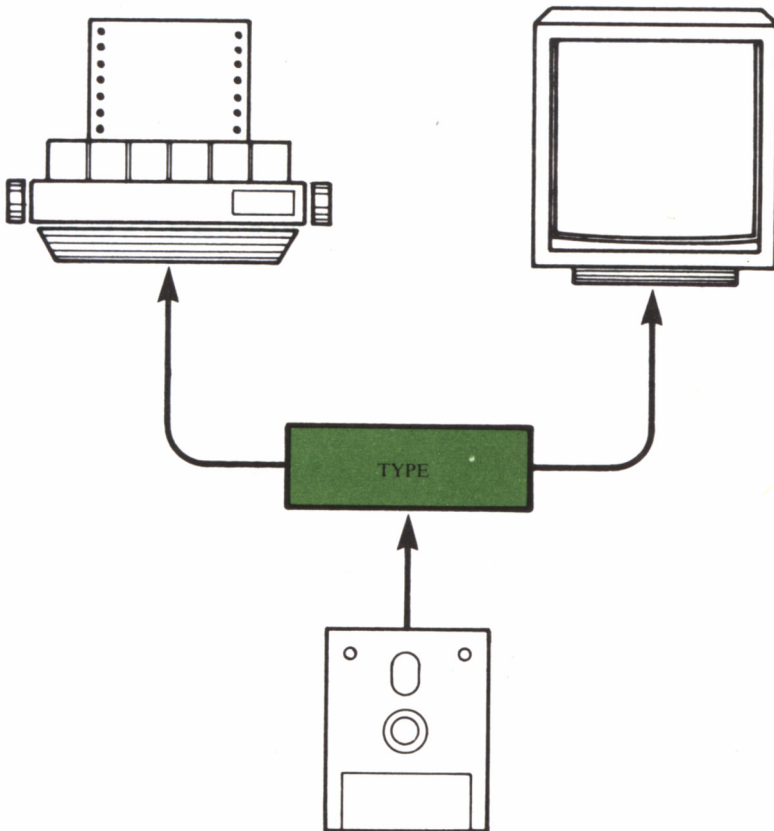
2.7.4. Los comandos .COM.

En este caso, el fichero denominado VIEJO, situado en la unidad B, pasa a llamarse NUEVO. Es importante tener presente que una vez efectuada la alusión tras la orden REN a la unidad B, es innecesario especificarlo en la segunda parte de la instrucción, es decir, se conseguirían idénticos resultados procesando las órdenes siguientes:

A>REN B:NUEVO=VIEJO o A>REN NUEVO=B:VIEJO

¿QUE CONTIENE UN FICHERO ASCII?

El último de los comandos residentes de CP/M 3.0 es TYPE. Su objetivo es mostrar el contenido de un fichero a través de la pantalla



2.7.5. El comando TYPE.

del monitor. Sin embargo, esta orden sólo funcionará correctamente con ficheros que contengan texto, puesto que los códigos de control que pudieran almacenar otros, confundirían al sistema provocando resultados imprevisibles. Su sintaxis completa es la siguiente:

```
TYPE {d:} {nombrefichero} {[PAGE|NOPAGE]}
```

Como siempre, d representa la unidad donde se encuentra el fichero a vaciar. Este, según el modelo de ordenador disponible, es presentado en secuencias de 23 líneas (CPC) o en bloques de 29 (PCW), pidiéndose la pulsación de una tecla para continuar el listado.

Como comando no residente admite dos opciones: PAGE y NOPAGE. La primera actúa de forma análoga a como reacciona TYPE si no señalamos ninguna opción, deteniendo el listado cada cierto número de líneas como antes hemos visto. La segunda provoca que el contenido del fichero completo desfile ante nuestros ojos. No obstante, puede ser interrumpido pulsando CONTROL+S (ALT+S, en los PCW) y reanudado de nuevo con CONTROL+Q (ALT+Q).

El volcado se efectúa a través de la impresora, si previamente la activamos pulsando CONTROL+P (ALT+P). Consideremos los siguientes ejemplos:

```
A>TYPE PROGRAM.TXT
```

Visualiza el contenido del fichero PROGRAM.TXT sobre la pantalla del monitor, siempre y cuando se encuentre en la unidad A.

```
A>TYPE B:PROGRAM.TXT[NOPAGE]
```

Muestra el contenido del fichero PROGRAM.TXT sin interrupción, siempre y cuando esté almacenado en el disco alojado en la unidad B.

EJERCICIOS

43. ¿Qué extensión tienen los denominados comandos residentes?

- A) COM.
- B) Ninguna.
- C) No sólo tienen sufijo, también tienen «sumovil».

44. ¿Cuál es el motivo de que no todas las utilidades de CP/M sean comandos residentes?

- A) Vagancia de Gary Kildall.
- B) Carencia de espacio.
- C) Las características particulares de cada comando.

45. ¿Bajo qué denominación se conocen los comandos no residentes?

- A) Extraordinarios.
- B) Extranjeros.
- C) Transitorios.

46. ¿Cómo se señalan las opciones en un comando CP/M?

- A) Entre corchetes.
- B) Entre paréntesis.
- C) Con el dedo índice.

47. ¿Cuál es la abreviatura del comando DIRSYS?

- A) DIRS
- B) DIR.
- C) DIRectorio únicamente para ficheros del tipo SYStem.

48. ¿Es RENAME un comando residente?

- A) No. Es un comando dos veces NAME.
- B) No. Es transitorio.
- C) Sí.

49. Entre las órdenes DIR y DIRSYS ¿Con cuál de las dos aparecen todos los ficheros del disco, tanto los DIR como los SYS?

- A) Con ninguna.
- B) Con DIR.
- C) Con DIRNOS.

50. ¿Un fichero SYS puede tener la extensión .COM?

- A) SYSeñor.
- B) No.
- C) Sí.

51. ¿Mediante qué comando protegería el acceso involuntario a sus ficheros en un disco por parte de otro usuario?

- A) SYS.
- B) USER.
- C) TOP SECRET.

52. ¿Se puede borrar un fichero SYS con el comando ERASE?

- A) No.
- B) Sí.
- C) No, con ERASYS.

53. ¿Cuántos parámetros requiere obligatoriamente la orden RENAME?

- A) 1.
- B) 3,56.
- C) 2.

54. ¿Es obligatorio en el comando TYPE incluir la especificación PAGE o NOPAGE?

- A) Sí.
- B) No.
- C) No. Lo importante es que PAGUE.

ED: EL EDITOR DE CP/M



D es el editor de texto que acompaña a CP/M. Es muy importante no confundirlo con un procesador de texto, como el Locoscript o el Amsword, pero se comporta bastante bien, aunque en principio parece algo complicado de manejar, cuando se trata de crear, por ejemplo, ficheros que posteriormente serán compilados o líneas de órdenes para un fichero de extensión .SUB.

Para entrar en el editor de CP/M basta con asegurarse de tener el fichero ED.COM en la unidad actual y teclear a continuación: ED nombrefichero. Si éste existe, aparece en la pantalla el prompt de ED, un asterisco (*); mientras que si no lo encuentra, el sistema asume que lo que el usuario pretende es crear un nuevo fichero, haciéndolo notar con la indicación NEW FILE, previa a la impresión del asterisco.

Otra manera de activar el editor consiste en introducir directamente la orden ED, produciéndose la siguiente secuencia de acontecimientos suponiendo, por ejemplo, que el fichero ED.COM estuviera sobre el disco de la unidad A:

```
A>ED
```

```
Enter input file:
```

El programa pide el nombre del fichero que el usuario piensa editar. Si en este punto se pulsa RETURN, se regresa al sistema operativo. Si por el contrario, la respuesta es cualquier nombre válido de fichero, ED presenta la siguiente indicación:

Enter output file:

Ahora es posible optar entre varias alternativas. Si pulsamos RETURN, el fichero de salida se grabará con el mismo nombre que el de entrada; si se introduce cualquier nombre de fichero válido, el programa comprueba si el fichero de entrada existe. De ser así, el de salida adopta el nombre señalado por el usuario, o aparece el mensaje ERROR-File not found, en caso contrario.

Tanto a la pregunta sobre nombre del fichero de entrada como de salida se puede indicar que sean buscados o creados en un disco situado en una unidad diferente a la actual, anteponiendo al nombre del fichero, como siempre, su indicativo seguido de dos puntos. En todo caso, el formato general de la orden de entrada al editor ED, debe seguir la siguiente sintaxis:

ED {d:}nombrefichero-entrada {d:}{nombrefichero-salida}

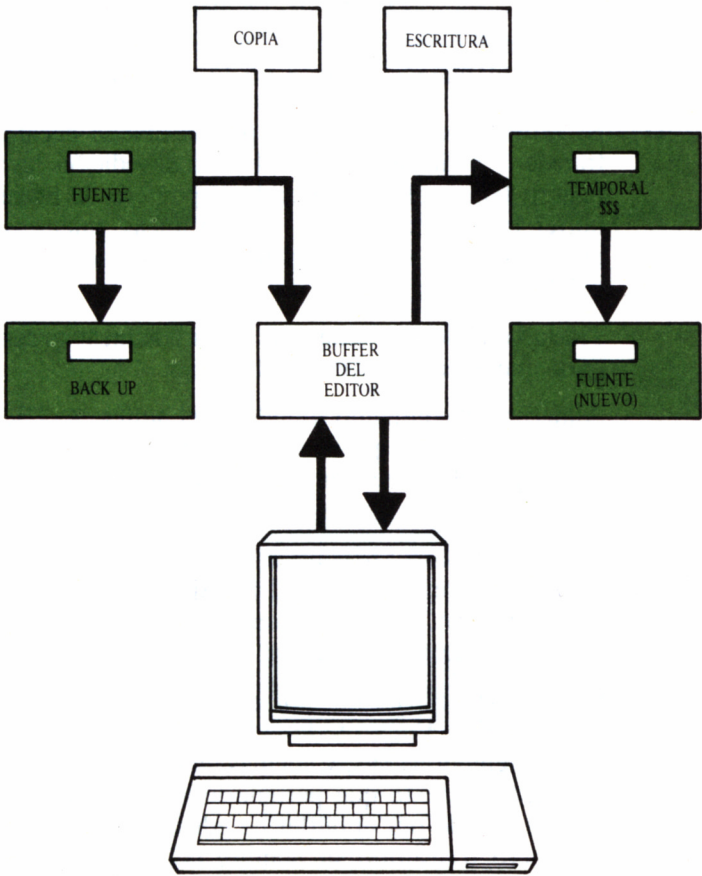
Otro detalle a tener en cuenta es que la mayoría de los editores, una vez cargado el texto en memoria, lo visualizan para su inmediata manipulación por parte del usuario. Sin embargo, cuando efectuamos la orden ED nombre fichero, aunque éste pudiera existir ya en el disco, la pantalla del monitor permanece vacía mostrando sólo el asterisco para indicar que en realidad no se ha realizado ninguna operación.

Antes de adentrarnos en el manejo de este editor y en la explicación de todos sus comandos, conviene entender un punto más: ED mantiene en todo momento un puntero señalando a una línea que en lo sucesivo denominaremos actual. Es muy importante no confundirlo con el cursor de texto, puesto que mientras éste puede estar al final del mismo, el puntero puede estar señalando todavía, por ejemplo, a la línea número uno.

Una última particularidad del editor de CP/M reside en su peculiar forma de trabajar y actualizar los ficheros con los que está trabajando. Las modificaciones que pudiéramos realizar en el fichero fuente no se efectúan directamente en éste, sino en una zona de la memoria que el programa reserva como memoria temporal. Para evitar que los caracteres allí almacenados se pierdan o puedan modificar el contenido del fichero origen, ED crea un fichero transitorio con extensión .\$\$\$, el cual solamente recibe su nombre definitivo una vez que le confirmamos al editor que efectivamente los nuevos datos son correctos.

LOS COMANDOS DE ED

A continuación están descritos todos los comandos manejados por ED agrupados según la función que realizan. En principio pueden parecer complicados y difíciles de memorizar y el lector comprobará que muchas de las potentes utilidades que se pueden encontrar en un procesador de textos comercial brillan por su ausencia en la presente lista. No obstante conviene no olvidar el objetivo de esta ayuda de CP/M: crear pequeños ficheros sin grandes aspiraciones en cuanto a su presentación.

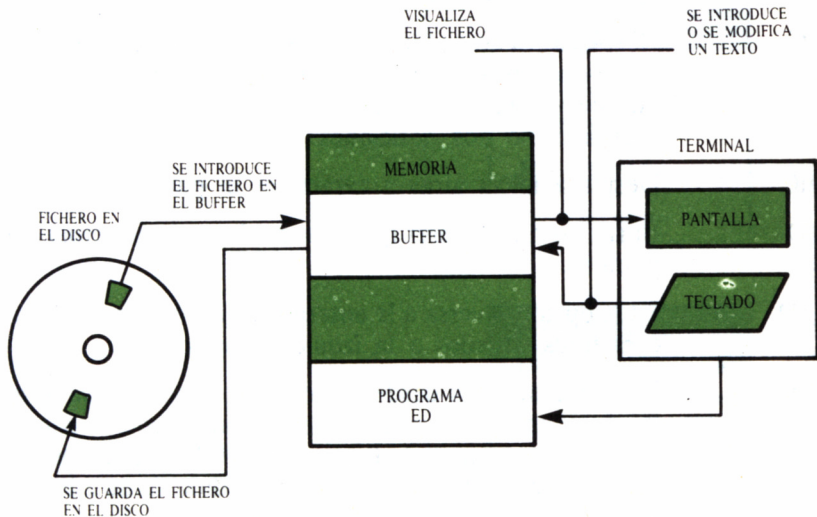


2.8.1. Esquema de trabajo con ED.

TRANSFERENCIAS DE BLOQUES

- nA** Añade n líneas (n como máximo puede valer 65535) del fichero fuente almacenado en el disco al buffer de edición. Naturalmente sólo las «añade», es decir, siguen sin visualizarse en la pantalla. Para la modificación de ficheros de grandes dimensiones los cuales podrían no caber en la zona de edición, este comando admite una posible variante:
- 0A** Carga en el buffer un número de líneas no superior a la mitad de la capacidad del mismo.
- #A** Carga en el buffer el fichero fuente completo, siempre y cuando el tamaño de éste no lo desborde.
- nW** Es la orden contraria a nA puesto que se encarga de transferir n líneas desde el buffer de edición hasta el fichero temporal, actualizando el puntero de líneas del buffer trasladándolo hacia atrás n líneas, mientras que mueve hacia adelante la misma cantidad que el puntero del fichero temporal.
- 0W** Traslada al fichero temporal todas las líneas necesarias para que medio buffer quede vacío.
- #W** Almacena el contenido completo del buffer en el fichero temporal. Nótese que manejando conjuntamente 0A y 0W es posible la edición de ficheros de grandes dimensiones.
- nX** Extrae n líneas del fichero de texto y las almacena en un fichero temporal X\$\$\$\$\$.LIB. Si no existe este fichero en el disco, ED creará uno automáticamente, para cuya eliminación se utilizará la opción 0X.
- E** Es el utilizado para dar por concluida la sesión de trabajo con ED. Vuelca todo el contenido del buffer en el fichero temporal .\$\$\$, y además también traslada a éste las líneas de texto que perteneciendo al fichero fuente no han pasado por el buffer de edición. Finalmente, da al fichero transitorio su nombre definitivo (el elegido por el usuario como nombre del fichero de salida), borra el buffer y regresa al sistema operativo.

- H Obliga a ED a realizar las mismas funciones que E, sin embargo, no se regresa al sistema operativo sino que se activa nuevamente el editor sobre la nueva versión del fichero. Esta es una opción que debe tomarse con cierta frecuencia durante largas sesiones de trabajo con ED para evitar la pérdida accidental de información.
- Q Abandona todas las modificaciones realizadas y devuelve al usuario al sistema operativo, tanto el buffer de edición como el contenido del fichero temporal se pierde, y por tanto, se conserva el fuente tal y como estaba antes de comenzar su edición. Puesto que esta opción se debe manejar con cierta precaución, al ejecutar este comando el programa presenta el siguiente mensaje:
 Q-(Y/N)?
 solicitando una última confirmación de que efectivamente es esta orden la que se pretende ejecutar (Y=Si; N=No).
- O Conduce a los mismos resultados que Q, pero en lugar de regresar al sistema operativo reactiva el editor sobre el mismo fichero, es decir, la información almacenada tanto en el buffer como en el fichero temporal se pierde y el fuente permanece inalterado.



2.8.2. Movimientos de memoria con ED.

CONTROL DEL BUFFER DE EDICION

Para obtener el máximo rendimiento de las siguientes órdenes, es preciso conocer cómo maneja ED el puntero de caracter dentro de una línea determinada. El posicionamiento del cursor no se hace visible en ningún momento, y realmente no señala a ningún caracter, sino al espacio comprendido entre éste y el siguiente.

Así por ejemplo, si en la línea: GRAN BIBLIOTECA AMS-TRAD, pidiéramos que el cursor avanzara siete caracteres, ésta seguiría presentando el mismo aspecto, aunque el cursor se encontraría entre las letras I y B de BIBLIOTECA.

Naturalmente, estas circunstancias contribuyen a dificultar el trabajo del usuario, quien no tendrá más remedio que reiterar el manejo del comando T (en seguida comentaremos su función) con el fin de totalizar los caracteres y líneas por los que se ha movido el puntero. Las órdenes que lo controlan son las siguientes:

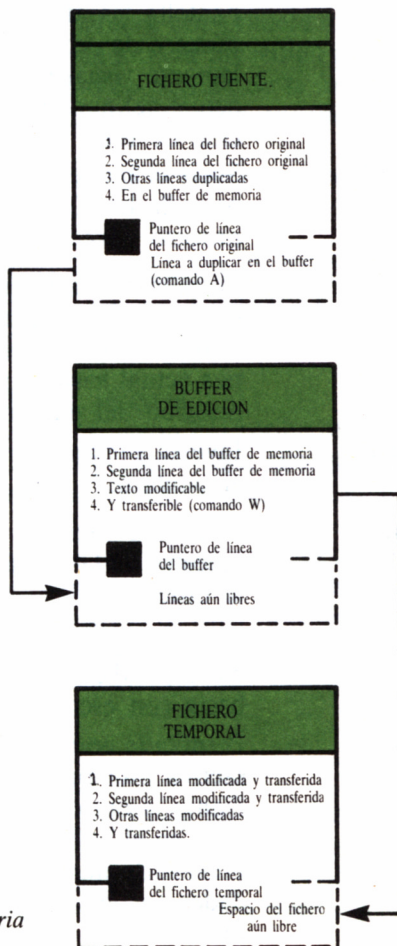
- n Avanza o retrocede, según el signo de n, el número de líneas especificadas a partir de la posición actual.
- B Sitúa el puntero en el comienzo de la primera línea almacenada en el buffer de edición.
- B Mueve el puntero al final del último caracter contenido en el buffer de edición.
- nC Desplaza el cursor (puntero) n caracteres hacia adelante sobre una línea de texto. Es importante tener en cuenta que si se llega al final de una línea el paso a la siguiente cuenta como dos posiciones.
- nC Desplaza hacia atrás n caracteres el puntero.
- nD Borra los n caracteres que siguen a la posición actual del puntero.
- nD Es la opción inversa a la anterior, es decir, elimina los n caracteres situados a la izquierda del cursor.
- n:I Inserta caracteres desde la posición actual del puntero (téngase en cuenta que no tienen nada que ver con el cursor de la pantalla). Presionando CTRL-Z finalizará la inserción; si el comando I va precedido de un número de línea n, el puntero se desplaza hasta dicha línea antes de comenzar la inserción. Naturalmente, la inserción no

tiene por qué comenzar al principio de la línea, siendo posible desplazar adelante y atrás el puntero mediante el comando C.

Esta opción es fundamental, dado que se emplea para abandonar la línea de órdenes y comenzar la escritura propiamente dicha del texto.

nK

Borra las n líneas situadas a continuación del puntero, con la salvedad de que si éste no está al principio de una línea los caracteres que le preceden no son eliminados.



2.8.3. Organización de memoria con ED.

- nK Elimina las n líneas que preceden al puntero. Como antes, si éste no se encuentra al inicio de una línea, los caracteres que le siguen no se borran.
- nL Mueve el puntero al principio de la línea actual y lo desplaza n líneas hacia adelante en el texto. Si n es cero, el puntero se sitúa al principio de la línea en curso.
- nL Tiene idéntico comportamiento a la orden anterior, pero el puntero se traslada por el texto n líneas hacia atrás.
- nP Sitúa el puntero 23 líneas hacia adelante y muestra en la pantalla las afectadas, repitiéndose esta operación un número de veces determinadas por n. La opción 0P solamente visualiza las 23 líneas siguientes a la posición actual del puntero, sin producirse alteración en la posición de éste.
- nP Sitúa el puntero 23 líneas hacia atrás y muestra en la pantalla las 23 líneas siguientes, repitiéndose el mismo proceso n veces.
- nT Permite visualizar en la pantalla n líneas desde el primer carácter que sigue al puntero. La opción 0T permite visualizar todos los caracteres desde el principio de la línea actual hasta el puntero. La orden T no modifica la posición del puntero.
- nT Inversa a la anterior, visualiza las n líneas que preceden al puntero.
- n: Sitúa el puntero al principio de la línea número n y la muestra en la pantalla siendo, por tanto, equivalente a la orden nLT.
- U Fuerza a que ED convierta las letras minúsculas en sus correspondientes mayúsculas. Para inhibir esta opción basta con escribir -U.
- V Permite la numeración de las líneas; tecleando -V se desactiva esta opción. Esta orden permite además, mediante la opción 0V, visualizar la cantidad de memoria libre en el buffer y el tamaño de éste, ambos en su expresión decimal; efectuando la resta de estas dos cantidades puede determinarse el tamaño del fichero de texto.

nZ Provoca una pausa de n/2 segundos entre la ejecución de cada comando. Esta opción permite al usuario ver todo lo que está haciendo el editor cuando se ejecutan órdenes combinadas.

BUSQUEDAS Y SUSTITUCIONES

Entre las opciones disponibles en el editor de texto de CP/M también se cuentan la de búsqueda y sustitución de una cadena determinada de caracteres. Para ello se han de seguir las siguientes normas. ni:nf:nF cadena CTRL-Z

Encuentra el lugar del texto donde aparecen por primera vez los caracteres definidos en cadena. Si el comando F va precedido por el número n, ED buscará n veces la aparición en el texto de la cadena. En estos casos el editor desplaza el puntero a la línea apropiada y visualiza su número.

Si el usuario no precisa buscar a través del fichero completo, entonces deben indicarse los números ni y nf entre los que se ha de ejecutar la orden.

nScadena buscada CTRL-Z cadena para sustituir CTRL-Z

Esta orden busca la primera cadena y una vez localizada la sustituye por la segunda. La operación puede efectuarse un número dado de veces determinado por n.

nJcadena buscada CTRL-Z cadena a insertar

Introduce una cadena al final de otra. Cuando ED recibe esta orden, buscará desde la posición actual del puntero hasta encontrar la «cadena buscada» y entonces insertará la «cadena a insertar», inmediatamente después de ella.

Por otra parte, hay que considerar que todos los comandos de ED que hemos estado viendo en este capítulo pueden ser encadenados, escribiéndolos simplemente uno a continuación de otro en la línea de órdenes (tras el asterisco).

ORDENES INMEDIATAS

Dentro del repertorio de órdenes de ED existe un grupo especial destinado a provocar inmediatamente el efecto deseado, sin necesidad de tener que pulsar RETURN. Están todas formadas por la pulsación de la tecla CONTROL (ALT) más una letra. Veamos a continuación cuáles son las admitidas por ED:

CTRL-C	Interrumpe instantáneamente la ejecución de cualquier trabajo que estuviera realizando el editor y regresa a CP/M.
CTRL-E	Provoca un salto de línea y permite continuar la introducción de órdenes en la nueva. Es útil cuando se pretende escribir una larga línea de órdenes.
CTRL-H	Borra el último carácter escrito.
CTRL-I	Desplaza el cursor a la siguiente posición de tabulación.
CTRL-J	Tiene el mismo efecto que pulsar RETURN.
CTRL-L	Inserta los códigos de salto de línea (LF) y retorno de carro (CR) en una línea de comandos. Manejando esta opción junto con los comandos F o S permite la búsqueda del final de una línea.
CTRL-M	Idéntico resultado que pulsando RETURN.
CTRL-X	Borra la línea actual.
CTRL-Z	Define el final de introducción de caracteres en una línea de comandos.

LOS MENSAJES DE ED

Manejando ED, a tenor de lo visto anteriormente, es bastante fácil cometer errores. Todos ellos siguen el mismo formato:

BREAK «indicador» AT «letra de comando»

El indicador identifica el tipo de error, mientras que la letra indica el comando que se estaba ejecutando cuando éste se produjo. Los indicadores de error tienen los siguientes significados:

- ? ED no ha podido reconocer la orden ya sea porque las letras no se corresponden con ningún comando o por estar mal especificado.
- > El buffer de texto está completo. Será necesario grabar una parte de él para continuar el trabajo.
- # No fue encontrada la cadena buscada, o ED no puede obedecer la repetición de un comando el número de veces especificado.

0 El editor no puede abrir el fichero .LIB especificado en un comando R. Este error suele ocurrir cuando colocamos un disco equivocado en la unidad con la que estamos trabajando.

No es de extrañar visto todo lo anterior que familiarizarse con los comandos de ED pueda resultar una labor si no imposible, ardua para cualquier usuario. Sin embargo, existe un factor que habla a su favor: ha permanecido virtualmente inalterado en el paso de unas versiones a otras de CP/M, y muchos programadores lo consideran, todavía, una herramienta insustituible en la confección de ficheros de texto.

2.8.4. Edición de un texto mediante ED.

```
A>ed
Enter Input file: GBA.doc
Enter Output file:

NEW FILE
: #1
1:      A principio de los años setenta, Gary Kildall, uno de los
2:      genios legendarios de Silicon Valley, comenzo a comercializar
3:      una version de lo que el mismo denomino CP/M, siglas de
4:      Control Program for Microprocessors o programa de control para
5:      microprocesadores.
6:
: #e

A>
```

```
A>ed
Enter Input file: GBA.doc
Enter Output file: INTRO.doc

: #0a
1: #5t
2:      A principio de los años setenta, Gary Kildall, uno de los
3:      genios legendarios de Silicon Valley, comenzo a comercializar
4:      una version de lo que el mismo denomino CP/M, siglas de
5:      Control Program for Microprocessors o programa de control para
6:      microprocesadores.
7:      El exito alcanzado por aquella version, le
8:      llevo a fundar la DIGITAL RESAERCH, hoy en dia una de las mas
9:      importantes firmas a nivel mundial de software de base y
10:
: #5:
11: #18c
12: #1
13: microprocesadores. El exito alcanzado por aquella version, le
14: llevo a fundar la DIGITAL RESAERCH, hoy en dia una de las mas
15: importantes firmas a nivel mundial de software de base y
16:
```

```
8: #6:
0: #29c
0: #2d
0: #1
0: llevo a fundar la DIGITAL RESEA
0: #6:
0: #1t
0: llevo a fundar la DIGITAL RESEARCH, hoy en día una de las mas
0: #2
8:
8: #1
8: importantes.
```

```
9:
9: #1:
1: #sDIGITAL RESEARCH`ZDigital Research`Z
0: #1t
hoy en día una de las mas
0: #0l
0: #1t
0: llevo a fundar la Digital Research, hoy en día una de las mas
0: #1:10t
1: A principio de los años setenta, Gary Kildall, uno de los
2: genios legendarios de Silicon Valley, comenzo a comercializar
3: una version de lo que el mismo denomino CP/M, siglas de
4: Control Program for Microprocessors o programa de control para
5: microprocesadores. El exito alcanzado por aquella version, le
6: llevo a fundar la Digital Research, hoy en día una de las mas
7: importantes firmas a nivel mundial de software de base y
8: importantes.
9:
10:
```

```
1: #8:k
8: #1
8: aplicaciones.
8: #1:-v10t0v
1: A principio de los años setenta, Gary Kildall, uno de los
genios legendarios de Silicon Valley, comenzo a comercializar
una version de lo que el mismo denomino CP/M, siglas de
Control Program for Microprocessors o programa de control para
microprocesadores. El exito alcanzado por aquella version, le
llevo a fundar la Digital Research, hoy en día una de las mas
importantes firmas a nivel mundial de software de base y
aplicaciones.
```

```
39545/40027
#e
```

```
A>
```

EJERCICIOS

55. ¿Se puede crear un texto con ED como si fuera un procesador de textos?

- A) Sí, aunque no es recomendable dada su complejidad de manejo.
- B) No, porque sólo sirve para la edición de órdenes.
- C) Sí, pero para eso están las bases de datos.

56. ¿Es preciso indicar el nombre del fichero a editar en ED?

- A) Sí, siempre es conveniente saber con quién se anda.
- B) Sí, es indispensable.
- C) No, si vamos a crear uno nuevo.

57. ¿Qué nombre toma el fichero de salida de ED?

- A) El que indique el usuario.
- B) Siempre el mismo que el de entrada.
- C) Según le viene.

58. ¿Los cambios grabados con ED en un fichero se van almacenando en el disco según los realizamos?

- A) Sí.
- B) No.
- C) No, porque ED no sirve para nada.

59. Una vez modificado en la pantalla el fichero fuente ¿puede éste recuperarse?

- A) Sí, pero con amplios conocimientos de código máquina.
- B) Sí, simplemente abandonando la edición con Q.
- C) Santa Rita, Rita, Rita... lo que se escribe no se quita.

60. Una vez editado un fichero ¿se visualiza inmediatamente su contenido en la pantalla?

- A) Sí, nada más finalizar la carga.
- B) No, sólo si introducimos algunas órdenes a tal fin.
- C) Sí, y emite destellos parpadeantes de los que parpadean.

61. Para escribir texto en una línea, ¿qué operación es necesario efectuar?

- A) Ninguna; se escribe directamente.
- B) ¿Es que esto puede escribir?
- C) Introducir la orden I.

62. ¿Cómo se retorna desde la introducción de texto al modo de órdenes?

- A) Pulsando RETURN.
- B) ¡A lo bruto!: con CONTROL-C.
- C) Finalizando la línea con CONTROL-Z.

63. Una vez comunicado a ED cuál es el texto a editar ¿cuál de las siguientes secuencias de órdenes sería la correcta para visualizar las primeras 50 líneas?

- A) Abra-Cadabra: visualiza 50 líneas.
- B) 0A50T.
- C) 0A50C.

64. Una vez actualizado un texto ¿cómo se graba la nueva versión en el fichero destino?

- A) Mediante la orden E.
- B) Mediante la orden Q.
- C) Mediante la orden SAVE «texto».

65. ¿Qué es el fichero temporal de ED?

- A) Es lo mismo que el buffer de edición.
- B) El utilizado por el editor una vez terminada la edición para crear el fichero definitivo.
- C) Si su extensión es A.M está en la memoria; si es P.M no.

66. Manejando ED ¿es lo mismo puntero que cursor de texto?

- A) No.
- B) Sí, siempre.
- C) Depende de lo que estemos señalando.

67. ¿Qué efecto tiene para ED la orden 0V?
- A) Provoca un error puesto que no existe tal orden.
 - B) Carga el buffer de edición con la mitad de las líneas del fichero de texto.
 - C) Muestra la cantidad de memoria libre en el buffer y el tamaño de éste.

DIRECTORIOS PARA TODOS

LOS GUSTOS

Las opciones que pueden seguir al comando DIR constituyen una ampliación de sus posibilidades, ya comentadas en el capítulo dedicado a las órdenes residentes. Cada una de las opciones muestra un directorio de los ficheros almacenados en el disco con gran variedad de formatos.

DIR puede buscar los ficheros en una o en todas las unidades, para cualquier zona de usuario que se le solicite. Las opciones, como norma sintáctica habitual de CP/M, deben encerrarse entre corchetes. No obstante, sólo es preciso señalar una o dos de sus iniciales, no siendo necesario escribir el último corchete que engloba las opciones.

Pueden emplearse los símbolos comodín para sustituir cualquier grupo de caracteres dentro del nombre de un fichero y tareas como mostrar un catálogo completo del disco con sus ficheros ordenados alfabéticamente o los que contienen una fecha determinada, son sólo algunas de sus posibilidades. A continuación se detallan cada una de las opciones que admite DIR como comando transitorio.

La sintaxis correcta para una línea de órdenes del comando DIR, debe seguir alguna de las dos formas siguientes según que el directo-

rio se refiera a todo el disco o sólo a algunos de los ficheros que contenga, respectivamente:

DIR {d:} [opciones]

DIR {nombrefich} {nombrefich} ... [opciones]

siendo d la unidad de disco especificada, y nombrefich el nombre del fichero más su extensión.

TODAS LAS OPCIONES DE DIR

ATT muestra todos los atributos de fichero.

DATE visualiza la hora y fecha de creación, actualización o último acceso de los ficheros seleccionados, siempre y cuando esta opción haya sido activada previamente, como veremos en el capítulo dedicado al comando SET.

DIR muestra el directorio de todos los ficheros con el atributo del sistema DIR.

DRIVE=ALL visualiza todos los ficheros almacenados en todas las unidades.

DRIVE=(A,B,C,...,P) muestra el directorio de los discos ubicados en las unidades señaladas.

DRIVE=d muestra los ficheros en el disco almacenado en la unidad identificada por d.

EXCLUDE visualiza una lista de todos los ficheros cuyos nombres no coincidan con los señalados en el comando.

FF envía hacia la impresora un salto de página (*Form Feed*) antes de emitir el directorio por ésta, si es que fue activada mediante CTRL-P (ALT-P, en los PCW).

FULL permite visualizar el nombre, tamaño, número de registros de 128 bytes ocupados, y los atributos del sistema de todos los ficheros. Si el directorio tiene alguna etiqueta, son mostrados el modo de protección y las fechas registradas.

LENGTH=n visualiza n líneas de salida antes de volver a insertar la cabecera, debiendo estar n comprendido entre 5 y 65536.

MESSAGE muestra las unidades de disco y los números de usuario en los que está efectuando el directorio.

NOSORT visualiza el directorio de los ficheros sin clasificar, y en el orden en que los va encontrando en el disco.

RO muestra un directorio con sólo los ficheros que tengan el atributo del sistema de sólo lectura (RO).

RW visualiza sólo los ficheros con el atributo de lectura-escritura (RW).

SIZE muestra una lista del nombre y tamaño en kilobytes, de los ficheros almacenados en el disco.

SYS visualiza una lista de los ficheros con el atributo del sistema system (SYS).

USER=ALL muestra los ficheros almacenados en todas las zonas de usuario utilizadas en el disco situado en la unidad por defecto.

USER=n lista los ficheros alojados en el área de usuario número n.

USER=(0,1,...,15) muestra los ficheros localizados en las zonas de usuario marcadas entre paréntesis.

Pueden señalarse varias opciones conjuntamente, dentro de la línea de órdenes del comando DIR. En estos casos, basta con encerrarlas entre corchetes, como siempre, pero separadas una de otra, por una coma (,).

2.9.1. Muestra de directorios.

```
A>dir
A: CARTAS GRP : FRASES EST : PLANTILL EST : LEA ME
SYSTEM FILE(S) EXIST
A>dirsys
A: MAIL232 COM : J11SLOCO EMS : MATRIZ EST : SCRIPT JOY
NON-SYSTEM FILE(S) EXIST
A>m:
M>dir a:[dir]

Scanning Directory...
Sorting Directory...
Directory For Drive A: User 0

  Name      Bytes  Recs  Attributes      Name      Bytes  Recs  Attributes
-----
CARTAS GRP   0k     0 Dir RW   FRASES EST   1k     8 Dir RW
LEA ME      2k     9 Dir RW   PLANTILL EST 1k     4 Dir RW

Total Bytes =      4k Total Records =    21 Files Found =    4
Total 1k Blocks =    4 Used/Max Dir Entries For Drive A:  33/ 64

M>dir a:[sys]

Scanning Directory...
Sorting Directory...
Directory For Drive A: User 0

  Name      Bytes  Recs  Attributes      Name      Bytes  Recs  Attributes
-----
J11SLOCO EMS 43k   344 Sys RW   MAIL232 COM   4k    32 Sys RW
MATRIZ EST   7k    56 Sys RW   SCRIPT JOY   31k   248 Sys RW

Total Bytes =    85k Total Records =   680 Files Found =    4
Total 1k Blocks =    85 Used/Max Dir Entries For Drive A:  33/ 64
```

```

A>dir
A: J12SCPM3 EMS : BASIC COM : DIR COM : ED COM : ERASE COM
A: KEYS WP : LANGUAGE COM : PALETTE COM : PAPER COM : PIP COM
A: PROFILE ENG : RENAME COM : SET COM : SET24X80 COM : SETDEF COM
A: SETKEYS COM : SETLST COM : SETSIO COM : SHOW COM : SUBMIT COM
A: TYPE COM : RPED BAS : RPED SUB : DISCKIT COM
A>dir m:
M: DIR COM : ED COM : ERASE COM : PIP COM : RENAME COM
M: SET COM : SHOW COM : TYPE COM : DISCKIT COM : DATE COM
M: INITDIR COM

```

```

A>dir [drive=all]
Scanning Directory...
Sorting Directory...

```

Directory For Drive A: User 0

Name	Bytes	Recs	Attributes	Name	Bytes	Recs	Attributes
BASIC COM	28k	224	Dir RW	DIR COM	15k	114	Dir RW
DISCKIT COM	7k	56	Dir RW	ED COM	10k	73	Dir RW
ERASE COM	4k	29	Dir RW	J12SCPM3 EMS	40k	320	Dir RW
KEYS WP	1k	7	Dir RW	LANGUAGE COM	1k	8	Dir RW
PALETTE COM	1k	8	Dir RW	PAPER COM	2k	16	Dir RW
PIP COM	9k	68	Dir RW	PROFILE ENG	1k	2	Dir RW
RENAME COM	3k	23	Dir RW	RPED BAS	7k	56	Dir RW
RPED SUB	1k	1	Dir RW	SET COM	11k	81	Dir RW
SET24X80 COM	1k	8	Dir RW	SETDEF COM	4k	32	Dir RW
SETKEYS COM	2k	16	Dir RW	SETLST COM	2k	16	Dir RW
SETSIO COM	2k	16	Dir RW	SHOW COM	9k	66	Dir RW
SUBMIT COM	6k	42	Dir RW	TYPE COM	3k	24	Dir RW

```

Total Bytes = 170k Total Records = 1306 Files Found = 24
Total 1k Blocks = 170 Used/Max Dir Entries For Drive A: 27/ 64

```

```

Scanning Directory...
Sorting Directory...

```

Directory For Drive M: User 0

Name	Bytes	Recs	Attributes	Name	Bytes	Recs	Attributes
DATE COM	3k	23	Dir RD	DIR COM	15k	114	Dir RD
DISCKIT COM	7k	56	Dir RD	ED COM	10k	73	Dir RW
ERASE COM	4k	29	Dir RW	INITDIR COM	32k	250	Dir RW
PIP COM	9k	68	Dir RW	RENAME COM	3k	23	Dir RW
SET COM	11k	81	Dir RW	SHOW COM	9k	66	Dir RW

Press RETURN to Continue

```

TYPE COM 3k 24 Dir RW
Total Bytes = 106k Total Records = 807 Files Found = 11
Total 1k Blocks = 106 Used/Max Dir Entries For Drive M: 12/ 64

```

```

A>dir *.com(exclude)

```

```

Scanning Directory...
Sorting Directory...

```

Directory For Drive A: User 0

Name	Bytes	Recs	Attributes	Name	Bytes	Recs	Attributes
J12SCPM3 EMS	40k	320	Dir RW	KEYS WP	1k	7	Dir RW
PROFILE ENG	1k	2	Dir RW	RPED BAS	7k	56	Dir RW
RPED SUB	1k	1	Dir RW				

```

Total Bytes = 50k Total Records = 386 Files Found = 5
Total 1k Blocks = 50 Used/Max Dir Entries For Drive A: 27/ 64

```

A>dir [full]

Scanning Directory...

Sorting Directory...

Directory For Drive A: User 0

Name	Bytes	Recs	Attributes	Name	Bytes	Recs	Attributes
AMSDOS	COM	1k	8 Dir RW	BANKMAN	BAS	1k	7 Dir RW
BANKMAN	BIN	2k	12 Dir RW	C10CPM3	EMS	25k	200 Dir RW
DATE	COM	3k	23 Dir RW	DEVICE	COM	8k	58 Dir RW
DIR	COM	15k	114 Dir RW	DISCKIT3	COM	6k	48 Dir RW
ED	COM	10k	73 Dir RW	ERASE	COM	4k	29 Dir RW
GET	COM	7k	51 Dir RW	KEYS	CCP	1k	3 Dir RW
KEYS	WP	1k	3 Dir RW	LANGUAGE	COM	1k	8 Dir RW
PALETTE	COM	1k	8 Dir RW	PIP	COM	9k	68 Dir RW
PROFILE	ENG	1k	1 Dir RW	PUT	COM	7k	55 Dir RW
RENAME	COM	3k	23 Dir RW	SET	COM	11k	81 Dir RW
SET24X80	COM	1k	8 Dir RW	SETDEF	COM	4k	32 Dir RW
SETKEYS	COM	2k	16 Dir RW	SETLST	COM	2k	16 Dir RW
SETSIO	COM	2k	16 Dir RW	SHOW	COM	9k	66 Dir RW
SUBMIT	COM	6k	42 Dir RW	TYPE	COM	3k	24 Dir RW

Total Bytes = 146k Total Records = 1093 Files Found = 28
Total 1k Blocks = 146 Used/Max Dir Entries For Drive A: 29/ 64

A>dir [nosort]

Scanning Directory...

Directory For Drive A: User 0

Name	Bytes	Recs	Attributes	Name	Bytes	Recs	Attributes
C10CPM3	EMS	25k	200 Dir RW	BANKMAN	BAS	1k	7 Dir RW
PROFILE	ENG	1k	1 Dir RW	SUBMIT	COM	6k	42 Dir RW
SETKEYS	COM	2k	16 Dir RW	KEYS	CCP	1k	3 Dir RW
LANGUAGE	COM	1k	8 Dir RW	SET24X80	COM	1k	8 Dir RW
PALETTE	COM	1k	8 Dir RW	SETSIO	COM	2k	16 Dir RW
SETLST	COM	2k	16 Dir RW	DISCKIT3	COM	6k	48 Dir RW
DATE	COM	3k	23 Dir RW	DEVICE	COM	8k	58 Dir RW
DIR	COM	15k	114 Dir RW	ED	COM	10k	73 Dir RW
ERASE	COM	4k	29 Dir RW	GET	COM	7k	51 Dir RW
PIP	COM	9k	68 Dir RW	PUT	COM	7k	55 Dir RW
RENAME	COM	3k	23 Dir RW	SHOW	COM	9k	66 Dir RW
TYPE	COM	3k	24 Dir RW	SET	COM	11k	81 Dir RW
SETDEF	COM	4k	32 Dir RW	AMSDOS	COM	1k	8 Dir RW
BANKMAN	BIN	2k	12 Dir RW	KEYS	WP	1k	3 Dir RW

Total Bytes = 146k Total Records = 1093 Files Found = 28
Total 1k Blocks = 146 Used/Max Dir Entries For Drive A: 29/ 64

A>dir m:[ro]

Scanning Directory...

Sorting Directory...

Directory For Drive M: User 0

Name	Bytes	Recs	Attributes	Name	Bytes	Recs	Attributes
DATE	COM	3k	23 Dir RO	DIR	COM	15k	114 Dir RO
DISCKIT	COM	7k	56 Dir RO				

Total Bytes = 25k Total Records = 193 Files Found = 3
Total 1k Blocks = 25 Used/Max Dir Entries For Drive M: 12/ 64

A>dir (length=10)

Scanning Directory...

Sorting Directory...

Directory For Drive A: User 0

Name	Bytes	Recs	Attributes	Name	Bytes	Recs	Attributes
AMSDOS	COM	1k	8 Dir RW	BANKMAN	BAS	1k	7 Dir RW
BANKMAN	BIN	2k	12 Dir RW	C10CPM3	EMS	25k	200 Dir RW
DATE	COM	3k	23 Dir RW	DEVICE	COM	8k	58 Dir RW
DIR	COM	15k	114 Dir RW	DISCKIT3	COM	6k	48 Dir RW
ED	COM	10k	73 Dir RW	ERASE	COM	4k	29 Dir RW

Press RETURN to Continue

Directory For Drive A: User 0

Name	Bytes	Recs	Attributes	Name	Bytes	Recs	Attributes
GET	COM	7k	51 Dir RW	KEYS	CCP	1k	3 Dir RW
KEYS	WP	1k	3 Dir RW	LANGUAGE	COM	1k	8 Dir RW
PALETTE	COM	1k	8 Dir RW	PIP	COM	9k	68 Dir RW
PROFILE	ENG	1k	1 Dir RW	PUT	COM	7k	55 Dir RW
RENAME	COM	3k	23 Dir RW	SET	COM	11k	81 Dir RW

Press RETURN to Continue

Directory For Drive A: User 0

Name	Bytes	Recs	Attributes	Name	Bytes	Recs	Attributes
SET24X80	COM	1k	8 Dir RW	SETDEF	COM	4k	32 Dir RW
SETKEYS	COM	2k	16 Dir RW	SETLST	COM	2k	16 Dir RW
SETSIO	COM	2k	16 Dir RW	SHOW	COM	9k	66 Dir RW
SUBMIT	COM	6k	42 Dir RW	TYPE	COM	3k	24 Dir RW

Press RETURN to Continue

Total Bytes = 146k Total Records = 1093 Files Found = 28
Total 1k Blocks = 146 Used/Max Dir Entries For Drive A: 29/ 64

M>dir a:{user=(0,7)}

Scanning Directory...

Sorting Directory...

Directory For Drive A: User 0

Name	Bytes	Recs	Attributes	Name	Bytes	Recs	Attributes
ED	COM	10k	73 Dir RW	ENTRADA	BAK	1k	1 Dir RW
ENTRADA	TXI	1k	1 Dir RW	GBA	BAK	1k	3 Dir RW
GBA	DOC	1k	3 Dir RW	INTRO	\$\$\$	0k	0 Dir RW

Total Bytes = 14k Total Records = 81 Files Found = 6
Total 1k Blocks = 14 Used/Max Dir Entries For Drive A: 24/ 64

Directory For Drive A: User 7

Name	Bytes	Recs	Attributes	Name	Bytes	Recs	Attributes
DATE	COM	3k	23 Dir RO	DIR	COM	15k	114 Dir RO
ERASE	COM	4k	29 Dir RW	RENAME	COM	3k	23 Dir RW
TYPE	COM	3k	24 Dir RW				

Total Bytes = 28k Total Records = 213 Files Found = 5
Total 1k Blocks = 28 Used/Max Dir Entries For Drive A: 24/ 64

```

A>dir (size)

Scanning Directory...

Sorting Directory...

Directory For Drive A: User 0

A: AMSDOS COM 1k : BANKMAN BAS 1k : BANKMAN BIN 2k
A: CIOCPM3 EMS 25k : DATE COM 3k : DEVICE COM 8k
A: DIR COM 15k : DISCKIT3 COM 6k : ED COM 10k
A: ERASE COM 4k : GET COM 7k : KEYS CCP 1k
A: KEYS WP 1k : LANGUAGE COM 1k : PALETTE COM 1k
A: PIP COM 9k : PROFILE ENG 1k : PUT COM 7k
A: RENAME COM 3k : SET COM 11k : SET24X80 COM 1k
A: SETDEF COM 4k : SETKEYS COM 2k : SETLST COM 2k
A: SETSIO COM 2k : SHOW COM 9k : SUBMIT COM 6k
A: TYPE COM 3k

Total Bytes = 146k Total Records = 1093 Files Found = 28
Total 1k Blocks = 146 Used/Max Dir Entries For Drive A: 29/ 64

```

```

M>dir a:(user=all)

Scanning Directory...

Sorting Directory...

Directory For Drive A: User 0

-----
Name      Bytes  Recs  Attributes      Name      Bytes  Recs  Attributes
-----
ED        COM    10k   73 Dir RW      ENTRADA  BAK     1k    1 Dir RW
ENTRADA  TXT     1k    1 Dir RW      GBA      BAK     1k    3 Dir RW
GBA      DOC     1k    3 Dir RW      INTRO   $$$     0k    0 Dir RW
-----

Total Bytes = 14k Total Records = 81 Files Found = 6
Total 1k Blocks = 14 Used/Max Dir Entries For Drive A: 24/ 64

```

```

Directory For Drive A: User 1

-----
Name      Bytes  Recs  Attributes      Name      Bytes  Recs  Attributes
-----
DATE      COM     3k    23 Dir RO      DIR       COM    15k   114 Dir RO
DISCKIT  COM     7k    56 Dir RO      ED        COM    10k   73 Dir RW
ERASE     COM     4k    29 Dir RW      INITDIR   COM    32k   250 Dir RW
PIP       COM     9k    68 Dir RW      RENAME    COM     3k    23 Dir RW
SET       COM    11k   81 Dir RW      SHOW      COM     9k    66 Dir RW
TYPE      COM     3k    24 Dir RW
-----

Total Bytes = 106k Total Records = 807 Files Found = 11
Total 1k Blocks = 106 Used/Max Dir Entries For Drive A: 24/ 64

```

```

Directory For Drive A: User 3

-----
Name      Bytes  Recs  Attributes      Name      Bytes  Recs  Attributes
-----
Press RETURN to Continue
RPED      BAS     7k    56 Dir RW
-----

Total Bytes = 7k Total Records = 56 Files Found = 1
Total 1k Blocks = 7 Used/Max Dir Entries For Drive A: 24/ 64

```

```

Directory For Drive A: User 7

-----
Name      Bytes  Recs  Attributes      Name      Bytes  Recs  Attributes
-----
DATE      COM     3k    23 Dir RO      DIR       COM    15k   114 Dir RO
ERASE     COM     4k    29 Dir RW      RENAME    COM     3k    23 Dir RW
TYPE      COM     3k    24 Dir RW
-----

Total Bytes = 28k Total Records = 213 Files Found = 5
Total 1k Blocks = 28 Used/Max Dir Entries For Drive A: 24/ 64

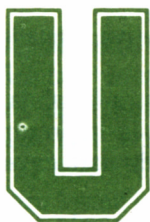
```

EJERCICIOS

68. ¿Cómo se marcan las opciones para el comando DIR?
- A) Entre paréntesis.
 - B) Entre corchetes (el de cerrado no es indispensable).
 - C) Con un hierro al rojo.
69. ¿Es posible utilizar símbolos comodín en el comando DIR?
- A) No.
 - B) Sí.
 - C) Se niegan a ser utilizados.
70. Cuando se extrae un directorio ¿siempre corresponde a la unidad de disco por defecto?
- A) Sí.
 - B) Si la unidad de disco está defectuosa no se puede extraer el directorio.
 - C) No; es posible indicar la unidad de disco deseada.
71. ¿Cómo se puede conseguir que DIR muestre los ficheros presentes en todas las unidades?
- A) Con la opción DRIVE=ALL.
 - B) Con la opción DIR.
 - C) Coaccionándole convenientemente.
72. ¿Mediante qué opción se muestran los ficheros con atributo sistema?
- A) SYS.
 - B) ATT.
 - C) OPSYS.
73. ¿Cómo saber mediante DIR el tamaño de los ficheros del disco?
- A) MESSAGE.
 - B) SIZE.
 - C) Preguntádoselo con educación.

74. ¿Es posible indicar varias opciones en el comando DIR?
- A) Sí, separándolas con puntos y comas.
 - B) De una en una, por favor.
 - C) Sí, separándolas con comas.

PIP: GENIO DE LAS TRANSFERENCIAS



Uno de los más potentes y flexibles programas de CP/M 3.0 es PIP, abreviatura de *Peripheral Interchange Program* o programa de intercambio con los periféricos, en su acepción española. Como su nombre sugiere, las funciones principales de este programa pasan por copiar ficheros, combinarlos, y transferirlos entre las unidades de disco, impresoras, consolas, o cualquier otro dispositivo conectado al ordenador. Se puede utilizar PIP, por ejemplo, para enviar un fichero hacia una impresora o hacia un interface serie conectado quizás a un modem.

Además, permite combinar las entradas de datos procedentes de varias fuentes para producir una única salida. También transfiere los atributos del sistema que pudiera tener el fichero fuente al fichero destino, incluyendo la palabra clave (*passwords*) si la hubiere.

Existen dos maneras de manejarlo, una consistente en teclear un comando que lo cargará en memoria, transferirá un fichero de un lugar a otro y seguidamente regresará a CP/M. Alternativamente, es posible cargar PIP directamente en la memoria del ordenador y trabajar sucesivamente con él. En este caso, tras la aparición del mensaje CP/M 3 PIP VERSION 3.0 aparece en la pantalla un asterisco indicando que el programa está listo para recibir órdenes.

La sintaxis completa con todas las posibles opciones que debe seguir el comando PIP es la siguiente:

```
PIP d1:{Gn}|fichdestino{{[Gn]}=d2:fichfuente{{[o]},...
```

De primeras parece ser algo terrible componer una orden completa para su ejecución con PIP, pero tengamos en cuenta que la mayoría de los parámetros señalados son opcionales. En principio, no es preciso escribir PIP si es que previamente se ha cargado este programa en memoria.

La unidad de disco o dispositivo identificado por D1 es el de destino de la información, mientras que D2 es el de origen y como enseña comprobaremos también puede suprimirse en algunas órdenes. «Fichdestino» es el nombre asignado al fichero dispositivo donde acabaran los datos a trasladar, mientras que «fichfuente» es el nombre del original donde se recogen éstos.

Los identificadores de unidad destino pueden ser AUX:, CON:, PRN o LST:, mientras que las unidades fuente pueden representarse mediante AUX:, COM:, NUL:, o EOF:. Veamos a continuación su significado.

- CON (CONsole o consola) representa el dispositivo formado por teclado más pantalla.
- AUX es cualquier dispositivo auxiliar de entrada-salida.
- LST (LiST) o PRN (PRiNter) identifican a la impresora.
- NUL representa un dispositivo fuente empleado para enviar 40 caracteres ASCII NUL.
- EOF (End Of File, fin de fichero) significa un solo carácter «control Z».

Naturalmente, a los nombres anteriores se les pueden sumar los identificativos de cualquier unidad de disco conectada al sistema, tanto como origen como destino de la información, aunque LST, por ejemplo, no puede indicarse como unidad de entrada.

Todavía hay algunas cosas más que opcionalmente el usuario puede indicar a PIP dentro de una orden. Si observamos la línea de sintaxis anterior tras «fichdestino» y entre corchetes está GN, donde n significa cualquier zona de usuario. Esta opción, de utilizarla, es la única que puede figurar tras el nombre del fichero destino de la información.

Sin embargo, tras el nombre del fichero origen pueden señalarse una serie de opciones (indicadas en la línea de sintaxis mediante una «o») que mejoren la operatividad del comando. Y no es esto todo, hemos finalizado la línea con una coma seguida de puntos suspensivos, lo que indica que pueden señalarse dentro de la misma orden una serie de ficheros fuente cada uno con sus respectivas opciones.

EJEMPLOS

Aclaremos todo lo anterior con algunos ejemplos. Supongamos que disponemos de dos unidades de disco A y B, y queremos copiar el fichero DIR.COM desde la A a la B. Para ello, una vez cargado en la memoria PIP y visualizado su prompt escribiremos:

```
*B:=A:DIR.COM
```

Nótese que no se ha señalado ningún nombre como fichero destino, simplemente hemos indicado la unidad a la cual debía ser transferido. En esta situación, PIP da automáticamente al fichero destino el mismo nombre que tuviera el fichero fuente.

Consideremos algo más complicado, suponiendo que lo que se pretende efectuar es copiar el fichero DIR.COM situado en la zona de usuario 3 del disco ubicado en la unidad A hasta el área de usuario 9 en la B, pero con la salvedad de que el nombre final con que pretendemos quede sea CAT.COM. La orden adecuada para realizar esta operación sería:

```
*B:CAT.COM[G9]=A:DIR.COM[G3]
```

Supongamos ahora que lo que se pretende es a partir de una serie de ficheros fuente FUENT1.TXT, FUENT2.TXT, FUENT3.TXT por ejemplo, obtener un único fichero destino SUMA.TXT. Para ello bastaría con indicarle a PIP la orden:

```
*SUMA.TXT=FUENT1.TXT,FUENT2.TXT,FUENT3.TXT
```

Otra forma útil de emplear PIP es para trasladar ficheros desde un dispositivo físico o lógico, en situación de generar datos, hacia otro en condiciones de aceptarlos. Por ejemplo, para imprimir un fichero de texto denominado DOCUM.TXT, bastaría con la orden:

```
*LST:=DOCUM.TXT
```

Naturalmente, el contenido del fichero enviado a la impresora, debe ser ASCII; de lo contrario, en forma similar a como ocurría con la orden TYP, los resultados pueden ser imprevisibles.

Si en cambio la orden ejecutada es:

```
*PANT.BAS=CON:
```

el resultado se traduce en que todo lo que el usuario escriba en la consola se transfiera hacia el fichero que en el ejemplo hemos denominado PANT.BAS.

TODAS LAS OPCIONES DE PIP

Como ya se ha hecho mención, otra de las particularidades de

PIP, son las diferentes opciones que pueden señalarse acompañando al nombre de un fichero, para aumentar la potencia de la orden o limitarla a un determinado tipo de acción. Las opciones han de señalarse entre corchetes. Veamos cuáles están disponibles en las órdenes enviadas a PIP y cuál es su significado:

- A Transfiere solamente los ficheros que hayan sido cambiados desde la última copia. Para ello es imprescindible que esté activada la función del sello de fecha de hora.
- B Función de lectura de bloques: carga cada vez un bloque del fichero fuente, antes de grabarlo en el fichero destino. Puede ser útil para manejar dispositivos sobre los que el ordenador no tiene control de arranque y parada. Normalmente no se utilizan en los ordenadores AMSTRAD.
- C Confirmación: PIP pregunta antes de copiar un fichero si está autorizado a transferirlo.
- Dn Envía solamente el número de caracteres determinado por n a cada línea del fichero destino, eliminando el resto. Se utiliza habitualmente para definir la longitud de línea en un dispositivo que la tenga limitada, por ejemplo, en impresora de 80 ó 132 columnas.
- E Visualiza el contenido del fichero recién copiado en la pantalla. Debe emplearse solamente con ficheros de texto.
- F Elimina el carácter de salto de página (*Form Feed*), al objeto de adecuar la salida impresa a las necesidades de una impresora concreta.
- Gn Transfiere un fichero de, o hacia el área de usuario n. En caso de señalarse esta opción para el fichero destino debe ir situada inmediatamente a continuación del nombre de éste, y es la única que puede situarse allí.
- H Transfiere ficheros de datos en formato hexadecimal. Es aconsejable utilizarlo cuando se copian ficheros .HEX, pues PIP se encarga de comprobar si éstos están escritos en el formato de ficheros hexadecimal según la norma Inter.
- I Ignora la marca de final del fichero (00H) en la transferencia de ficheros hexadecimales. La opción I activa automáticamente la opción H.

- K** Inhibe en la pantalla la indicación de que un fichero se está copiando.
- L** Convierte todos los caracteres del fichero puente en mayúsculas de sus correspondientes minúsculas del fichero destino.
- N** Añade un número de línea a cada una del fichero destino antes de grabarlo. La numeración comienza a partir de uno, incrementándose en una unidad por cada línea, los números consisten en seis cifras como máximo y detrás de ellos se sitúan en dos puntos (:). También se puede utilizar la opción N2; en este caso PIP añade un cero a los números de línea y un carácter TAB, a continuación.
- O** Esta opción asume que son «objeto» los ficheros al ser transferidos, ignorándose los caracteres CTRL-Z que actúan como marcadores de final de fichero. Normalmente, se utiliza para combinar una serie de ficheros objetos que den como resultado un único final.
- Pn** Obliga a PIP a insertar un carácter de salto de hoja (FF) después del final de cada página del fichero destino. La longitud de página por defecto es de 60 líneas, pudiendo ser definida mediante n al número que se crea conveniente.
- Qs†Z** Transfiere los datos del fichero fuente al destino hasta que la cadena definida por s es encontrada. La propia cadena también es transferida. Dentro de la orden la cadena debe terminar con un CTRL-Z(†Z).
- R** Lee ficheros con el atributo del sistema SYS que normalmente ignora PIP.
- Ss†Z** Transfiere el fichero fuente desde el punto en el cual es encontrada la cadena s hasta el fichero destino. La propia cadena también es copiada. Manejando los comandos Q y S simultáneamente es posible transferir un bloque determinado de un fichero de texto.
- Tn** Expande los caracteres TAB (CTRL-I) a n espacios. Normalmente CP/M trabaja con una amplitud de 8 caracteres. Cuando un carácter TAB se lee en el fichero fuente es sustituido por el número de espacios apropiados, de forma que el siguiente carácter del fichero se sitúa en un lugar de la línea actual divisible por n (amplitud del tabulador).

- U Transforma todos los caracteres en minúsculas del fichero fuente en sus correspondientes mayúsculas en el fichero destino.
- V Verifica el fichero destino. Cada bloque del fichero destino vuelve a cargarse de nuevo en la memoria del ordenador, y es comparado allí con el contenido del fichero fuente.
- W Permite escribir sobre un fichero situado en la unidad destino que tuviera los atributos RO (Read Only, sólo lectura). Normalmente PIP pedirá confirmación al usuario antes de sobrescribir en un fichero con estas características.
- Z Pone a cero el octavo bit (bit de paridad) de los datos del fichero destino.

Para terminar este capítulo, hay que añadir que para abandonar el programa PIP y retornar al sistema operativo, basta con pulsar CTRL-C (ALT-C, en el PCW) o RETURN tras la aparición del asterisco inductor de órdenes.

•

2.10.1. Muestra de opciones PIP.

```

A>pip
CP/M 3 PIP VERSION 3.0
*#:#.#.#[c]

COPYING -
GULI (Y/N)? y
LIS-CORT (Y/N)? n
NORMAS (Y/N)? n
CODIGOS (Y/N)? y
TELEF (Y/N)? y
FACTURA (Y/N)? n
SP-CARG. BAS (Y/N)? y
SP-SOFT. BIN (Y/N)? y
DISC. BAS (Y/N)? n
ANSWORD. BIN (Y/N)? n
PAGOS (Y/N)? n
INFORMAT (Y/N)? y
REL. P (Y/N)? y
CARTA (Y/N)? n
SABOTEUR (Y/N)? y
CIRCLEJ (Y/N)? n
RPMAYO. BAK (Y/N)? y
RPMAYO (Y/N)? y
RPMAYDA (Y/N)? y
SPITFIRE (Y/N)? n
O-MONTA (Y/N)? n
BANSAN (Y/N)? y
TONIN (Y/N)? n
•

```

```
A>pip
CP/M 3 PIP VERSION 3.0
*m:=codigos{e}
```

```
TU MICRO AMSTRAD
```

```
COD.1
```

```
TU MICRO AMSTRAD
SECCION: Comunicado interno
TITULO: CODIGOS DE SECCION
```

```
XXX.1
```

```
CODIGOS DE SECCION
```

Seccion	Codigo
SUMARIO	SUM
AL DIA	A.D
CONCURSO	CON
EN LA CUMBRE	E.C
RASTRO	RAS
EL CARTERO	CAR
A TOPE	A.T
AMSWARE	AMS
TECLEANDO	TEC
TALLER	TAL
PASO A PASO	P.P
RET	RET
BASICO	BAS
RANDOMIZE	RAN
MULTISOFT	MUL
AULA INFORMATICA	AIN
PUCHO Y FARADIO	PUF
EL PROFESIONAL	PRO
COMUNICADO INTERNO	XXX

```
A>pip
CP/M 3 PIP VERSION 3.0
*m:=codigos{l}
*
```

```
A>type m:codigos{nopage}
```

```
tu micro amstrad
```

```
cod.1
```

```
tu micro amstrad
seccion: comunicado interno
titulo: codigos de seccion
```

```
xxx.1
```

```
codigos de seccion
```

seccion	codigo
sumario	sum
al dia	a.d
concurso	con
en la cumbre	e.c
rastros	ras
el cartero	car
a tope	a.t
amsware	ams
tecleando	tec
taller	tal
paso a paso	p.p
ret	ret
basico	bas
randomize	ran
multisoft	mul
aula informatica	ain
pucho y faradio	puf
el profesional	pro
comunicado interno	xxx

```
A>pip
CP/M 3 PIP VERSION 3.0
*m:=codigos(n)
*
```

```
A>type m:codigos(nopage)
1:
```

```
2: TU MICRO AMSTRAD COD.1
```

```
3:
4: TU MICRO AMSTRAD XIX.1
```

```
5: SECCION: Comunicado interno
```

```
6: TITULO: CODIGOS DE SECCION
```

```
7:
```

```
8:
```

```
9:
```

```
CODIGOS DE SECCION
```

```
10:
```

```
11:
```

```
12:
```

```
13:
```

```
14:
```

```
15:
```

```
16:
```

```
17:
```

```
18:
```

```
19:
```

```
20:
```

```
21:
```

```
22:
```

```
23:
```

```
24:
```

```
25:
```

```
26:
```

```
27:
```

```
28:
```

```
29:
```

```
30:
```

```
31:
```

```
32:
```

```
Seccion ----- Codigo
```

```
SUMARIO SUM
```

```
AL DIA A.D
```

```
CONCURSO CON
```

```
EN LA CUMBRE E.C
```

```
RASTRO RAS
```

```
EL CARTERO CAR
```

```
A TOPE A.T
```

```
AMSWARE AMS
```

```
TECLEANDO TEC
```

```
TALLER TAL
```

```
PASO A PASO P.P
```

```
RET RET
```

```
BASICO BAS
```

```
RANDOMIZE RAN
```

```
MULTISOFT MUL
```

```
AULA INFORMATICA AIN
```

```
FUCHO Y FARADIO PUF
```

```
EL PROFESIONAL PRO
```

```
COMUNICADO INTERNO XXX
```

```
A>pip
CP/M 3 PIP VERSION 3.0
*m:=codigos('TITULO'ZQP.P'Z)
*
```

```
A>type m:codigos
TITULO: CODIGOS DE SECCION
```

```
CODIGOS DE SECCION
```

```
Seccion ----- Codigo
```

```
SUMARIO SUM
```

```
AL DIA A.D
```

```
CONCURSO CON
```

```
EN LA CUMBRE E.C
```

```
RASTRO RAS
```

```
EL CARTERO CAR
```

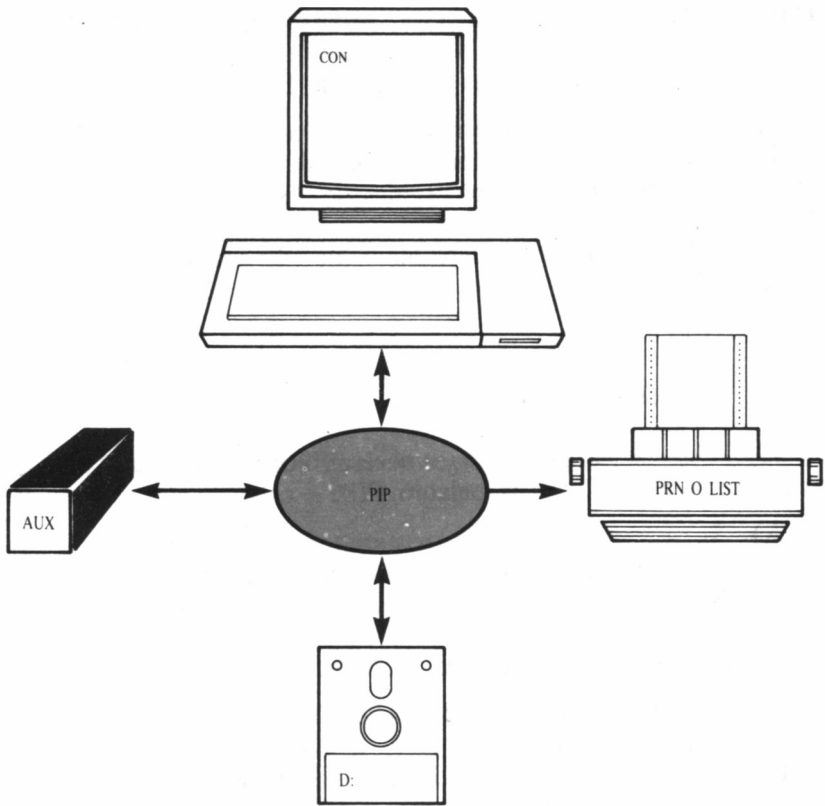
```
A TOPE A.T
```

```
AMSWARE AMS
```

```
TECLEANDO TEC
```

```
TALLER TAL
```

```
PASO A PASO P.P
```



2.10.2. Posibles periféricos para PIP.

EJERCICIOS

75. ¿Cuál es el objetivo principal de PIP?
- A) Intercambio de información con los periféricos.
 - B) Emitir un sonido similar al pío del pollo.
 - C) Obtener copias desprotegidas de discos.
76. Al transferir un fichero mediante PIP ¿queda la copia desprotegida?
- A) No.
 - B) Sí.
 - C) Sólo ante la Ley.
77. Una vez cargado PIP ¿es necesario comenzar todas las órdenes destinadas a él con la palabra PIP?
- A) Sí.
 - B) Sí y deben ser pronunciadas con una pipa en la boca.
 - C) No.
78. ¿Cuál será el efecto de la orden PIP LST:=a:fichero.txt?
- A) Transferir el contenido de «fichero.txt» situado en la Unidad A a la impresora.
 - B) Transferir el contenido de «fichero.txt» situado en la unidad por defecto a la impresora.
 - C) Confundir enormemente al ordenador.
79. ¿Cuántos ficheros se puede llegar a transferir con una sola orden PIP?
- A) Todos los del disco.
 - B) Sólo uno.
 - C) Todos los que se dejen.
80. ¿Qué efecto tiene la orden PIP b:=a:*. *?*
- A) Transferir todos los ficheros situados en la unidad A a la unidad B.
 - B) Los símbolos comodín no están permitidos en las órdenes PIP.
 - C) Desconocido por el momento.

81. ¿Mediante qué opción podemos conseguir que PIP requiera confirmación por parte del usuario para la transferencia de un fichero?

- A) C.
- B) TRANSCONUSU.
- C) U.

82. ¿Con qué opción se señala el área de usuario para los ficheros fuente y destino de la transferencia?

- A) U.
- B) FDGUSER.
- C) G.

83. ¿Podemos con PIP, sin especificar ninguna opción, copiar un fichero con el atributo SYS?

- A) Sí, los SYS se transfieren como los DIR.
- B) No, sólo con la opción R.
- C) A veces, según y como, en fin... ¡ni idea!

84. ¿Cómo se abandona el programa PIP?

- A) Sólo con CONTROL-C.
- B) Con CONTROL-C o RETURN tras la aparición del *prompt*.
- C) Sin consideración.

SET: BAJO LA PROTECCIÓN DE CP/M



SET es uno de los programas transitorios de CP/M más potentes y flexibles. Son muchos los trabajos que pueden efectuarse con esta poderosa herramienta; desde proteger discos o ficheros mediante una palabra clave, la cual prohíbe el acceso al usuario que la desconozca, hasta modificar los atributos del sistema asignados a cada fichero o unidad. Su campo de acción no se limita sólo a estas actividades, sino que es mucho más amplio como pronto comprobaremos.

A lo largo del libro, en bastantes ocasiones se ha hablado de «atributos» del sistema. Pero, ¿qué son los atributos? Pues son simplemente una determinada información adicional directamente unida al fichero y cuyo objetivo es provocar en éste un efecto específico. Por ejemplo, hemos comentado que algunos ficheros pueden tener el atributo RO (sólo lectura), queriendo especificar que sobre tales ficheros solamente están permitidas las operaciones de lectura, siendo imposible alterar su contenido al intentar escribir en ellos.

ATRIBUTOS DEL SISTEMA

Antes de profundizar en las posibilidades del comando SET, con-

viene conocer cuáles son los atributos que podemos asignarle a un determinado fichero sobre el disco, a veces, incluso a la totalidad de éste. Algunos nos serán ya familiares de los capítulos anteriores, pero en cualquier caso he aquí la lista completa:

RO Define a un fichero con el que sólo se pueden mantener operaciones de lectura.

RW Los ficheros que lo tengan asignado, permiten operaciones tanto de lectura como de escritura.

Estos dos atributos pueden especificarse tanto a un fichero o grupo de ficheros, como una unidad de disco determinada.

SYS Los así marcados, no son visualizados en la pantalla al realizar un directorio normal del disco.

DIR Permite que un fichero que lo tenga definido sea visualizado al efectuar un directorio del disco.

ARCHIVE=OFF Significa que el fichero que lo contenga, todavía no ha sido copiado (archivado).

ARCHIVE=ON Informa al usuario y al sistema que el fichero implicado ya fue copiado.

Los atributos de «archivado» pueden ser modificados además de por el comando SET, por la orden PIP, cuando se transfiere un grupo de ficheros indicando la opción [A] de este programa. (Ver todas las opciones de PIP). Este tipo de atributo puede visualizarse manejando los comandos SHOW y DIR.

F1=ON:OFF Bascula activando o desactivando el atributo definible por el usuario F1.

F2=ON:OFF Bascula activando o desactivando el atributo definible por el usuario F2.

F3=ON:OFF Bascula activando o desactivando el atributo definible por el usuario F3.

F4=ON:OFF Bascula activando o desactivando el atributo definible por el usuario F4.

Este tipo de atributos pueden visualizarse en la pantalla ejecutando la orden DIR con la opción [ATT], y para asignarlos se sigue una de las dos formas sintácticas siguientes, según se trate de la unidad de disco, o de ficheros:

SET d: [atributo]

SET d: nombrefich[atributo]

ETIQUETANDO UN DISCO

Cuando se trabaja con muchos discos a la vez, es muy posible que dos de ellos contengan los mismos ficheros, y aunque similares en el nombre, puede suceder que sus contenidos sean muy diferentes. Para poder distinguirlos se puede emplear la orden SET de una forma particular para que grave en el directorio del disco un identificativo o etiqueta (*label*). Esta sigue las mismas normas que el nombre de un fichero en cuanto a la longitud de su nombre y extensión (ocho caracteres para el nombre y tres para la extensión, separados por un punto).

La sintaxis que debe respetar una orden de este tipo es:

```
SET d:[NAME=nombeti.tip]
```

donde d, representa la unidad en la que se encuentra el disco que se desea etiquetar, y «nombeti.tip» el nombre dado a la etiqueta y su extensión. El nombre asignado a un disco, una vez etiquetado éste, no puede verse al efectuar el directorio del mismo.

PROTECCIONES

Además de asignarle un nombre al disco, también es posible definir una palabra clave de manera que sólo quien la conozca tenga acce-

```
A>set {protect=on}
Label for drive A:
Directory  -  Passwds  Stamp  Stamp  Stamp
Label      Reqd    Create  Access  Update
-----
A:LABEL    .      on      off     off     off

A>set {name=libro2.gba}
Label for drive A:
Directory  Passwds  Stamp  Stamp  Stamp
Label      Reqd    Create  Access  Update
-----
A:LIBRO2   .GBA    on      off     off     off
```

2.11.1. Atributos SET.

so a los ficheros almacenados en él. Las órdenes SET que controlan esta opción son las siguientes:

```
SET [PASSWORD=palabra clave]
SET [PASSWORD=<cr>]
```

En el primer caso, cada vez que pretendamos efectuar un acceso al disco protegido, el sistema operativo nos reclama la palabra clave que hayamos seleccionado. Es muy importante no olvidarla por motivos obvios. Además, mientras la tecleamos aparentemente no está sucediendo nada, puesto que se nos muestran en la pantalla los caracteres pulsados.

En el segundo, se define la palabra clave como la pulsación de RETURN o INTRO, y será esto lo que habrá que efectuar cuando el sistema la reclame.

Pero no sólo se puede proteger la etiqueta del disco, además es posible asignarle una clave a cada fichero individual, de forma que unos puedan estar protegidos y los otros no. Incluso la palabra clave (password) puede ser diferente para cada uno de los ficheros.

Antes de definir claves de acceso es necesario ejecutar otra orden SET. Se trata de

```
SET [PROTECT=ON]
```

```
A>set p*.*(password=carlos)
A: PALETTE .COM Protection = READ, Password = CARLOS
A: PAPER .COM Protection = READ, Password = CARLOS
A: PIP .COM Protection = READ, Password = CARLOS
A: PROFILE .ENG Protection = READ, Password = CARLOS
A>set language.com(password=fernando)
A: LANGUAGE.COM Protection = READ, Password = FERNANDO
A>set language.com(protect=delete)
A: LANGUAGE.COM Protection = DELETE
A>era language.com
A: LANGUAGE.COM Not erased, Password Error
Password:
A: LANGUAGE.COM Not erased, Password Error
A>dir language.com
A: LANGUAGE.COM
A>era language.com
A: LANGUAGE.COM Not erased, Password Error
Password:
A>dir language.com
No File
```

2.12.1. Parámetros SHOW.

la cual activa la protección por claves para todos los ficheros sobre el disco que así lo tuvieran establecido, o permite a partir de ese momento asignárselas.

Naturalmente, la orden anterior admite la posibilidad de desactivarse mediante

SET [PROTECT=OFF]

siempre y cuando recordemos la clave de acceso al disco, pues inevitablemente el sistema la pedirá si es que estaba, previamente definida.

Una vez activada la función de protección, asignarle a un fichero o un grupo de estos una palabra clave es de lo más sencillo mediante la orden:

SET d: nombrefich[PASSWORD=palabra clave]

Como es norma habitual en CP/M, «nombrefich» admite la inclusión de símbolos comodín, por si queremos asignar a varios ficheros a la vez la misma palabra clave.

OTROS MODOS DE PROTECCIÓN

Adicionalmente a la asignación de claves para el acceso a los ficheros, también el comando SET contempla la posibilidad de protegerlos automáticamente bajo determinadas condiciones. Para ello se ordena a CP/M que ejecute el siguiente comando:

SET d:nombrefich[PROTECT=modo]

donde «modo» representa alguna de las opciones siguientes:

- | | |
|--------|---|
| READ | La palabra clave es solicitada cada vez que se pretenda leer, copiar, escribir, borrar o cambiar de nombre al fichero. |
| WRITE | En este caso, la clave se pide para escribir, borrar, o renombrar el fichero, no siendo necesaria cuando lo que se pretende es leerlo. |
| DELETE | Solamente se solicita la palabra clave cuando se trate de realizar operaciones de borrado o cambio de nombre en el fichero, lo cual asegura que no se perderá su contenido en un borrado accidental, por ejemplo. |
| NONE | Selecciona que no exista palabra clave para el fichero o grupo de estos afectado. Si la había previamente a esta operación, la orden SET la elimina del fichero. |

FECHA Y HORA

Todavía SET puede realizar algunas operaciones más. Además de todo lo visto anteriormente, el usuario de un determinado fichero puede mantener periódicamente un control estadístico sobre datos como, cuándo fue creado, cuándo se actualizó por última vez, o en qué fecha y hora se realizó la última utilización del mismo.

Pero antes de tener acceso a estas nuevas posibilidades del comando SET, será preciso que ejecute el programa INITDIR. Este se encarga de reorganizar el directorio del disco de manera que se reserve el espacio suficiente para alojar la fecha y hora a anotar en cada fichero.

Para la estampación del sello de fecha y hora (timestamps), SET dispone de las siguientes opciones:

CREATE=ON Activa la función de estampación de fecha sobre el directorio del disco almacenado en la unidad señalada. Para almacenar la fecha de creación de un determinado fichero, esta opción ha de encontrarse activa, antes de que sea grabado en el disco.

ACCESS=ON Habilita la función de sello de fecha del último acceso a los ficheros situados en la unidad implicada. **ACCESS** y **CREATE** son opciones mutuamente excluyentes; solamente una de ellas puede estar activa a la vez. Si en una unidad que tuviera activa la anotación de fecha de creación de los ficheros, se habilita la opción de último acceso, la de fecha de creación se desactiva automáticamente.

UPDATE=ON Es la encargada de dar entrada a la opción que permite marcar un fichero con la fecha y hora de la última modificación realizada sobre él.

Debe tenerse en cuenta que si las opciones de fecha de creación y última actualización están activas simultáneamente, cada vez que se modifique un fichero será anotada, y se considerará como nueva por el sistema, y esta fecha de actualización será la considerada también, como fecha de creación.

Para visualizar las fechas registradas en el directorio de un disco, basta con efectuar un catálogo del mismo mediante la orden **DIR [FULL]**

B>initdir a:

INITDIR WILL ACTIVATE TIME STAMPS FOR SPECIFIED DRIVE.
Do you want to re-format the directory on drive: A (Y/N)? y

B>date set

Enter today's date (MM/DD/YY): 08/26/86
Enter the time (HH:MM:SS): 22:28:00
Press any key to set time

B>date
Tue 08/26/86 22:29:12

B>set a:[create=on]

Label for drive A:

Directory Label	Passwds Reqd	Stamp Create	Stamp Access	Stamp Update
A:LABEL	off	on	off	off

B>set a:[name=libro2.gba]

Label for drive A:

Directory Label	Passwds Reqd	Stamp Create	Stamp Access	Stamp Update
A:LIBRO2 .GBA	off	on	off	off

A>dir [full]

Scanning Directory...

Sorting Directory...

Directory For Drive A: User 0

Name	Bytes	Recs	Attributes	Prot	Update	Access
BASIC	COM	28k	224 Dir RW	None	08/27/86 09:11	08/27/86 09:11
DIR	COM	15k	114 Dir RW	None		08/27/86 09:12
DISCKIT	COM	7k	56 Dir RW	None		08/26/86 22:35
ED	COM	10k	73 Dir RW	None		08/26/86 22:35
ERASE	COM	4k	29 Dir RW	None		08/26/86 22:35
J14SCPM3	EMS	40k	320 Dir RW	None		08/26/86 22:34
KEYS	WP	1k	7 Dir RW	None		08/26/86 22:37
LANGUAGE	COM	1k	8 Dir RW	None		08/27/86 09:11
PALETTE	COM	1k	8 Dir RW	None		08/26/86 22:35
PAPER	COM	2k	16 Dir RW	None		08/26/86 22:35
PIP	COM	9k	68 Dir RW	None		08/27/86 09:11
PROFILE	ENG	1k	2 Dir RW	None		08/26/86 22:37
RENAME	COM	3k	23 Dir RO	None		08/26/86 22:35
RPED	BAS	7k	56 Dir RO	None		08/26/86 22:37
RPED	SUB	1k	1 Dir RO	None		08/26/86 22:37
SET	COM	11k	81 Dir RW	None		08/27/86 09:10
SET24X80	COM	1k	8 Dir RW	None		08/26/86 22:36
SETDEF	COM	4k	32 Dir RW	None		08/26/86 22:36
SETKEYS	COM	2k	16 Dir RW	None		08/26/86 22:36
SETLST	COM	2k	16 Dir RW	None		08/26/86 22:36
SETSIO	COM	2k	16 Dir RW	None		08/26/86 22:36
SHOW	COM	9k	66 Dir RW	None		08/27/86 09:04
SUBMIT	COM	6k	42 Dir RW	None		08/26/86 22:36
TYPE	COM	3k	24 Dir RW	None		08/26/86 22:37

Press RETURN to Continue

Total Bytes = 170k Total Records = 1306 Files Found = 24
Total 1k Blocks = 170 Used/Max Dir Entries For Drive A: 37/ 64

B>set a:(access=on update=on)

Label for drive A:

Directory Label	Passwds Reqd	Stamp Create	Stamp Access	Stamp Update
A:LIBR02	.GBA	off	off	on

B>a:

A>set r*.*(ro)

A:RENAME .COM set to directory (DIR), Read Only (RO)
A:RPED .BAS set to directory (DIR), Read Only (RO)
A:RPED .SUB set to directory (DIR), Read Only (RO)

A>language 0

A>pip a:=b:basic.com

```
B>pip a:=b:*.*(k)
```

```
B>dir a:[full]
```

```
Scanning Directory...
```

```
Sorting Directory...
```

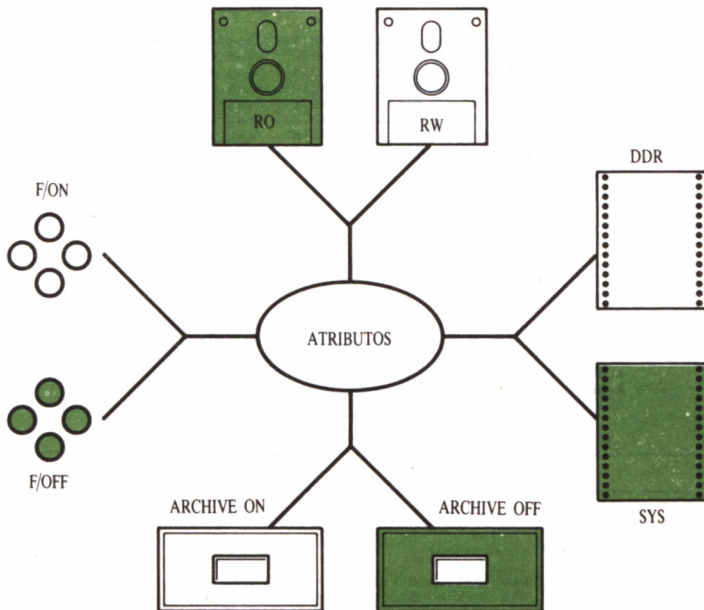
```
Directory For Drive A: User 0
```

Name	Bytes	Recs	Attributes	Prot	Update	Create
BASIC	COM	28k	224 Dir RW	None		08/26/86 22:34
DIR	COM	15k	114 Dir RW	None		08/26/86 22:35
DISCKIT	COM	7k	56 Dir RW	None		08/26/86 22:35
ED	COM	10k	73 Dir RW	None		08/26/86 22:35
ERASE	COM	4k	29 Dir RW	None		08/26/86 22:35
J14SCPM3	EMS	40k	320 Dir RW	None		08/26/86 22:34
KEYS	WP	1k	7 Dir RW	None		08/26/86 22:37
LANGUAGE	COM	1k	8 Dir RW	None		08/26/86 22:35
PALETTE	COM	1k	8 Dir RW	None		08/26/86 22:35
PAPER	COM	2k	16 Dir RW	None		08/26/86 22:35
PIP	COM	9k	68 Dir RW	None		08/26/86 22:35
PROFILE	ENG	1k	2 Dir RW	None		08/26/86 22:37
RENAME	COM	3k	23 Dir RW	None		08/26/86 22:35
RPED	BAS	7k	56 Dir RW	None		08/26/86 22:37
RPED	SUB	1k	1 Dir RW	None		08/26/86 22:37
SET	COM	11k	81 Dir RW	None		08/26/86 22:36
SET24X80	COM	1k	8 Dir RW	None		08/26/86 22:36
SETDEF	COM	4k	32 Dir RW	None		08/26/86 22:36
SETKEYS	COM	2k	16 Dir RW	None		08/26/86 22:36
SETLST	COM	2k	16 Dir RW	None		08/26/86 22:36
SETSIO	COM	2k	16 Dir RW	None		08/26/86 22:36
SHOW	COM	9k	66 Dir RW	None		08/26/86 22:36
SUBMIT	COM	6k	42 Dir RW	None		08/26/86 22:36
TYPE	COM	3k	24 Dir RW	None		08/26/86 22:37

```
Press RETURN to Continue
```

```
Total Bytes = 170k Total Records = 1306 Files Found = 24
```

```
Total 1k Blocks = 170 Used/Max Dir Entries For Drive A: 37/ 64
```



2.11.1. Atributos SET.

2.11.5. Gráfico atributos SET DOC.27 (yo).

A>dir [att]

Scanning Directory...

Sorting Directory...

Directory For Drive A: User 0

Name	Bytes	Recs	Attributes	Name	Bytes	Recs	Attributes
AMSDOS	COM	1k	8 Dir RW	BANKMAN	BAS	1k	7 Dir RW A
BANKMAN	BIN	2k	12 Dir RW A	CIOCPM3	EMS	25k	200 Dir RW
DATE	COM	3k	23 Dir RW 3	DEVICE	COM	8k	58 Dir RW 3
DIR	COM	15k	114 Dir RW 3	DISCKI13	COM	6k	48 Dir RW 3
ED	COM	10k	73 Dir RW	ERASE	COM	4k	29 Dir RW
GET	COM	7k	51 Dir RW	KEYS	CCP	1k	3 Dir RW
KEYS	WP	1k	3 Dir RW	LANGUAGE	COM	1k	8 Dir RW
PALETTE	COM	1k	8 Dir RW	PIP	COM	9k	68 Dir RW
PROFILE	ENG	1k	1 Dir RW	PUT	COM	7k	55 Dir RW
RENAME	COM	3k	23 Dir RW	SET	COM	11k	81 Sys RO
SET24X80	COM	1k	8 Sys RO	SETDEF	COM	4k	32 Sys RO
SETKEYS	COM	2k	16 Sys RO	SETLST	COM	2k	16 Sys RO
SETSIO	COM	2k	16 Sys RO	SHOW	COM	9k	66 Dir RO
SUBMIT	COM	6k	42 Dir RO	TYPE	COM	3k	24 Dir RW

Total Bytes = 146k Total Records = 1093 Files Found = 28
 Total 1k Blocks = 146 Used/Max Dir Entries For Drive A: 29/ 64

A>set s.*[ro]

A:SUBMIT .COM set to directory (DIR), Read Only (RO)
 A:SETKEYS .COM set to directory (DIR), Read Only (RO)
 A:SET24X80.COM set to directory (DIR), Read Only (RO)
 A:SETSIO .COM set to directory (DIR), Read Only (RO)
 A:SETLST .COM set to directory (DIR), Read Only (RO)
 A:SHOW .COM set to directory (DIR), Read Only (RO)
 A:SET .COM set to directory (DIR), Read Only (RO)
 A:SETDEF .COM set to directory (DIR), Read Only (RO)

A>set set.*[sys]

A:SETKEYS .COM set to system (SYS), Read Only (RO)
 A:SET24X80.COM set to system (SYS), Read Only (RO)
 A:SETSIO .COM set to system (SYS), Read Only (RO)
 A:SETLST .COM set to system (SYS), Read Only (RO)
 A:SET .COM set to system (SYS), Read Only (RO)
 A:SETDEF .COM set to system (SYS), Read Only (RO)

A>set b.*[archive=on]

A:BANKMAN .BAS set to directory (DIR), Read Write (RW) A
 A:BANKMAN .BIN set to directory (DIR), Read Write (RW) A

```
A>set d.*[f3-on]
```

```
A:DISCKIT3.COM set to directory (DIR), Read Write (RW) 3  
A:DATE .COM set to directory (DIR), Read Write (RW) 3  
A:DEVICE .COM set to directory (DIR), Read Write (RW) 3  
A:DIR .COM set to directory (DIR), Read Write (RW) 3
```

```
A>dir [att]
```

```
Scanning Directory...
```

```
Sorting Directory...
```

```
Directory For Drive A: User 0
```

Name	Bytes	Recs	Attributes	Name	Bytes	Recs	Attributes
AMSDOS	COM	1k	8 Dir RW	BANKMAN	BAS	1k	7 Dir RW
BANKMAN	BIN	2k	12 Dir RW	CIOCPM3	EMS	25k	200 Dir RW
DATE	COM	3k	23 Dir RW	DEVICE	COM	8k	58 Dir RW
DIR	COM	15k	114 Dir RW	DISCKIT3	COM	6k	48 Dir RW
ED	COM	10k	73 Dir RW	ERASE	COM	4k	29 Dir RW
GET	COM	7k	51 Dir RW	KEYS	CCP	1k	3 Dir RW
KEYS	WP	1k	3 Dir RW	LANGUAGE	COM	1k	8 Dir RW
PALETTE	COM	1k	8 Dir RW	PIP	COM	9k	68 Dir RW
PROFILE	ENG	1k	1 Dir RW	PUT	COM	7k	55 Dir RW
RENAME	COM	3k	23 Dir RW	SET	COM	11k	81 Dir RW
SET24X80	COM	1k	8 Dir RW	SETDEF	COM	4k	32 Dir RW
SETKEYS	COM	2k	16 Dir RW	SETLST	COM	2k	16 Dir RW
SETSIO	COM	2k	16 Dir RW	SHOW	COM	9k	66 Dir RW
SUBMIT	COM	6k	42 Dir RW	TYPE	COM	3k	24 Dir RW

```
Total Bytes = 146k Total Records = 1093 Files Found = 28  
Total 1k Blocks = 146 Used/Max Dir Entries For Drive A: 29/ 64
```

EJERCICIOS

85. ¿A qué se denomina *password*?
- A) A la palabra clave de acceso a un fichero.
 - B) Al sistema de protección contra borrado de un fichero.
 - C) En inglés, «passwato».
86. ¿Qué es un atributo?
- A) Un «atributo» que no sabe pronunciar la erre.
 - B) Determinada información adicional directamente unida a un fichero.
 - C) La clave de acceso a un fichero.
87. ¿Existe algún atributo que pueda asignarse a todo un disco?
- A) Sí.
 - B) Sólo a los Long-Play.
 - C) No.
88. ¿Con qué atributo protegeríamos un fichero contra su escritura accidental?
- A) Con un plástico.
 - B) RW.
 - C) RO.
89. ¿Cómo se puede evitar que un fichero sea visualizado en un directorio por DIR?
- A) Empleando el atributo SYS.
 - B) Empleando el atributo RO.
 - C) Poniendo al usuario una venda en los ojos.
90. ¿Cómo protegeríamos mediante la clave de acceso GBA el fichero de CAP. III que se encuentra en la unidad B?
- A) SET B:CAP.III[GBA]
 - B) SET B:CAP.III[PASSWORD=GBA]
 - C) ¡Como no diga GBA, aquí no entra nadie! ¿Enterado?

91. En un disco cuyos ficheros tengan activados los atributos de estampación de fechas ¿qué operación realizaríamos para conocer dichas fechas?

- A) DIR[FULL].
- B) Un buril y decapante.
- C) DIR.

COMO COMPLEMENTO DE SET



Como se ha podido comprobar en el capítulo anterior, la facilidad para asignar a los ficheros determinados parámetros de protección o incluir en el directorio datos que permitan controlar su nivel de utilización estadísticamente, contribuye de una forma, más que notable, a mejorar el rendimiento que del sistema puede obtener el usuario.

Sin embargo, ciertas opciones de este potente comando sólo se consiguen activar si previamente nos hemos servido de alguna otra orden de CP/M (tal es el caso de INITDIR), o los resultados sólo podrán visualizarse apoyándose en otra orden transitoria (SHOW), incluso todavía algún lector quizás siga preguntándose como se le informa al sistema de la fecha actual (programa DATE).

INITDIR

El comando INITDIR, como adelantábamos en el capítulo anterior, reorganiza o en su caso inicializa el directorio de un disco para permitir las estampaciones de los sellos de fecha y hora relacionados con cada fichero, reservando para ello el espacio necesario.

La sintaxis de la orden es la siguiente:

INITDIR d:

siendo d, la unidad en la que está ubicado el disco en que se va a reformatear su directorio. Tras esto, el programa presenta el mensaje:

INITDIR WILL ACTIVATE TIME-STAMPS FOR SPECIFIED
DRIVE (Y/N)

INITDIR activará el registro de la fecha para la unidad especificada, traducido al castellano. Si contestamos afirmativamente (Y), el directorio será reorganizado, mientras que, en caso contrario, no se producirá ninguna acción sobre éste. La reorganización no significa la pérdida de la información contenida en el disco, aunque INITDIR comprobará si dispone de espacio suficiente para llevarla a cabo.

CONOCER EL ESPACIO DISPONIBLE

El comando transitorio SHOW se encarga de mostrar información de diversa índole sobre algunas características del disco. Su sintaxis responde al siguiente esquema:

SHOW {d:} {[SPACE | LABEL | USERS | DIR | DRIVE]}

en los que los parámetros alternativos que configuran las opciones, siguen los significados que a continuación se detallan:

- | | |
|-------|---|
| SPACE | Muestra el espacio disponible en Kbytes sobre cada unidad identificada por el sistema, y el tipo de acceso permitido. |
| LABEL | Imprime la etiqueta del directorio de la unidad especificada, si es que existe ésta. |
| USERS | Muestra el número de todas las zonas de usuario definidas sobre el disco, el número de ficheros existentes en cada área, y la cantidad de reseñas libres para el directorio del disco especificado en la orden. |
| DIR | Muestra el número de reseñas todavía disponibles para el directorio de la unidad (el máximo en discos de densidad sencilla es de 64, mientras que los de doble, para la segunda boca del PCW 8512 admiten 256 anotaciones). |
| DRIVE | Emite una lista de las características de la unidad señalada. |

A>show

A: RW, Space: 3k
B: RW, Space: 396k

A>show {space}

A: RW, Space: 3k

A>show {label}

Label for drive A:

Directory Label	Passwds Reqd	Stamp Access	Stamp Update	Label Created	Label Updated
LIBRO2	.GBA	off	on	08/26/86 22:31	08/27/86 09:10

A>show {dir}

A: Number of time/date directory entries: 16
A: Number of free directory entries: 20

A>show b:{dir}

B: Number of free directory entries: 231

A>show b:{users}

B: Active User : 0
B: Active Files: 0 8
B: # of files : 27 7

B: Number of free directory entries: 26

A>show {drive}

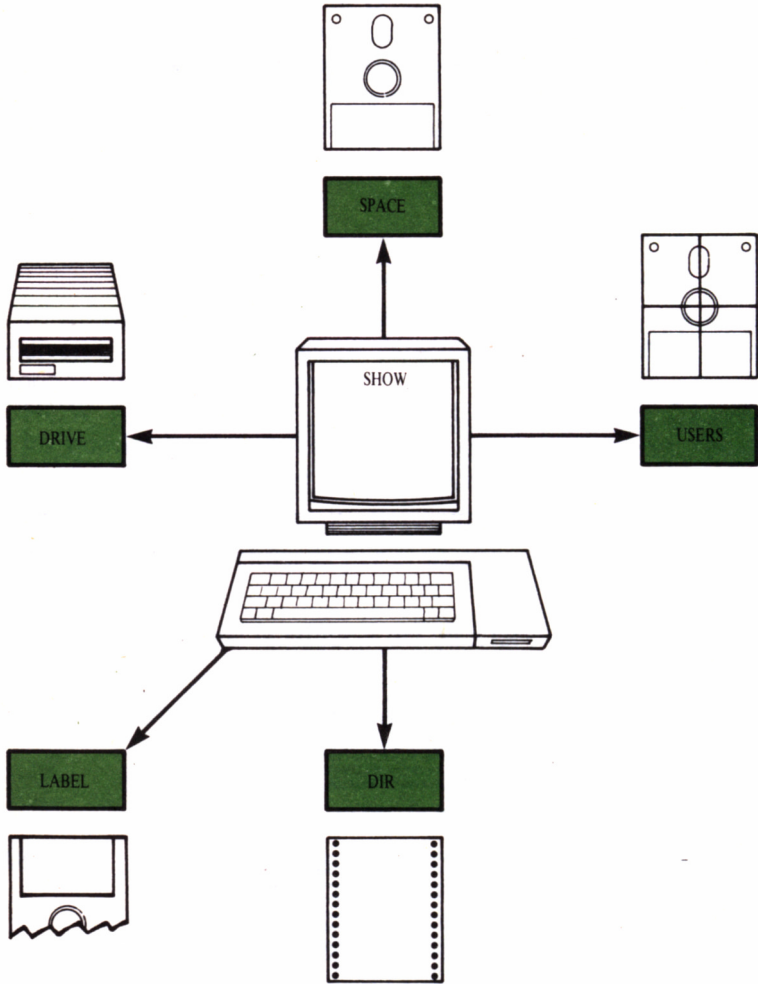
A: Drive Characteristics
1,409: 128 Byte Record Capacity
175: Kilobyte Drive Capacity
64: 32 Byte Directory Entries
64: Checked Directory Entries
128: Records / Directory Entry
8: Records / Block
36: Sectors / Track
1: Reserved Tracks
512: Bytes / Physical Record

A>show b:{drive}

B: Drive Characteristics
5,712: 128 Byte Record Capacity
714: Kilobyte Drive Capacity
256: 32 Byte Directory Entries
128: Checked Directory Entries
128: Records / Directory Entry
16: Records / Block
36: Sectors / Track
1: Reserved Tracks
512: Bytes / Physical Record

FECHA DEL SISTEMA

Cuando se emplea la opción de registro del sello de fecha y hora como complemento de la información que de un determinado fichero o grupo de éstos se puede obtener, se debe pasar ineludiblemente por la utilización del comando DATE antes de comenzar una sesión de trabajo.



2.12.1. Parámetros SHOW.

Naturalmente, a no ser que le sea comunicada otra cosa, el ordenador no conoce cuál es la fecha de hoy, y si la empleamos para imprimirla como etiqueta de creación, actualización o modificación de un fichero, siempre anotará la misma 12/15/82, como puede comprobarse si tras el prompt de CP/M se teclea la orden DATE (el fichero DATE.COM debe estar almacenado en el disco situado en la unidad por defecto).

Para visualizar la fecha y hora actuales del sistema, se ejecuta la orden DATE C, la cual la muestra continuamente, segundo a segundo.

DATE SET permite introducir la fecha y hora actuales, para que el sistema las considere una vez activadas en cualquier anotación que deba realizar en fichero. Para ello, se sigue la siguiente secuencia de acontecimientos:

A>DATE SET

Enter today's date (MM/DD/YY) (Introduzca la fecha de hoy)

Enter the time (HH:MM:SS) (Introduzca la hora)

Press any key to set time (Pulse cualquier tecla para activar la hora).

Es importante respetar el formato tanto de fecha como de hora solicitado por el sistema. La fecha debe introducirse en el orden mes, día y año, con dos cifras cada uno, y separados por el signo «/». La hora, según la secuencia, horas, minutos y segundos, nuevamente cada campo con dos dígitos y separados por dos puntos (:). El reloj no comienza a contar hasta que no se pulsa una tecla, pudiendo así sincronizarlo.

También está permitido definir la fecha directamente, señalándola a continuación del comando DATE. Por ejemplo, para introducir como fecha actual el veintidós de septiembre de mil novecientos ochenta y seis y, como hora, las siete y veinticinco minutos de la tarde, se tecleará la orden:

A/DATE 09/22/86 19:25:00

EJERCICIOS

92. ¿Cuál es la misión del comando INITDIR?
- A) Inicializar un disco (formatearlo).
 - B) Preparar el directorio para la estampación de fechas.
 - C) Servir como pregunta en un test de conocimientos.
93. ¿Implica la utilización del comando INITDIR la pérdida de la información contenida en el disco?
- A) Sí.
 - B) Sólo de la confidencial.
 - C) No.
94. ¿Cómo imprimíamos la etiqueta del directorio de una determinada unidad?
- A) Con el comando SHOW y la opción LABEL.
 - B) Con el comando SHOW y la opción SANDY.
 - C) Con el comando SHOW y la opción DIR.
95. ¿Qué orden se utilizaría para conocer las características del disco situado en la unidad A?
- A) A>SHOW [DRIVE]
 - B) A>SET [DRIVE]
 - C) Preguntando en la casa discográfica más próxima.
96. ¿Es cierto que el ordenador conserva, aun después de apagado, la última fecha introducida?
- A) Sí, pero sólo hasta la fecha de caducidad.
 - B) No.
 - C) Sí.
97. ¿Mediante qué orden podemos presentar la fecha y hora continuamente en la pantalla?
- A) DATE C.
 - B) Con ninguna.
 - C) Preguntando cada segundo ¿qué hora es?

98. ¿En qué formato indicaríamos la fecha 10 de diciembre de 1986 al comando DATE SET?

- A) 10/12/86.
- B) 12/10/86.
- C) En mayúsculas.

99. ¿Se pone el reloj en funcionamiento cuando pulsamos RETURN tras la introducción de los segundos?

- A) No; es preciso que pulsemos una tecla para la sincronización.
- B) Sí; aguardaremos a pulsar RETURN en el momento de la sincronización.
- C) Depende de si está engrasado el mecanismo de relojería.

100. ¿Puede introducirse la fecha en la misma línea que el comando DATE?

- A) Depende de la línea, la fecha... en fin ¡son tantas y tantas cosas!
- B) Sí.
- C) No.

TABLA DE RESPUESTAS

UNIDADES

	0	1	2	3	4	5	6	7	8	9
0	*	C	B	C	A	C	C	B	C	A
1	A	C	A	A	A	C	C	A	C	C
2	A	B	B	C	C	B	B	B	A	B
3	A	C	C	C	C	B	B	B	C	A
4	A	A	B	B	B	C	A	A	C	A
5	C	B	B	C	B	A	B	A	B	B
6	B	C	C	B	A	B	A	C	B	B
7	C	A	A	B	C	A	A	C	A	A
8	A	A	C	B	B	A	B	A	C	A
9	B	A	B	C	A	A	B	A	B	A
10	B	*	*	*	*	*	*	*	*	*

D
E
C
E
N
A
S

Y

C
E
N
T
E
N
A
S

Manejo de la tabla: En las columnas localizaremos la unidad del número de respuesta a buscar, mientras que en las filas, se encuentran las decenas y centenas (si las hubiera); el punto de intersección entre fila y columna nos proporcionará el resultado.

Por ejemplo: Para hallar la respuesta a la pregunta 72, nos situaremos en la intersección de la fila 7 y la columna 2.

NOTAS



principios de los años setenta, Gary Kildall, uno de los genios legendarios de Silicon Valley, comenzó a comercializar una versión de lo que él mismo denominó CP/M, siglas de *Control Program for Microprocessors* o programa de control para microprocesadores. El éxito alcanzado por aquella versión, le llevó a fundar la Digital Research, hoy en día una de las más importantes firmas a nivel mundial de software de base y aplicaciones.

GRAN BIBLIOTECA
AMSTRAD