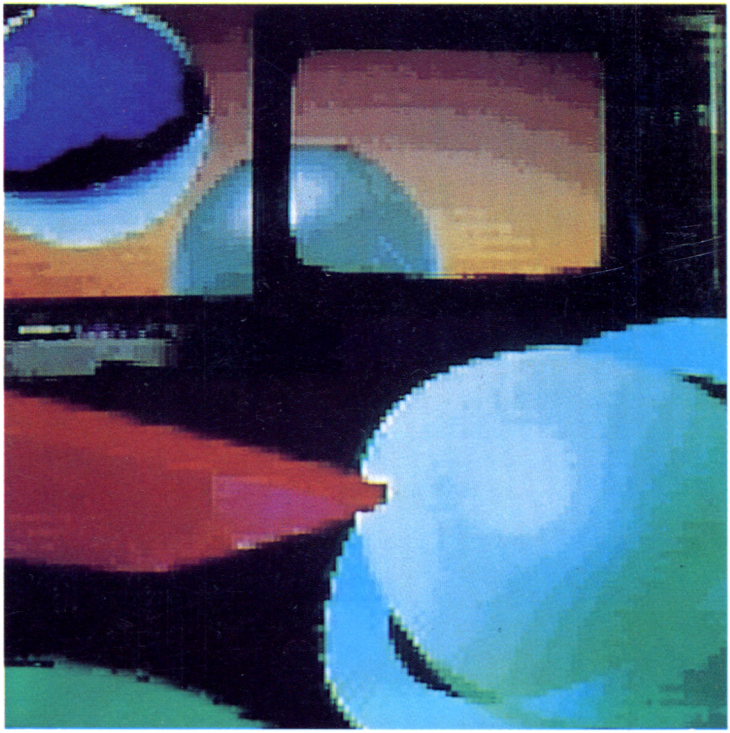
GRAN BIBLIOTECA

AMSTRAD



EL CP/M A FONDO

LA IMPORTANCIA DE UN SISTEMA OPERATIVO

EDICIONES INGELEK

GRAN BIBLIOTECA AMSTRAD

13

EL CP/M A FONDO

Director editor:
Antonio M.ª Ferrer Abelló

Director de producción:
Vicente Robles

Director de la obra:
Fernando López Martínez

Redactor técnico:
Pedro Toledano Sobremazas

Colaboradores:
L-H Servicios Informáticos
Pilar Manzanera Amaro
Carlos de la Ossa Villacañas

Diseño:
Bravo/Lofish

Maquetación:
Carlos González Amezúa

Dibujos:
José Ochoa

Fotografía:
Grupo Gálata

ISBN del tomo: 84-7708-043-7
ISBN de la obra: 84-7708-004-6
Fotocomposición: Andueza, S. A.
Imprime: Eurosur
Depósito Legal: M-12176-1987
Precio en Canarias, Ceuta y Melilla: 435 ptas.
Julio 1987

EL CP/M A FONDO

INTRODUCCIÓN

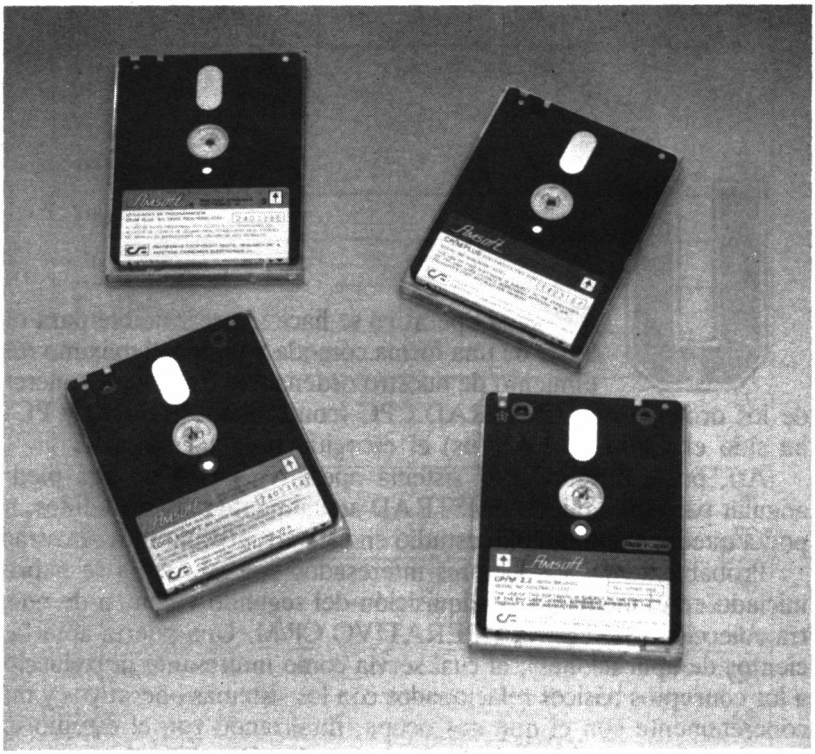
n sistema operativo se hace imprescindible para obtener de una forma cómoda y rápida el máximo rendimiento de nuestro ordenador. En el caso concreto de los ordenadores AMSTRAD CPC (con unidad de disco) y PCW ha sido el CP/M (2.2 ó Plus) el escogido para tal tarea.

Así, pues, tratándose el sistema operativo CP/M de una piedra angular para los equipos AMSTRAD actualmente más difundidos, no podía quedar sin un amplio estudio en esta Gran Biblioteca Amstrad.

Probablemente, los lectores interesados por este tema se habrán iniciado en él mediante la adquisición del segundo volumen de nuestra colección (SISTEMA OPERATIVO CP/M. Una puerta abierta a cientos de aplicaciones), el cual servía como interesante introducción a los conceptos básicos relacionados con los sistemas operativos y más concretamente con el que nos ocupa, finalizando con el estudio exhaustivo de las órdenes de uso más común: Dir, Pip, Set, etc., así como de los programas de utilidad de mayor interés (Ed, Diskit, etc...).

Nos encontramos ahora ante una profundización en el estudio de las herramientas que CP/M pone a nuestra disposición, tratando todas aquellas órdenes y programas de este sistema operativo, concretamente en su versión 3.0 (Plus), que quedaron por tocar en el volumen de esta biblioteca al que anteriormente hacíamos referencia.

No obstante, para aquellos que disponiendo ya de unos sólidos conocimientos sobre el concepto de Sistema Operativo, y dominando las órdenes y programas de ayuda básicos que soporta CP/M, se incorporan por vez primera con este libro al tratamiento de este tema en la colección, comenzaremos por repetir el capítulo en el cual se describían los convenios y signos que van a ser empleados a lo largo del volumen en la sintaxis de las órdenes para su entera comprensión.



13.0.1 Los discos del sistema contienen una versión del CP/M Plus, adaptada a los ordenadores Amstrad.

```
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
A>dir
A: AMSDOS   COM : SID      COM : DUMP     COM : GENCOM   COM : HEXCOM   COM
A: HIST     UTL : INITDIR  COM : LIB      COM : LINK     COM : MAC      COM
A: PATCH    COM : RMAC     COM : SAVE     COM : TRACE    UTL : XREF     COM
A: DD-DMP1  PRL : DDSHINWA PRL : DDHP7470 PRL : ASM      COM
A>dir
A: AMSDOS   COM : HELP     COM : HELP     HLP : SUBMIT   COM : SETKEYS  COM
A: KEYS     CCP : KEYS     DRL : LOGO3    SUB : GSX      SYS : GENGRAF  COM
A: DDMODE2  PRL : DDFXLR7  PRL : DDMODE1  PRL : DDMODE0  PRL : ASSIGN   SYS
A: DRIVERS  GSX : LOGO3    COM
```

13.0.2. Directorios de los discos del sistema de CPC.

```
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
A: DATE     COM : DDHP7470 PRL : DEVICE   COM : DUMP     COM : GENCOM   COM
A: GET      COM : HEXCOM   COM : HIST     UTL : INITDIR  COM : LIB      COM
A: LINK     COM : MAC      COM : PALETTE  COM : PATCH    COM : PUT      COM
A: RMAC     COM : SAVE     COM : SID      COM : TRACE    UTL : XREF     COM
A>dir
A: ASSIGN   SYS : DDFXHR8  PRL : DDFXLR8  PRL : DDSCREEN PRL : GENGRAF  COM
A: GSX      SYS : HELP     COM : HELP     HLP : KEYS     DRL : LOGO     COM
A: LOGO     SUB
```

13.0.3. Directorios de los discos del sistema de PCW.

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, siguiendo 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.

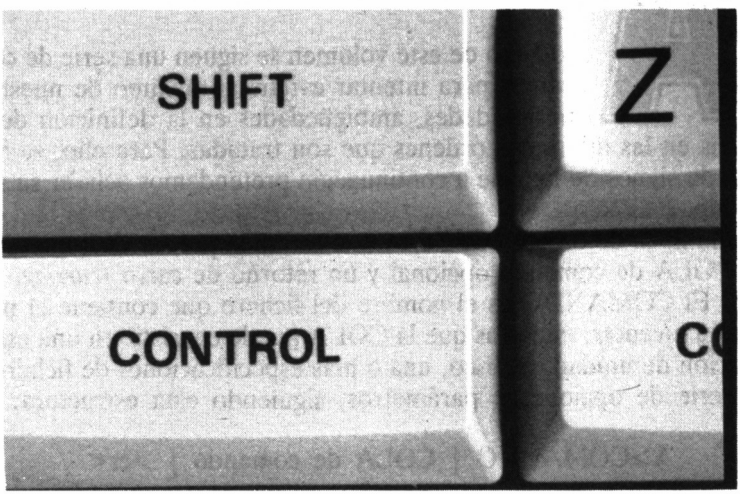
s señala una cadena de caracteres alfanuméricos.

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.



13.1.1. La tecla de CONTROL de los CPC tiene su homólogo en ALT de PCW.

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.

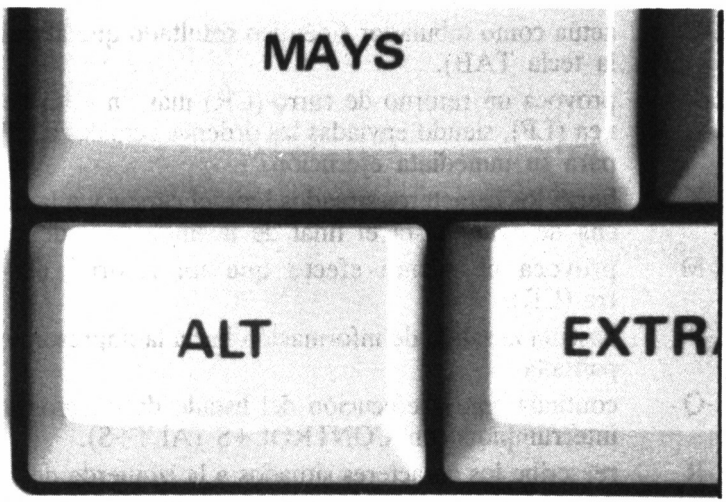
... indica que el elemento que los precede dentro de la sintaxis de una orden puede repetirse el número de veces requerido por el comando.

* 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 siste-



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

ma 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.

CTRL-H borra el carácter situado a la izquierda de la posición actual del cursor.

CTRL-I actúa como tabulador (idéntico resultado que al pulsar la tecla TAB).

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).

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.

DEVICE

omenzamos el tratamiento de los comandos con uno de los más conflictivos, dado que resulta harto complicado intentar combinar operadores lógicos con operadores físicos, más concretamente, dispositivos (en inglés, *device)*.

DEVICE asigna los dispositivos lógicos de CP/M Plus a los físicos (periféricos) conectados al ordenador, así como establece los parámetros de comunicación asociados. Por otra parte, también sirve para mostrar la asignación actual de los dispositivos lógicos siguientes:

CONIN: Entrada de consola (teclado).
CONOUT: Salida de consola (monitor).
AUXIN: Sección de entrada de un dispositivo auxiliar (por ejemplo, un interface de comunicaciones, que bien pudiera ser un MODEM).
AUXOUT: Sección de salida de un dispositivo auxiliar (igualmente, podría tratarse de un MODEM).
LST: La impresora.

Finalmente, hemos de añadir que los dispositivos físicos que pueden intervenir en las órdenes DEVICE dependen de la configuración hardware concreta del sistema.

OPERADORES PARA DEVICE

A continuación, presentamos la relación de operadores cuyo uso es posible con DEVICE:

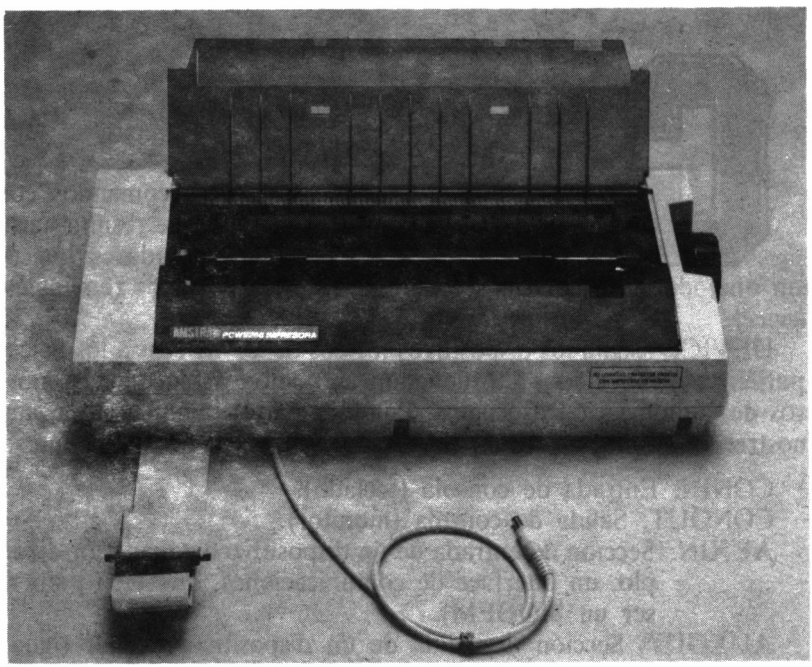
DEVICE NAMES

Proporciona la lista de los dispositivos físicos aceptables por el ordenador, junto con un resumen de sus características básicas (tres datos por dispositivo).

DEVICE VALUES

Presenta la asignación actual de los dispositivos lógicos.

DEVICE dispositivo-físico



13.2.1. DEVICE es un comando fundamental en la gestión de dispositivos por CP/M.

Da las características del dispositivo físico indicado.

DEVICE dispositivo-lógico

Actúa de forma similar al anterior, aunque respecto a los dispositivos lógicos.

DEVICE dispositivo-lógico-dispositivo-físico [opción [,opción]]

Asigna el dispositivo lógico al dispositivo físico.

DEVICE dispositivo-lógico-dispositivo-físico 1, dispositivo-físico 2 [,dispositivo físico N]

Realiza la asignación simultánea de dispositivos lógicos a varios dispositivos físicos.

DEVICE dispositivo-lógico-null

Desconecta el dispositivo lógico de todos los dispositivos físicos.

DEVICE

```
A>device names

Physical Devices:
I=Input,O=Output,S=Serial,X=Xon-Xoff
CRT     NONE  IO      LPT     NONE  O
```

13.2.2. Actuación de la opción DEVICE NAMES.

```
A>device values

Current Assignments:
CONIN:   = CRT
CONOUT:  = CRT
AUXIN:   = Null Device
AUXOUT:  = Null Device
LST:     = LPT
```

13.2.3. Actuación de la opción DEVICE VALUES.

Lista los dispositivos físicos aceptables por el sistema, junto con sus características, al igual que DEVICE NAMES, así como las asignaciones actuales de los dispositivos lógicos. A continuación, solicita al usuario nuevas asignaciones de dispositivos, que pueden tomar la forma de dispositivo lógico = dispositivo físico.

OTRAS OPCIONES DEVICE

Las opciones asociadas a la orden externa DEVICE sirven para modificar la interacción con algún dispositivo físico, generalmente un interface de comunicaciones serie o paralelo, como por ejemplo el interface RS232C/Centronics de AMSTRAD, ya sea para comunicaciones (MODEM) o cualquier otra función.

En este caso, las opciones deben ser escritas entre corchetes, separadas unas de otras por comas, debiendo dejar un espacio a continuación del nombre del dispositivo y el signo abrir corchetes.

Si se ha conectado un interface del tipo mencionado anteriormente, las listas de dispositivos físicos obtenidas con DEVICE incluirán líneas para SIO (serie, *Serial Input Output)* y CEN (Centronics). Además, en tal caso los dispositivos AUXIN y AUXOUT son asignados automáticamente a SIO cuando se carga el sistema operativo.

XON — activa el procedimiento de señalización por el cual el periférico avisa al ordenador que está dispuesto para recibir más datos.

```
A>device crt

Physical Device:   CRT
Baud Rate:         NONE
Characteristics:   INPUT
                   OUTPUT
                   PARALLEL
```

13.2.4. Actuación de la opción DEVICE CRT.

NOXON en oposición a la característica anterior, permite que el ordenador envíe datos al periférico, esté o no el mismo preparado para recibirlos; es decir, sin necesidad de recibir la señal de «dispuesto para recibir datos».

n establece la velocidad de transmisión de datos al periférico; donde «n» es el número de baudios. Los valores admitidos son: 50, 75, 110, 134,5, 150, 300, 600, 1.200, 1.800, 2.400, 3.600, 4.800, 7.200, 9.600 y 19.200. No obstante, tengamos en cuenta que tan sólo alguno de estos valores serán adecuados para cada dispositivo concreto.

MÁS FUNCIONES DE DEVICE

En otro orden de funciones de DEVICE, disponemos de una serie de operadores que nos son de gran utilidad a la hora de confeccionar las configuraciones para las distintas aplicaciones:

DEVICE CRT

Muestra los atributos del dispositivo físico CRT.

DEVICE CON

Muestra los atributos del dispositivo lógico CON.

DEVICE CONOUT=LPT,CRT

Asigna al dispositivo de salida por teclado (CONOUT) la impresora (LPT) y la pantalla (CRT).

DEVICE LST=SIO [XON,9600]

```
A>device lpt

Physical Device:   LPT
Baud Rate:         NONE
Characteristics:   OUTPUT
                   PARALLEL
```

13.2.5. Actuación de la opción DEVICE LPT.

Asigna al dispositivo registrador (LST) el de comunicación opcional (SIO), utilizando el registro XON/XOFF y pasa la transmisión de datos en proporción a 9.600 baudios.

DEVICE LST=NULL

Desconecta el dispositivo de registro de salida lógica.

DEVICE SIO [XON/9600]

Pasa el registro XON/XOFF por el dispositivo físico opcional SIO y fija la velocidad a 9.600 baudios.

DEVICE CONSOLE [PAGE]

Muestra la progresión de la anchura en columnas y longitud en líneas.

DEVICE CONSOLE [COLUMNS=40 LINES=16]

Pasa a ajustar la pantalla, por ejemplo, a 40 columnas y 16 líneas.

DISPOSITIVOS FÍSICOS CON DEVICE

En los ejemplos anteriores, el dispositivo lógico va seguido de una coma y, como ya hemos dicho anteriormente, los nombres de los dispositivos físicos dependen del hardware, y pueden variar entre los distintos ordenadores.

```
A>device auxout=crt

Physical Devices:
I=Input,O=Output,S=Serial,X=Xon-Xof
CRT     NONE  IO      LPT     NONE  O

Current Assignments:
CONIN:   = CRT
CONOUT:  = CRT
AUXIN:   = Null Device
AUXOUT:  = CRT
LST:     = LPT
```

13.2.6. Asignación de dispositivo AUXOUT a CRT.

CP/M Plus soporta cinco nombres de dispositivos lógicos:

1. CONIN.
2. CONOUT.
3. AUXIN.
4. AUXOUT.
5. LST.

Los nombres de estos dispositivos lógicos son también conocidos por las siguientes denominaciones de dispositivos lógicos adicionales:

```
A>device auxout=crt,lpt

Physical Devices:
I=Input,O=Output,S=Serial,X=Xon-Xoff
CRT     NONE  IO      LPT     NONE  O

Current Assignments:
CONIN:   = CRT
CONOUT:  = CRT
AUXIN:   = Null Device
AUXOUT:  = CRT     LPT
LST:     = LPT
```

13.2.7. Asignación múltiple de dispositivo.

```
A>device auxout=null

Physical Devices:
I=Input,O=Output,S=Serial,X=Xon-Xoff
CRT     NONE  IO      LPT     NONE  O

Current Assignments:
CONIN:   = CRT
CONOUT:  = CRT
AUXIN:   = Null Device
AUXOUT:  = Null Device
LST:     = LPT
```

13.2.8. Reasignación a NULL de dispositivo.

Nombre dispositivo	Dispositivo lógico
CON	CONIN Y CONOUT
CONSOLE	CONIN Y CONOUT
KEYBOARD	CONIN
AUX	AUXIN Y AUXOUT
AUXILIARY	AUXIN Y AUXOUT
PRINTER	LST

CONCLUSIÓN

Para finalizar, diremos que DEVICE es un comando o utilidad de refuerzo, diseñado para gestionar y operar de dispositivos lógicos a físicos. Puede trabajar con ambos sistemas, así como controlar la tasa de baudios a la que debe transferirse la información.

```
A>device

Physical Devices:
I=Input,O=Output,S=Serial,X=Xon-Xoff
CRT     NONE  IO      LPT     NONE  O

Current Assignments:
CONIN:   = CRT
CONOUT:  = CRT
AUXIN:   = Null Device
AUXOUT:  = Null Device
LST:     = LPT
Enter new assignment or hit RETURN
auxout=crt,lpt

Physical Devices:
I=Input,O=Output,S=Serial,X=Xon-Xoff
CRT     NONE  IO      LPT     NONE  O

Current Assignments:
CONIN:   = CRT
CONOUT:  = CRT
AUXIN:   = Null Device
AUXOUT:  = CRT     LPT
LST:     = LPT
```

13.2.9. Actuación del comando DEVICE.

DUMP

o sabemos realmente por qué, pero siempre que hablamos del sistema hexadecimal, código ASCII y demás «familia» nos encontramos un poco perdidos sólo de pensar en lo que nos espera. El objeto de DUMP no es otro que hacer, precisamente gracias al formato hexadecimal y al código ASCII, más llevadera la siempre ardua tarea de examinar un fichero.

UTILIZACIÓN DE DUMP

Así, pues, este comando nos muestra en la pantalla el contenido de un fichero, en formato hexadecimal y ASCII, en líneas de 16 bytes, con la representación ASCII de los caracteres, de ser posible a la derecha de la pantalla. Por otra parte, la simplicidad de su utilización es apabullante:

DUMP nombre-fichero

No obstante, esta orden no es posible emplearla en forma de programa; es decir, introduciendo previamente

A>DUMP

y esperando que a continuación el sistema solicite mediante algún mensaje el nombre de fichero a tratar, dándonos el tiempo necesario para el cambio de disco si fuera necesario. Por el contrario, de hacerlo así se producirá el siguiente efecto:

A>DUMP

CP/M 3 DUMP - Version 3.0
ERROR: File Not Found

Por tanto, si deseamos examinar un fichero que no se encuentra en la misma cara que DUMP.COM deberemos recurrir a indicar que éste se halla en otra unidad de disco. Por ejemplo:

A>DUMP B:nombre-fichero

```
A>dump ddhp7470.prl

CP/M 3 DUMP - Version 3.0
0000: 00 55 24 00 00 00 00 00 00 00 00 00 00 00 00 00  .U$.............
0010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0100: EB 5E 23 56 23 EB 22 16 01 EB 5E 23 56 23 44 4D  .^RVR."...^RVRDM
0110: 2A 16 01 C3 18 01 00 00 22 D1 20 EB 22 D3 20 3E  *.......". .". >
0120: 03 21 D5 20 CD D0 1F 2A D1 20 7E 32 DB 20 21 04  .!. ...*. ~2. !.
0130: 00 EB 2A D1 20 19 22 34 22 21 00 00 EB 2A 34 22  ..*. ."4"!...*4"
0140: 73 23 72 3A DB 20 3D D6 01 9F 2F 32 36 22 B7 C2  sRr:. =.../26"..
0150: 87 03 21 04 00 EB 2A D1 20 19 22 34 22 21 06 00  ..!...*. ."4"!..
0160: EB 2A 34 22 73 23 72 21 08 00 EB 2A D1 20 19 22  .*4"sRr!...*. ."
0170: 37 22 21 2D 00 EB 2A 37 22 73 23 72 21 01 00 22  7"!-..*7"sRr!.."
0180: DC 20 2A DC 20 29 22 34 22 2B 2B EB 2A D7 20 19  . *. )"4"++.*. .
0190: 22 37 22 2A 34 22 11 C0 21 19 7E 23 66 6F EB 2A  "7"*4"..!.~Rfo.*
01A0: 37 22 73 23 72 2A DC 20 23 3E 2D 95 3E 00 9C F2  7"sRr*. R>-.>...
01B0: 7F 01 21 01 00 22 DC 20 2A DC 20 29 22 34 22 2B  ..!..". *. )"4"+
01C0: 2B EB 2A D9 20 19 22 37 22 2A 34 22 11 1A 22 19  +.*. ."7"*4"..".
Press RETURN to continue
```

13.3.1. Volcado de un fichero en la unidad de disco implícita.

```
A>dump b:keys.wp

CP/M 3 DUMP - Version 3.0
0000: 31 34 20 4E 20 53 20 20 22 5E 45 22 20 20 20 20  14 N S  "^E"
0010: 20 5E 45 0D 0A 31 34 20 41 20 53 41 20 22 5E 27   ^E..14 A SA "^'
0020: 23 39 45 27 22 20 5E 51 45 0D 0A 20 36 20 4E 20  R9E'" ^QE.. 6 N
0030: 53 20 20 22 5E 44 22 20 20 20 20 20 5E 44 0D 0A  S  "^D"      ^D..
0040: 37 39 20 4E 20 53 20 20 22 5E 58 22 20 20 20 20  79 N S  "^X"
0050: 20 5E 58 0D 0A 37 39 20 41 20 53 41 20 22 5E 27   ^X..79 A SA "^'
0060: 23 39 38 27 22 20 5E 51 58 0D 0A 31 35 20 4E 20  R98'" ^QX..15 N
0070: 53 20 20 22 5E 53 22 20 20 20 20 20 5E 53 0D 0A  S  "^S"      ^S..
0080: 20 35 20 4E 20 20 20 20 22 5E 44 22 20 20 20 20   5 N    "^D"
0090: 20 5E 44 0D 0A 20 35 20 41 20 20 20 20 22 5E 53   ^D.. 5 A    "^S
00A0: 22 20 20 20 20 20 5E 53 0D 0A 20 35 20 20 20 53  "     ^S.. 5   S
00B0: 20 20 22 5E 46 22 20 20 20 20 20 5E 46 0D 0A 20    "^F"     ^F..
00C0: 35 20 20 20 53 41 20 22 5E 41 22 20 20 20 20 20  5   SA "^A"
00D0: 5E 41 0D 0A 31 33 20 4E 20 20 20 20 22 5E 27 23  ^A..13 N    "^'R
00E0: 39 43 27 22 20 5E 51 44 0D 0A 31 33 20 20 20 53  9C'" ^QD..13   S
00F0: 20 20 22 5E 27 23 39 43 27 22 20 5E 51 44 0D 0A    "^'R9C'" ^QD..
0100: 31 33 20 20 20 53 41 20 22 5E 27 23 39 44 27 22  13   SA "^'R9D'"
0110: 20 5E 51 53 0D 0A 31 32 20 4E 20 20 20 20 22 5E   ^QS..12 N    "^
0120: 43 22 20 20 20 20 20 5E 43 0D 0A 31 32 20 41 20  C"      ^C..12 A
0130: 20 20 20 22 5E 52 22 20 20 20 20 20 5E 52 0D 0A     "^R"      ^R..
0140: 31 32 20 20 20 53 20 20 22 5E 27 23 39 30 27 22  12   S  "^'R90'"
0150: 20 5E 51 43 0D 0A 31 32 20 20 20 53 41 20 22 5E   ^QC..12   SA "^
0160: 27 23 39 31 27 22 20 5E 51 52 0D 0A 32 30 20 4E  'R91'" ^QR..20 N
0170: 20 20 20 20 22 5E 27 23 39 32 27 22 20 5E 51 46      "^'R92'" ^QF
0180: 0D 0A 32 30 20 20 20 53 20 20 22 5E 27 23 39 33  ..20   S  "^'R93
0190: 27 22 20 5E 51 41 0D 0A 31 30 20 4E 20 20 20 20  '" ^QA..10 N
01A0: 22 5E 27 23 39 34 27 22 20 5E 4B 42 0D 0A 31 30  "^'R94'" ^KB..10
01B0: 20 20 20 53 20 20 22 5E 27 23 39 35 27 22 20 5E     S  "^'R95'" ^
01C0: 4B 4B 0D 0A 31 31 20 4E 20 53 20 20 22 5E 27 23  KK..11 N S  "^'R
01D0: 39 36 27 22 20 5E 4B 43 0D 0A 20 33 20 4E 20 53  96'" ^KC.. 3 N S
01E0: 20 20 22 5E 27 23 39 37 27 22 20 5E 4B 56 0D 0A    "^'R97'" ^KV..
01F0: 20 31 20 4E 20 53 20 20 22 5E 42 22 20 20 20 20   1 N S  "^B"
0200: 20 5E 42 0D 0A 32 33 20 4E 20 53 20 20 22 5E 56   ^B..23 N S  "^V
0210: 22 20 20 20 20 20 5E 56 0D 0A 31 36 20 4E 20 53  "      ^V..16 N S
0220: 20 20 22 5E 47 22 20 20 20 20 20 5E 47 0D 0A 31    "^G"     ^G..1
0230: 36 20 41 20 53 41 20 22 5E 27 23 39 41 27 22 20  6 A SA "^'R9A'"
0240: 5E 51 59 0D 0A 37 32 20 41 20 53 41 20 22 5E 27  ^QY..72 A SA "^'
0250: 23 39 42 27 22 20 5E 51 64 65 6C 0D 0A 20 38 20  R9B'" ^Qdel.. 8
0260: 4E 20 53 20 20 22 5E 27 23 31 42 27 22 20 65 73  N S  "^'R1B'" es
0270: 63 0D 0A 36 36 20 4E 20 53 20 20 22 5E 55 22 20  c..66 N S  "^U"
0280: 20 20 20 20 5E 55 0D 0A 0D 0A 45 20 23 39 30 20      ^U....E R90
0290: 22 5E 51 43 22 0D 0A 45 20 23 39 31 20 22 5E 51  "^QC"..E R91 "^Q
02A0: 52 22 0D 0A 45 20 23 39 32 20 22 5E 51 46 22 0D  R"..E R92 "^QF".
02B0: 0A 45 20 23 39 33 20 22 5E 51 41 22 0D 0A 45 20  .E R93 "^QA"..E
02C0: 23 39 34 20 22 5E 4B 42 22 0D 0A 45 20 23 39 35  R94 "^KB"..E R95
02D0: 20 22 5E 4B 4B 22 0D 0A 45 20 23 39 36 20 22 5E   "^KK"..E R96 "^
02E0: 4B 43 22 0D 0A 45 20 23 39 37 20 22 5E 4B 56 22  KC"..E R97 "^KV"
02F0: 0D 0A 45 20 23 39 38 20 22 5E 51 58 22 0D 0A 45  ..E R98 "^QX"..E
0300: 20 23 39 41 20 22 5E 51 59 22 0D 0A 45 20 23 39   R9A "^QY"..E R9
0310: 42 20 22 5E 51 5E 27 23 37 46 27 22 0D 0A 45 20  B "^Q^'R7F'"..E
0320: 23 39 43 20 22 5E 51 44 22 0D 0A 45 20 23 39 44  R9C "^QD"..E R9D
0330: 20 22 5E 51 53 22 0D 0A 45 20 23 39 45 20 22 5E   "^QS"..E R9E "^
0340: 51 45 22 0D 0A 1A 4B 42 22 0D 0A 45 20 23 39 35  QE"...KB"..E R95
0350: 20 22 5E 4B 4B 22 0D 0A 45 20 23 39 36 20 22 5E   "^KK"..E R96 "^
0360: 4B 43 22 0D 0A 45 20 23 39 37 20 22 5E 4B 56 22  KC"..E R97 "^KV"
0370: 0D 0A 45 20 23 39 38 20 22 5E 51 58 22 0D 0A 45  ..E R98 "^QX"..E
```

13.3.2. Volcado de un fichero en la unidad de disco B.

APLICACIONES DE DUMP

Este comando se aplica principalmente al análisis de contenidos de ficheros que no pueden ser representados por sus valores ASCII o en busca de los caracteres de control que estén dentro de los textos de ficheros ASCII, de ahí su diferencia con la orden TYPE.

En este mismo sentido, es de especial interés para el volcado de información por la impresora que precisa una absoluta fidelidad, ya que ciertos datos, tales como caracteres de control, suelen ser interpretados de forma errónea por este tipo de periféricos, de forma que se deforma o amputa la información original.

Análogamente, puede ser de extraordinaria utilidad para el análisis detenido, generalmente por impresora, de programas en código máquina, que por su corta longitud no precisan la concurrencia de un ensamblador.

GENCOM

La utilidad del comando que a continuación tratamos se encuentra en base a nuestro conocimiento de las extensiones del sistema residentes (RSX, *Resident System Extensions)*. Estas son módulos adicionales al sistema operativo, temporalmente residentes en la memoria, que amplían las funciones de CP/M Plus.

GENCOM incorpora una o varias «extensiones del sistema residente», hasta un máximo de 15, a un fichero de orden (tipo .COM), permitiendo así que el programa contenido en el fichero utilice las RSX en cuestión.

OPCIONES

Las opciones para la orden GENCOM son las siguientes:

LOADER:

Conecta una bandera para el mantenimiento en activo del programa cargador.

NULL:

Indica que sólo los ficheros RSX están especificados. GENCOM genera un fichero de simulación .COM para los ficheros RSX. La denominación para dicho fichero se toma del nombre del primer fichero RSX.

SCB-(offset-value):

Prepara el System Control Block desde el programa, utilizando los valores hexadecimales especificados por: offset, value (desplazamiento, valor).

EJEMPLOS

Como ejemplo de lo anteriormente citado, y para que el lector tenga una idea más acertada sobre el comando que nos ocupa, he aquí cuatro representaciones de estas opciones de GENCOM:

A>GENCOM PRUEBA

GENCOM elimina la cabecera del fichero PRUEBA.COM y borra todos los RSX asociados, para volverlo a su formato COM original.

A>GENCOM PROG1 PROG2 [NULL]

Genera un fichero PROG1.COM con los RSX PROG1 y PROG2.

A>GENCOM PRUEBA PROG1 PROG2

Genera un nuevo fichero PRUEBA.COM con los RSX asociados PROG1 y PROG2.

A>GENCOM PRUEBA PROG1 PROG2

GENCOM busca ahora en el fichero PRUEBA.COM, que ya ha sido sometido anteriormente a su acción, si los ficheros PROG1.RSX y PROG2.RSX ya están asociados al módulo. Si al menos uno de ellos ya se halla asociado, GENCOM lo sustituye por el nuevo módulo RSX. De no ser así, GENCOM añade los ficheros RSX especificados al fichero .COM.

GENCOM EN BDOS

Analizando más si cabe el comando GENCOM y refiriéndonos al BDOS, ampliamente estudiado en el segundo volumen de esta colec-

ción, tenemos que saber que éste incorpora una página denominada GENCOM, la cual ilustramos a continuación con todo tipo de detalle.

CABECERA GENCOM — Función de cargador

Despl.	Tamaño	Descripción
00H	01H	'C9H' Identifica GENCOM.
01H	Palabra	Longitud de programa COM desde comienzo hasta 0100H.
03H	0Dh	Rutina inicializadora en cabecera. RET termina.
(Primer RSX)		
10H	Palabra	Desplazamiento al comienzo del módulo RSX 0000 = fin de RSX.
12H	Palabra	Longitud del módulo RSX (Excluido mapa BIT).
14H	Byte	Fija a OFFH si RSX no está cargado en los bancos del sistema.
(Segundo RSX)		
20H	Palabra	Desplazamiento al comienzo del módulo RSX 0000 = fin de RSX.
22H	Palabra	Longitud del módulo RSX (Excluido mapa BIT).
24H	Byte	Fija a 0FFH si RSX no está cargado en los bancos del sistema.
etc., hasta el desplazamiento RSX = 0000.		
100H	01H	'C9H' identifica programa NULL COM y fija el desplazamiento 17H bit 1 del System Control Block.

GET

En este capítulo estudiaremos un comando de gran ayuda a la hora de presentar por pantalla ciertos ficheros: GET.

El efecto de éste es obligar al sistema a que en el futuro lea en el fichero especificado todas las entradas que normalmente captaría por la consola, es decir, por el teclado. Tales entradas pueden ser órdenes de CP/M, datos requeridos por los programas, o ambas cosas. Por otra parte, la entrada por consola se lee en el fichero especificado hasta que se agote éste o termine el programa.

Si no se indica otra cosa, mediante una de las opciones de GET, la entrada es reproducida en la pantalla del monitor. Las opciones se escriben entre corchetes, separadas unas de otras por comas o por espacios sencillos.

VARIANTES DE GET

A continuación, procedemos a estudiar las variantes del comando GET.

GET CONSOLE INPUT FROM FILE fichero: Ordena al sistema que lea en el fichero especificado las entradas al programa que se va a ejecutar a continuación, hasta que éste se agote o termine el programa. La orden que pone en marcha el programa tiene que

ser la siguiente que se dé por el teclado. Todas las entradas al programa se reproducen en la pantalla.

GET CONSOLE INPUT FROM FILE fichero [SYSTEM]: Ordena al sistema que lea en el fichero especificado todas las entradas que normalmente captaría por el teclado, hasta que el fichero se agote o termine el programa. Ese fichero puede incluir órdenes que ejecuten programas. Todas las entradas leídas en el fichero se reproducen en la pantalla.

GET CONSOLE INPUT FROM FILE fichero [NO ECHO]: Ordena al sistema que lea en el fichero especificado todas las entradas que normalmente captaría por el teclado, hasta que el fichero se agote o termine el programa. Ese fichero puede incluir órdenes que ejecuten programas. Todas las entradas leídas en el fichero no se reproducen en la pantalla. El efecto contrario se consigue con la opción [ECHO].

GET CONSOLE INPUT FROM CONSOLE: Ordena al sistema que vuelva a captar las entradas por el teclado. Cancela el efecto de una orden GET FILE anterior.

EJEMPLOS

Veamos ahora algunos ejemplos para la mejor comprensión del comando.

A>GET CONSOLE INPUT FROM FILE ENTRCONS.DAT [NO ECHO]
A>PROG

Ordena al sistema que lea en el fichero ENTRCONS.DAT de la unidad implícita todas las entradas al programa PROG que de otra forma captaría por el teclado.

```
A>
A>get
CP/M 3 GET Version 3.0
Get console input from a file
Enter file: ddhp7470.prl
Getting console input from file: DDHP7470.PRL
A>
```

13.5.1. En el ejemplo, suponemos que el fichero de datos de entrada será "ddhp7470.prl".

Las entradas no son reproducidas por pantalla.

A>GET CONSOLE INPUT FROM FILE B:ENTRCONS.DAT [SYSTEM,NO ECHO]

Ordena al sistema que lea en el fichero ENTRCONS.DAT que hay en el disco de la unidad B, todas las entradas que normalmente captaría por el teclado, hasta que el fichero se agote. Tales entradas no serán reproducidas en la pantalla.

OTRAS CONSIDERACIONES

En cierto modo, este comando se puede considerar similar a SUBMIT, del cual hablaremos más adelante, viniendo marcada la diferencia entre ambos, porque GET no puede controlar los parámetros del fichero utilizado en último lugar.

Si el fichero se agota antes de terminar de introducir el programa, éste busca subsecuencias introducidas desde el teclado. Si por el contrario, el programa acabó antes, estando introducido todo el fichero, el sistema regresa a la consola desde la introducción última que se realizó desde la misma.

```
A>get console input from console
GET  from file: A:DDHP7470.PRL stopped
Getting console input from console
A>
```

13.5.2. Muestra de la vuelta de entrada de datos a la consola.

```
A>get console input from file ddhp7470.prl ¡no echo¿
Getting console input from file: DDHP7470.PRL
A>
```

13.5.3. Entre las opciones de GET se encuentra [NO ECHO].

GSX

SX es un sistema de gestión de gráficos que permite a los programas de CP/M dar como salida gráficos además de textos. Así, pues, su uso se concentra en la confección de aplicaciones destinadas a la generación de diagramas de todo tipo, o bien a la impresión de textos de magnitud superior a la normal y especial calidad, tales como pueden ser títulos o encabezamientos. Ello es debido a que, como podemos suponer, su salida se puede dirigir tanto a la pantalla, como a un medio más perdurable: la impresora o un plotter.

Antes de continuar es necesario hacer una puntualización de gran importancia. GSX constituye un apoyo a la confección de gráficos, si bien no tiene capacidad por sí mismo para generarlos. Esto quiere decir que supone la base necesaria para que un programa de dibujo pueda llevar a cabo correcta y cómodamente su trabajo, quedando estandarizado el acceso a pantalla y demás periféricos, y posibilitando que dichas aplicaciones concretas sean ejecutables en otros ordenadores que soporten el sistema GSX de CP/M, requiriendo un mínimo esfuerzo de adaptación.

Así pues, debe quedar claro que GSX lleva a cabo el trabajo típico del sistema operativo: facilitar la gestión de aplicaciones y compatibilizar al máximo las diferentes máquinas, concretamente en lo concerniente al área gráfica.

Otra de las ventajas de GSX es su independencia de los dispositivos de los que hace uso. Cualquier aplicación que haga uso de él puede desentenderse de los típicos problemas anejos al trazado de líneas y demás tareas gráficas de los diferentes monitores, impresoras y plotters que se pueden utilizar.

No obstante, es claro que el sistema GSX no es capaz de controlar cualquiera de los múltiples periféricos presentes en el mercado, ni tampoco puede utilizarse con cualquier programa de gráficos. Tanto en un caso como en otro, impresoras, plotters o programas, tienen que haber sido concebidos para su funcionamiento bajo CP/M y más concretamente, sometidos al gobierno de GSX.

Por otra parte, hemos de tener en cuenta que se pueden dar casos de compatibilidades parciales; por ejemplo, que un determinado periférico no sea capaz de realizar una determinada función de las soportadas por GSX. De ser así, la aplicación se ejecutará normalmente, aunque llegada la hora de realizar el trabajo concreto para el cual no está capacitado el periférico, éste no será tenido en cuenta.

A pesar de estas objeciones, GSX es un sistema ampliamente estandarizado y difundido entre gran cantidad de fabricantes de hardware y software, lo cual implica que existe una extensa biblioteca de aplicaciones que lo soportan, así como un gran número de periféricos capaces de hacer de él una herramienta extremadamente eficaz para el diseño gráfico. Tengamos en cuenta, que en impresoras como la que incorpora el PCW 8256, o en general cualquier periférico de rela-

```
A>
A>type ASSIGN.SYS
21 a:ddfxlr7  ; Epson 7 bit printer
11 a:ddhp7470 ; Pen plotter
01 a:ddmode2  ; Screen in mode 2
02 a:ddmode1  ; Screen in mode 1
03 a:ddmode0  ; Screen in mode 0

A>
```

13.6.1. TYPE del fichero ASSIGN.SYS.

tiva calidad, basta con desviar la salida del monitor al elemento impresor, para obtener una copia sobre papel del gráfico de gran fidelidad.

MÁS SOBRE GSX

Completando lo dicho hasta el momento, GSX es un estándar gráfico, a modo de interface entre el programa y el hardware específico que se está utilizando en cada aplicación. Este, en resumen, intenta hacer con los gráficos la misma tarea de adaptación que CP/M lleva a cabo con el disco. Podríamos decir incluso que GSX se concibe para la representación gráfica de ideas, pero sin olvidar que por sí mismo no constituye más que el software adecuado, siendo absolutamente necesario un hardware que posibilite una perfecta identidad entre un elemento y otro.

Así, como ya hemos dicho, el cambio de una salida desde un periférico a otro (ambos terminales gráficos) como pueden ser de una impresora a un plotter, se reduce a cambiar, por ejemplo, un número en una rutina de software, pudiéndose representar así el mismo dibujo en cualquier dispositivo que haya sido instalado por GSX. Haciendo un poco de historia, este sistema se trata de una versión del

```
A>TYPE DRIVERS.GSX
The file ASSIGN.SYS contains a list of the device drivers available to
applications programs during a particular session. The program will call up
the required driver by its number ( 01 - 21 ), and the file ASSIGN.SYS gives
the drive and file name. The default drive can be indicated by using @: rather
than a: or b: You need only include the drivers you require, but there is a
maximum of five entries in the file, in descending order of size; with the
default screen device as number 01, the default printer (if required) as
number 11 and the default plotter (if required) as number 21.

Device drivers provided include:

DDFXLR7    Epson and Epson-compatible printers. Side 3.
DD-DMP1    Amstrad DMP1 printer. Side 2.
DDSHINWA   Printers using Shinwa mechanism. Side 2.
DDHP7470   Hewlett Packard 7470 and compatible pen plotters. Side 2.
DDMODE0    Screen in mode 0. Side 3.
DDMODE1    Screen in mode 1. Side 3.
DDMODE2    Screen in mode 2. Side 3.


A>
```

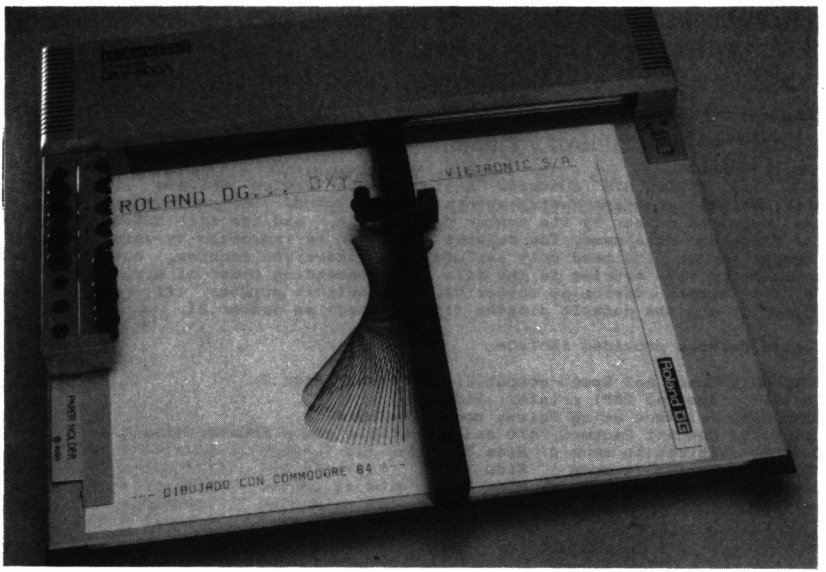
13.6.2. TYPE del fichero DRIVERS.GSX.

estándar ANSK GKS, el cual gestiona el manejo de gráficos controlado por software de alto nivel.

Aunque, lógicamente, la diferencia de calidad entre los mismos dibujos obtenidos con varios periféricos, como pueden ser una impresora de matriz de puntos de escasa calidad o un plotter de precisión, depende estrictamente de éstos, GSX está diseñado para poder extraer el máximo provecho de la capacidad generadora de gráficos del ordenador. Así, inicialmente se pensó para su utilización con periféricos gráficos de alta calidad y, por tanto, coste superior, con gran definición, tales como pantallas de alta resolución o plotters de precisión.

No obstante, el sistema en general no debe ser juzgado como caro por este hecho, dada la gran capacidad de adaptación de las aplicaciones diseñadas para GSX. Este índice de adaptación llega incluso a otras máquinas a las que se presupone con un hardware más caro que los equipos AMSTRAD, de gran consumo.

Obviamente, lo que no se puede esperar son milagros; los programas que están escritos usando GSX se encuentran mucho más preparados para ejecutarse en otras máquinas que ya llevan incorporado el sistema. Está todavía por verse la pantalla de baja resolución, que por



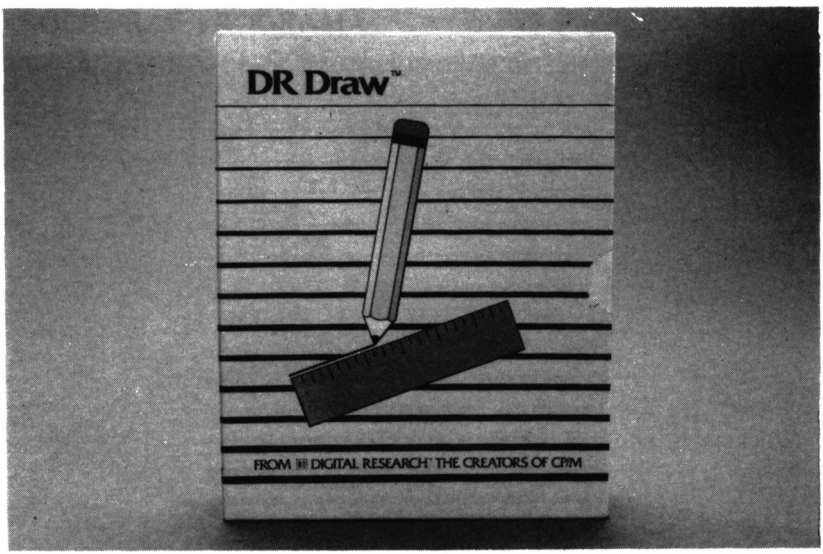
13.6.3. GSX puede funcionar con sofisticados dispositivos gráficos, tales como los plotters.

la magia de GSX dibuje una Giocconda capaz de engañar al propio Leonardo.

Por otra parte, se hace evidente que aplicaciones diseñadas específicamente para AMSTRAD CPC/PCW no precisan hacer uso de GSX, a menos que necesiten la concurrencia de las herramientas más perfeccionadas y la posibilidad de manejo de multitud de periféricos. Sin embargo, cualquier paquete gráfico concebido para su ejecución bajo CP/M en general, se verá en la obligación de recurrir inevitablemente al sistema.

Pese a todas las excelencias que hemos cantado de GSX, hemos de tener en cuenta que, como casi todo, tiene ciertos inconvenientes, entre los que se cuentan su lentitud de proceso o su incapacidad para la generación de gráficos tridimensionales; por otra parte, en ningún momento se debe confundir con un generador de gráficos para juegos, a no ser que a éstos no vaya a dotárseles de movimiento. No obstante, cumple a las mil maravillas su función en el diseño de planos de diversa índole, organigramas, circuitos y muchísimas otras aplicaciones.

Hemos de tener en cuenta que el programador en GSX dispone de muchas ventajas, entre las que se encuentran:



13.6.4. Entre las múltiples aplicaciones disponibles para GSX se encuentra DR Draw.

— Disponer de las sofisticadas herramientas para la creación de gráficos que aporta el sistema.

— Poder adaptar su aplicación al uso con un gran número de periféricos gráficos.

— Poder implantar su aplicación, con un mínimo esfuerzo de adaptación, a cualquier otra máquina que soporte CP/M.

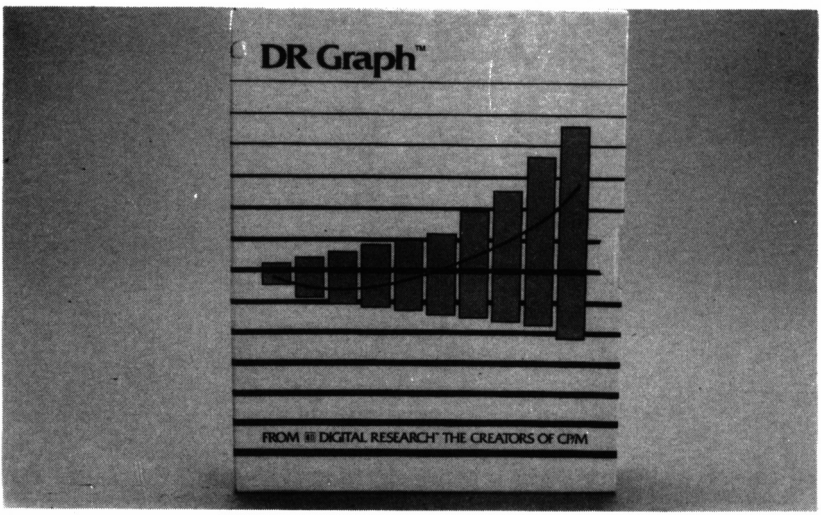
Por otro lado, las ventajas de disponer de una máquina que incorpore GSX son evidentes:

— Posibilidad de confeccionar gráficos.

— Incremento notable en la biblioteca de programas, al poder disponer de gran número de aplicaciones inicialmente diseñadas para otros ordenadores que soporten CP/M.

EJECUCIÓN DE PROGRAMAS GSX

Para crear un disco destinado a la ejecución de programas GSX deberemos copiar los ficheros GSX.SYS y ASSIGN.SYS, así como los controladores de dispositivos requeridos, junto con el propio programa de aplicación. El fichero ASSIGN.SYS contiene una lista de hasta tres controladores de dispositivo, en orden inverso a su tamaño.



13.6.5. DR Graph es un programa para el diseño gráfico, soportado sobre GSX, de excelentes resultados.

```
21 a:ddfx1r7      ; Epson 7 bit printer Impresora EPSON de 7 bit
11 a:ddhp7470     ; Pen plotter         Plotter
01 a:ddmode 2     ; Screen in mode 2    Pantalla en modo 2
02 a:ddmode 1     ; Screen in mode 1    Pantalla en modo 1
03 a:ddmode 0     ; Screen in mode 0    Pantalla en modo 0
```

Los números (01 a 21) informan a GSX del tipo de dispositivo de que se trata: impresora, plotter o pantalla. En todo momento hay más de un controlador de dispositivo cargado en la memoria; por ello, GSX necesita conocer el tamaño del mayor, con el fin de reservarle el espacio necesario.

Con los discos se proporcionan diversos controladores para los tres modos de pantalla del 6128 y para las impresoras y plotters estándar. El fichero DRIVERS.GSX contiene una lista de los disponibles, que podemos visualizar efectuando un TYPE del fichero DRIVERS.GSX, lo cual nos permitirá seleccionar los controladores de dispositivo requeridos.

Normalmente, los programas de aplicación incluirán un cargador GSX, de modo que para ejecutar el programa bastará con escribir el nombre del mismo en respuesta al *prompt* (A>). No obstante, si la aplicación no tiene instalado el cargador de GSX, deberemos copiar el fichero GENGRAF.COM y escribir:

```
GENGRAF PROGRAMA
```

donde PROGRAMA.COM es el nombre de la aplicación instalada. Ahora ya podemos borrar GENGRAF.COM, puesto que PROGRAMA.COM ya contiene el cargador de GSX.

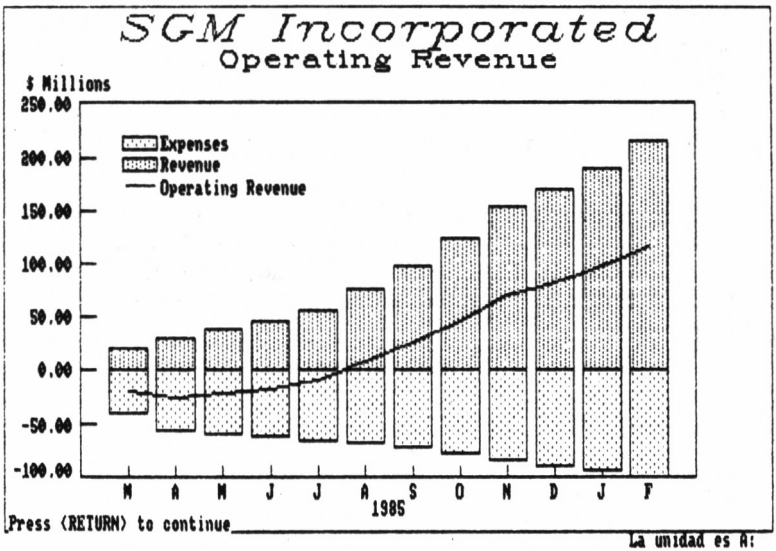
El Sistema GSX En El CPC Y PCW

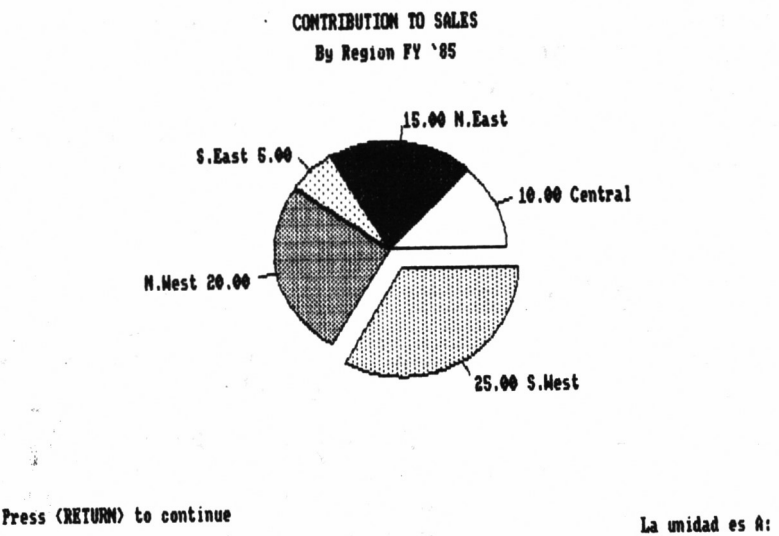
La mayor parte de los ficheros para ejecutar en el CPC y PCW programas basados en GSX están grabados en los discos del sistema.

GSX.SYS: Fichero que contiene el propio GSX.

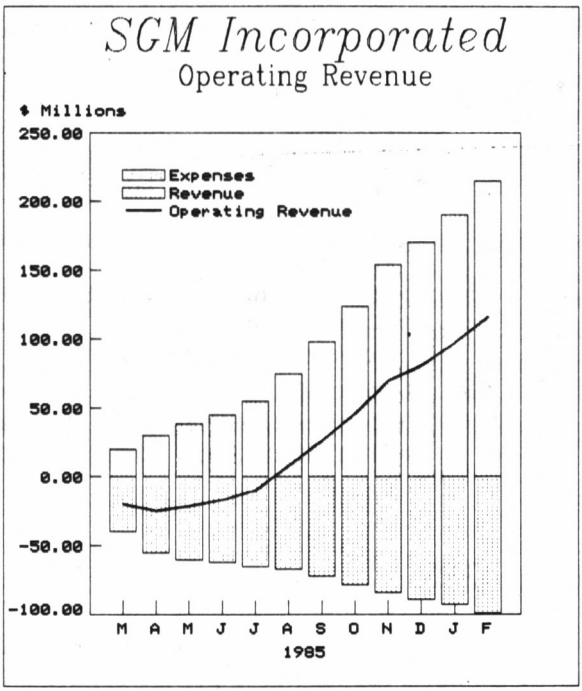
ASSIGN.SYS: Fichero en el que se anotan los dispositivos que están conectados y los números de dispositivo lógico por los que el programa los identifica.

GENGRAF.COM: Programa de ayuda que se utiliza para asegurar que GSX se cargue automáticamente junto con el programa que vaya a utilizarlo.



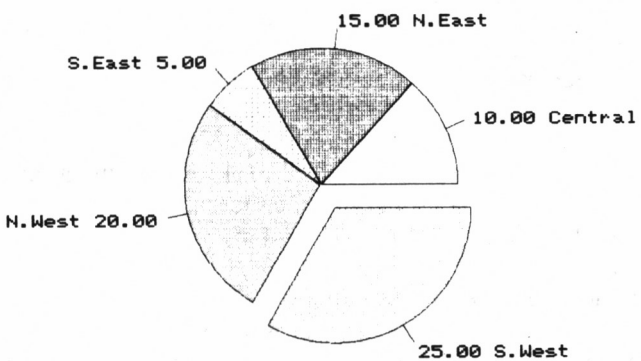


13.6.6. Diferentes muestras de salidas de gráficos obtenidas con DR Graph.



CONTRIBUTION TO SALES

By Region FY '85



DDSCREEN.PRL: Fichero controlador de dispositivos, en el cual se guarda cierta información relativa a las características del monitor del PCW 8256.

DDFXLR8.PRL: Fichero controlador de dispositivo, en el cual se guarda cierta información relativa a las características de la impresora del PCW 8256.

DDFXHR8.PRL: Segundo controlador de dispositivo, gracias al cual la impresora del PCW 8256 puede funcionar en alta resolución.

DDHP7470PRL: Controlador de dispositivo que contiene la información necesaria para trabajar con un plotter de plumillas compatible con HP.

PROGRAMAS GSX

Existen varios paquetes de aplicación comercial que producen gráficos a partir de los datos suministrados por los usuarios. Tales programas serán ejecutados en el PCW y CPC si han sido diseñados para funcionar bajo el control de CP/M 2.2 o CP/M Plus (CP/M) con GSX y si están grabados en disco de tres pulgadas de formato Amstrad.

También podemos escribir nuestros programas para dibujar gráficos, de forma similar a como escribiríamos cualquier otro programa para CP/M. Sin embargo, GSX tiene su propio vocabulario de instrucciones.

El paquete de aplicación que adquiramos estará adaptado a PCW o CPC o de lo contrario, vendrá acompañado de un programa de instalación, a través del cual podrá proporcionar al programa principal los parámetros que éste necesite para funcionar correctamente en estas dos máquinas.

La instalación de un programa para GSX suele ser más sencilla que la de cualquier otro programa de CP/M, puesto que gran parte de la información necesaria está grabada en los ficheros controladores de dispositivos. Los datos que tenemos que escribir son, por consiguiente, muchos menos.

Apelando a un mejor rendimiento del sistema recordamos, que como siempre que se compra algún programa, lo primero que se debe hacer con los discos en los que vienen los programas gráficos es co-

piarlos y guardar los originales. A las copias se deben añadir los ficheros de GSX necesarios.

Los programas comerciales de GSX incluirán, seguramente, un programa especial, el cargador GSX, cuya misión es cargar el sistema GSX en la memoria, al mismo tiempo que el propio sistema operativo. De no ser así, tendremos que incorporar el cargador a la aplicación.

Para ello, como ya hemos visto, se ejecuta la orden transitoria GENGRAF, que habremos copiado en el disco de trabajo, seguida del nombre del programa de aplicación. GENGRAF incorporará el cargador en el programa de aplicación con carácter permanente, de forma que ya no será necesario volver a ejecutar esta orden.

Así, preparado el cargador de GSX, todo lo que hay que hacer para ejecutar el programa de aplicación es escribir un nombre a continuación del *prompt*.

Utilización De Otras Impresoras Y Plotters

Si disponemos de un plotter compatible con el PCW 8256 o el CPC 6128, seguramente preferiremos utilizarlo como dispositivo de salida de los programas gráficos, puesto que su calidad es muy superior a la alcanzable con una impresora.

Después de asegurarnos de que el plotter es el adecuado, tendremos que comprobar si existe el controlador de dispositivo necesario.

Si el plotter es compatible con HP o tiene un interface serie, probablemente servirá el controlador DDHP7470.PRL, suministrado con los discos del sistema.

Aunque no conectemos un plotter, sí queremos al menos utilizar la impresora en modo de alta resolución, con una calidad de gráficos muy superior a la que se consigue en baja resolución, aunque no tan elevada como en un plotter. El controlador de dispositivo necesario es DDFXHR8.PRL (para PCW).

En cualquier caso, el controlador de dispositivo ha de ser copiado en el disco de trabajo, y también tendremos que modificar el fichero ASSIGN.SYS. Este contiene la lista de los dispositivos utilizables por el programa y los nombres de los correspondientes controladores.

Los dispositivos se identifican, tanto en el programa de aplicación como en el fichero ASSIGN.SYS, por su número de dispositivo lógi-

co. Estos son dos dígitos: los monitores suelen estar entre 01 y 10, los plotters entre 11 y 20 y las impresoras entre 21 y 30.

La norma es que si sólo se dispone de un dispositivo de una clase determinada, su número sea el más bajo de esta clase. Por ejemplo, si sólo se tiene conectado un monitor, su código será el 01. El máximo número de dispositivos mencionables en un fichero ASSIGN.SYS es cinco.

Para examinar la versión actual del fichero ASSIGN.SYS, podemos ejecutar una orden TYPE, tal como lo hacíamos con DRIVERS.GSX. Supongamos que obtenemos el siguiente resultado:

```
21 A:DDFXLR8 > SOLO EN PCW 8526
01 A:DDSCREEN>SOLO EN PCW 8526
```

Así, para incorporar un plotter al sistema, habría que editar este fichero e incluir en él la línea:

```
11 A: controlador plotter
```

El fichero ASSIGN.SYS se puede modificar mediante cualquier editor de texto, como puede ser el ED de CP/M.

La posición en la que se debe poner cada línea en el fichero ASSIGN.SYS depende de los tamaños de los ficheros en los que están grabados los controladores de dispositivo. GSX exige que el primero que se mencione sea el controlador más largo.

Además, para que el programa de aplicación envíe la salida gráfica al plotter en vez de a la impresora habrá que decirle que debe utilizar el dispositivo lógico 11 en lugar del 21.

Errores En Los Programas GSX

Veamos a continuación los distintos mensajes de error que pueden aparecer en un programa para GSX:

MENSAJE:	SOLUCION:
u:nombre.PRL not found	El controlador de dispositivo mencionado en ASSIGN.SYS no se encuentra en la unidad especificada. Comprobemos que hemos especificado el dispositivo deseado y que el nombre del controlador está bien escrito. Si todo lo demás

	falla, volvamos a copiar el controlador en el disco de trabajo.
u:nombre.PRL empty	Se ha encontrado el fichero del controlador de dispositivo, pero no tiene datos. Borrémoslo y volvamos a copiarlo en el disco de trabajo.
u:nombre.PRL contains absolute segment	Seguramente se ha estropeado el fichero del controlador de dispositivo. Volvamos a copiarlo en el disco de trabajo.
u:nombre.PRL close error	Seguramente hemos cambiado el disco estando abierto el fichero. Introduzcamos el disco correcto y volvamos a intentar la misma operación.
u:nombre.PRL load error	Seguramente no hemos puesto el nombre del controlador más largo el primero de la lista. Comprobemos y corrijamos esta circunstancia.

HELP

ste programa constituye una inestimable ayuda cuando la memoria nos falla respecto a nuestros conocimientos de un determinado comando o utilidad. Se trata de una especie de manual de instrucciones electrónico, que describe los programas de ayuda de CP/M Plus. Para ejecutarlo hemos de insertar el disco por la cara correspondiente y efectuar la llamada habitual de comando:

A>HELP

Inmediatamente después de haber elegido esta opción, el disco entrará en funcionamiento y a los pocos instantes aparecerán en la pantalla mensajes del tipo:

HELP UTILITY V1.1

At "HELP>" enter topic { , subtopic } ...

Example :HELP> DIR BUILT-IN

Topics available:

COMMANDS	CNTRLCHAR	COPYSYS	DATE	DEVICE
DIR	DISCKIT	DUMP	ED	ERASE
FILESPEC	GEMCOM	GET	GSX	HELP
HEXCOM	INITDIR	LANGUAGE	LIB	LINK
MAC	PALETTE	PAPER	PATCH	PIP
PRINTER	PUT	RENAME	REMAC	SAVE
SET	SET 24X80	SET DEF	SETKEYS	SETLST
SETSIO	SHOW	SID	SUBMIT	TYPE
USER	XREF			

Seguidamente, insertemos la opción que queramos estudiar; en unos instantes aparecerán listadas todas las diferentes observaciones que debemos tener en cuenta en el comando seleccionado, volviendo a preguntarnos si deseamos saber más sobre esta sentencia. Esto se concretará de la siguiente manera:

ENTER .subtopic FOR INFORMATION ON THE FOLLOWING SUBTOPICS:

OPTIONS MODIFIERS EXAMPLES

Si queremos ver un ejemplo, sólo tendremos que actuar de la siguiente manera:

HELP>.EXAMPLES

En este momento, HELP mostrará en la pantalla los diferentes ejemplos que puede darnos desde su memoria en el disco, acerca del comando cuya información general acaba de mostrar.

Si por el contrario, deseamos conocer algo más sobre las opciones o modificadores del comando, introduciremos, respectivamente:

HELP>.OPTIONS

o

HELP>.MODIFIERS

No obstante, no podemos acceder directamente a las opciones, por ejemplo, sin antes haber pasado por la información general del comando; es decir, no es posible introducir directamente algo como:

HELP>LIB.EXAMPLES

para acceder a los ejemplos de LIB.

Es importante saber que sólo podemos solicitar información sobre los comandos *(topics)* que se señalan en la lista mostrada por el programa.

13.7.1. Muestra de ejecución de HELP.

```
A>help


HELP UTILITY V1.1

At "HELP>" enter topic ¨,subtopic)...

EXAMPLE:  HELP> DIR BUILT-IN

Topics available:

COMMANDS      CNTRLCHARS    COPYSYS       DATE          DEVICE        DIR
DISCKIT       DUMP          ED            ERASE         FILESPEC      GENCOM
GET           GSX           HELP          HEXCOM        INITDIR       LANGUAGE
LIB           LINK          MAC           PALETTE       PAPER         PATCH
PIP (COPY)    PRINTER       PUT           RENAME        RMAC          SAVE
SET           SET24X80      SETDEF        SETKEYS       SETLST        SETSIO
SHOW          SID           SUBMIT        TYPE          USER          XREF

HELP> pip




  PIP (COPY)

Syntax:

              DESTINATION                    SOURCE

  PIP d:¨Gn) ñ filespec¨¡Gn¿) = filespec¨¡o¿),... ñ d:¨¡o¿)

Explanation:

The file copy program PIP copies files, combines files, and
transfers files between disks, printers, consoles, or other
devices attached to your computer.  The first filespec is the
destination. The  second filespec is the source.  Use two or more
source filespecs separated by commas to combine two or more files
into  one file.  ¡o¿ is any combination of the available options.
The ¡Gn¿ option in the destination filespec  tells  PIP  to  copy
your file to that user number.

PIP with no command tail displays an *  prompt  and  awaits  your
series  of  commands,  entered  and processed one line at a time.
The source or destination can be any CP/M Plus logical device.

ENTER .subtopic FOR INFORMATION ON THE FOLLOWING SUBTOPICS:

EXAMPLES      OPTIONS
```

```
HELP> .examples


   PIP (COPY)
     EXAMPLES


COPY A FILE FROM ONE DISK TO ANOTHER

      A>PIP b:=a:draft.txt
      A>PIP b:draft.txt = a:

      B3>PIP myfile.dat=A:¡G9¿
      A9>PIP B:¡G3¿=myfile.dat

COPY A FILE AND RENAME IT

      A5>PIP newdraft.txt=oldraft.txt
      C8>PIP b:newdraft.txt=a:oldraft.txt

COPY MULTIPLE FILES

      A>PIP b:=draft.*
      A>PIP b:=*.*
      B>PIP b:=c:.*.*
      C>PIP b:=*.txt¡g5¿
      C>PIP a:=*.com¡wr¿
      B>PIP a:¡g3¿=c:*.*

COMBINE MULTIPLE FILES

      A>PIP b:new.dat=file1.dat,file2.dat




COPY, RENAME AND PLACE IN USER 1

      A>pip newdraft.txt¡g1¿=oldraft.txt

COPY, RENAME AND GET FROM USER 1

      A>PIP newdraft.txt=oldraft.txt¡g1¿

COPY TO/FROM LOGICAL DEVICES

      A>PIP b:funfile.sue=con:
      A>PIP lst:=con:
      A>PIP lst:=b:draft.txt¡t8¿
      A>PIP prn:=b:draft.txt
```

```
HELP> .OPTIONS

  PIP (COPY)
    OPTIONS


PIP OPTIONS

A     Archive. Copy only files that have been  changed  since  the
       last copy.
C     Confirm. PIP prompts for confirmation before each file copy.
Dn    Delete any characters past column n.
E     Echo transfer to console.
F     Filter form-feeds from source data.
Gn    Get from or go to user n.
H     Test for valid Hex format.
I     Ignore :00 Hex data records and test for valid Hex format.
K     Kill display of filespecs on console.
L     Translate upper case to lower case.
N     Number output lines
O     Object file transfer, ^Z ignored.
Pn    Set page length to n.  (default n=60)
Qs^Z  Quit copying from source at string s.
R     Read files that have been set to SYStem.
Ss^Z  Start copying from the source at the string s.
Tn    Expand tabs to n spaces.
U     Translate lower case to upper case.
V     Verify that data has been written correctly.
W     Write over Read Only files without console query.
Z     Zero the parity bit.

All  options  except  C,G,K,O,R,V  and  W  force  an  ASCII  file
     transfer, character by character, terminated by a ^Z.

HELP>
```

LANGUAGE

El sistema operativo CP/M ha sido diseñado para su difusión a nivel mundial, debido a lo cual implementa gran cantidad de juegos de caracteres, que completando a los 127 del código ASCII estándar, facilitan la adaptación de ciertos caracteres a las necesidades específicas de las aplicaciones destinadas a funcionar en determinados idiomas, tal como pueden ser acentos o caracteres como la eñe, en el caso concreto del castellano.

Para la obtención en la pantalla del juego de caracteres adecuado, CP/M dispone del comando LANGUAGE, que deberá ir seguido de un número indicativo del juego a seleccionar. En nuestro caso concreto, los de uso más frecuente serán LANGUAGE 0, para el estadounidense, LANGUAGE 3, para el británico y LANGUAJE 7, para el español. La única diferencia entre los dos primeros estriba en el intercambio del símbolo libra (£) y *number* (#). No obstante, en el caso del LANGUAGE 7 las diferencias son mayores:

DECIMAL	HEX	IDIOMA 0	IDIOMA 7
35	#23	#	Pt
91	#5B	[	¡
92	#5C	\	Ñ
93	#5D	]	¿
123	#7B	{	¨
124	#7C	|	ñ

En definitiva, la finalidad de LANGUAGE no es otra que alterar la configuración de ALGUNOS caracteres de CP/M, lo que en el caso del PCW puede suponer una incidencia directa sobre el BASIC, dado que este lenguaje se carga tras la implantación del sistema operativo en la máquina, al contrario que en los CPC que viene grabado en ROM, entrando en acción en el mismo momento de inicializar el ordenador.

PROBLEMAS CON LANGUAGE

En la práctica, esta herramienta de compatibilización nos puede llevar a una falta de estandarización. Tengamos en cuenta que, si bien en la versión de CP/M proporcionada con los CPC el idioma por defecto es el cero (EE.UU.), en los PCW es el LANGUAGE 7. Dada esta circunstancia, el poseedor de este aparato se encontrará en ocasiones con problemas de cuya procedencia no tenía idea al iniciar la lectura de este párrafo. Así, por ejemplo, puede volverse loco buscando los signos de apertura y cerrado de corchetes para acceder a algunas opciones del propio CP/M. Este problema se resuelve pulsando, respectivamente, EXTRA + 1 y EXTRA + ?

Aunque la representación obtenida en pantalla haya sido la apertura de admiración y de interrogación, los códigos emitidos realmente habrán sido los correspondientes a los corchetes.

Otro inconveniente, quizá de mayor envergadura, se habrá presentado al utilizar el comando PRINT USING desde BASIC, dado que éste hace uso del símbolo almohada, *number* o *hush* (#), cuya tecla específica en el PCW no genera el código adecuado. Así, se dará la paradoja de que aunque en la pantalla veamos la instrucción escrita

correctamente, el intérprete se empeñará en detectar un error de sintaxis. Curiosamente, para que la instrucción funcione, lo visualizado en la pantalla deberá ser el signo peseta (Pt), en vez de la almohada (#), que se puede obtener con EXTRA + 4, dado que aunque la visualización no sea la correcta sí lo será el código emitido.

Algo lioso, ¿verdad? Realmente, quizá el esfuerzo «castellanizador» de Amstrad España haya sido excesivo en el caso del PCW, ya que probablemente reporte más problemas que beneficios, sobre todo dedicando el ordenador a tareas ajenas al uso de caracteres castellanos, como puede ser el trabajo con CP/M o BASIC. De todas formas, el problema es bastante fácil de solventar: LANGUAGE 0.

Finalmente, utilizando la orden LANGUAGE nos encontraremos con bastante frecuencia con un problema curioso, por muy experimentados que seamos y sobre todo cuando estemos operando con prisa, pensando en otra cosa, o efectuando alguna operación semiautomática, como por ejemplo: encender nuestro PCW, cargar CP/M, pasar el idioma cero y cargar el BASIC. Este problema vendrá tras escribir:

A>languaje 0

y se sintetizará en un inquisitivo mensaje...

LANGUAJE?
A>

Incluso quizá algún lector no haya aún adivinado dónde está el inconveniente. Los idiomas europeos, entre ellos el castellano y el inglés, comparten palabras muy parecidas, que en una traducción a la ligera nos pueden llevar a un error en su significado o en su ortografía, por su gran parecido fonético con una palabra de nuestro mismo idioma; son los denominados «falsos amigos». LANGUAGE es uno de ellos, y no debemos extrañarnos si un día, por muchas veces que lo hayamos hecho hasta el momento, nos sorprendemos escribiendo LANGUAJE (con jota y no ge), siguiendo una ortografía castellana casi instintiva.

Estemos al tanto de este «insignificante» detalle, ya que de no ser así no seremos los primeros que al ver el mensaje LANGUAJE? en nuestra pantalla perdemos cinco minutos con comprobaciones de la cara del disco, directorios para constatar la presencia del LANGUAGE.COM, pensando en algún borrado catastrófico, o apagando y encendiendo el ordenador para volver a cargar el CP/M con un «¡¿Pero qué pasa?!» en nuestra cabeza, entre asustados y admirados por el mal funcionamiento de nuestro equipo.

Adaptación De Impresoras Al Uso De Idioma

Si poseemos una impresora, normalmente haremos uso de ella casi tanto como de la pantalla, por lo que a la hora de alterar el idioma a utilizar, es conveniente tener en cuenta el estado en que se encuentra la impresora a este respecto.

La selección del juego de caracteres apropiado en uno de estos periféricos, generalmente se consigue mediante la acción sobre *microswitches*, que normalmente debe ser llevada a cabo ANTES de encender la impresora, para que su efecto sea tenido en cuenta.

En otras ocasiones, el cambio al juego de caracteres deseado se puede efectuar por software; ya sea simplemente gracias a LANGUAGE o mediante la emisión de una serie de códigos de control.

En el primer caso, recordamos que la orden deberá ir seguida de un número, pues en caso contrario se emitirá el mensaje de error *Bad number* (código erróneo). Dicho código de idioma deberá estar comprendido entre cero y siete, si bien números superiores a éste no serán rechazados, sino simplemente reducidos por módulo a la escala 0-7. Así, por ejemplo, el efecto de LANGUAGE 8 será el mismo que el de LANGUAGE 0 o el de LANGUAGE 19 el mismo que el de LANGUAGE 3 (3=19 MOD 8).

En cuanto a la emisión de caracteres de control para el cambio de idioma, normalmente suele ser la secuencia [ESC] 2 <n>, donde <n> es el código de idioma:

0 - Estados Unidos
1 - Francia
2 - Alemania
3 - Gran Bretaña
4 - Dinamarca
5 - Suecia
6 - Italia
7 - España

No obstante, los usuarios de los CPC deberán siempre tener en cuenta que el port centronics de su aparato tan sólo emite los siete primeros bits, quedando el último (bit 7), a efectos del código de carácter, siempre a cero. Así, pues, aun seleccionando distintos juegos de caracteres, no les será posible acceder a aquellos que se encuentren por encima del 127, lugar en el cual, desgraciadamente, se suelen ubicar las vocales acentuadas, eñe, etc.

12.8.1. Juego completo de caracteres español.

Decimal	Hexadecimal	Símbolo	Descripción	Segundo significado
0	#00	∞	infinito	control-@
1	#01	⊙	flecha saliente del papel	control-A
2	#02	Γ	gamma mayúscula	control-B
3	#03	Δ	delta mayúscula	control-C
4	#04	⊗	flecha entrante en el papel	control-D
5	#05	×	multiplicar	control-E
6	#06	÷	dividir	control-F
7	#07	∴	por consiguiente	control-G
8	#08	Π	pi mayúscula	control-H
9	#09	↓	flecha abajo	control-I
10	#0A	Σ	sigma mayúscula	control-J
11	#0B	←	flecha a la izquierda	control-K
12	#0C	→	flecha a la derecha	control-L
13	#0D	±	más menos	control-M
14	#0E	↔	flecha izquierda y derecha	control-N
15	#0F	Ω	omega mayúscula	control-O
16	#10	α	alfa minúscula	control-P
17	#11	β	beta minúscula	control-Q
18	#12	γ	gamma minúscula	control-R
19	#13	δ	delta minúscula	control-S
20	#14	ε	épsilon minúscula	control-T
21	#15	θ	teta minúscula	control-U
22	#16	λ	lambda minúscula	control-V
23	#17	μ	mu minúscula	control-W
24	#18	π	pi minúscula	control-X
25	#19	ρ	rho minúscula	control-Y
26	#1A	σ	sigma minúscula	control-Z
27	#1B	τ	tau minúscula	control-[
28	#1C	φ	fi minúscula	control-\
29	#1D	χ	ji minúscula	control-]
30	#1E	ψ	psi	control-↑
31	#1F	ω	omega minúscula	control-_
32	#20		espacio	
33	#21	!	cerrar admiración	
34	#22	"	comillas	
35	#23	Pt	Peseta	
36	#24	$	dólar	
37	#25	%	por ciento	
38	#26	&	etcétera ('ampersand')	
39	#27	'	apóstrofo	

40	#28	(	abrir paréntesis
41	#29	)	cerrar paréntesis
42	#2A	*	asterisco
43	#2B	+	más
44	#2C	,	coma
45	#2D	-	menos
46	#2E	.	punto
47	#2F	/	barra
48	#30	0	cero
49	#31	1	uno
50	#32	2	dos
51	#33	3	tres
52	#34	4	cuatro
53	#35	5	cinco
54	#36	6	seis
55	#37	7	siete
56	#38	8	ocho
57	#39	9	nueve
58	#3A	:	dos puntos
59	#3B	;	punto y coma
60	#3C	<	menor
61	#3D	=	igual
62	#3E	>	mayor
63	#3F	?	cerrar interrogación
64	#40	@	arroba ('at')
65	#41	A	A mayúscula
66	#42	B	B mayúscula
67	#43	C	C mayúscula
68	#44	D	D mayúscula
69	#45	E	E mayúscula
70	#46	F	F mayúscula
71	#47	G	G mayúscula
72	#48	H	H mayúscula
73	#49	I	I mayúscula
74	#4A	J	J mayúscula
75	#4B	K	K mayúscula
76	#4C	L	L mayúscula
77	#4D	M	M mayúscula
78	#4E	N	N mayúscula
79	#4F	O	O mayúscula

80	#50	P	P mayúscula
81	#51	Q	Q mayúscula
82	#52	R	R mayúscula
83	#53	S	S mayúscula
84	#54	T	T mayúscula
85	#55	U	U mayúscula
86	#56	V	V mayúscula
87	#57	W	W mayúscula
88	#58	X	X mayúscula
89	#59	Y	Y mayúscula
90	#5A	Z	Z mayúscula
91	#5B	¡	abrir admiración
92	#5C	Ñ	Ñ mayúscula
93	#5D	¿	abrir interrogación
94	#5E	↑	flecha arriba
95	#5F	_	subrayado
96	#60	`	acento grave
97	#61	a	A minúscula
98	#62	b	B minúscula
99	#63	c	C minúscula
100	#64	d	D minúscula
101	#65	e	E minúscula
102	#66	f	F minúscula
103	#67	g	G minúscula
104	#68	h	H minúscula
105	#69	i	I minúscula
106	#6A	j	J minúscula
107	#6B	k	K minúscula
108	#6C	l	L minúscula
109	#6D	m	M minúscula
110	#6E	n	N minúscula
111	#6F	o	O minúscula
112	#70	p	P minúscula
113	#71	q	Q minúscula
114	#72	r	R minúscula
115	#73	s	S minúscula
116	#74	t	T minúscula
117	#75	u	U minúscula
118	#76	v	V minúscula
119	#77	w	W minúscula
120	#78	x	X minúscula
121	#79	y	Y minúscula

122	#7A	z	Z minúscula
123	#7B	¨	diéresis
124	#7C	ñ	Ñ minúscula
125	#7D	}	cerrar llaves
126	#7E	~	tilde
127	#7F	0	cero sin barra
128-159	#80-#9F		códigos expansibles
160	#A0	ª	a voladita
161	#A1	º	o voladita
162	#A2	°	grados
163	#A3	£	libra
164	#A4	©	copyright
165	#A5	¶	calderón
166	#A6	§	párrafo
167	#A7	†	cruz
168	#A8	¼	un cuarto
169	#A9	½	un medio
170	#AA	¾	tres cuartos
171	#AB	«	abrir comillas
172	#AC	»	cerrar comillas
173	#AD	#	n.° ('hashmark')
174	#AE	]	cerrar corchetes
175	#AF	[	abrir corchetes
176	#B0	*f*	florín
177	#B1	¢	centavo
178	#B2	{	abrir llaves
179	#B3	´	acento agudo
180	#B4	^	acento circunflejo
181	#B5	‰	por mil
182	#B6	⅛	un octavo
183	#B7	⅜	tres octavos
184	#B8	⅝	cinco octavos
185	#B9	⅞	siete octavos
186	#BA	ß	doble S alemana
187	#BB	○	círculo
188	#BC	●	topo
189	#BD	¥	yen
190	#BE	®	marca registrada
191	#BF	™	marca comercial

192	#C0	Á	A mayúscula agudo
193	#C1	É	E mayúscula agudo
194	#C2	Í	I mayúscula agudo
195	#C3	Ó	O mayúscula agudo
196	#C4	Ú	U mayúscula agudo
197	#C5	Â	A mayúscula circunflejo
198	#C6	Ê	E mayúscula circunflejo
199	#C7	Î	I mayúscula circunflejo
200	#C8	Ô	O mayúscula circunflejo
201	#C9	Û	U mayúscula circunflejo
202	#CA	À	A mayúscula grave
203	#CB	È	E mayúscula grave
204	#CC	Ì	I mayúscula grave
205	#CD	Ò	O mayúscula grave
206	#CE	Ù	U mayúscula grave
207	#CF	Ÿ	Y mayúscula diéresis
208	#D0	Ä	A mayúscula diéresis
209	#D1	Ë	E mayúscula diéresis
210	#D2	Ï	I mayúscula diéresis
211	#D3	Ö	O mayúscula diéresis
212	#D4	Ü	U mayúscula diéresis
213	#D5	Ç	C mayúscula con cedilla
214	#D6	AE	diptongo AE mayúscula
215	#D7	Å	A mayúscula círculo
216	#D8	Ø	O mayúscula barra
217	#D9	\	barra a la izquierda
218	#DA	Ã	A mayúscula tilde
219	#DB	Õ	O mayúscula tilde
220	#DC	⩾	mayor o igual
221	#DD	⩽	menor o igual
222	#DE	≠	distinto
223	#DF	≃	aproximadamente igual
224	#E0	á	A minúscula agudo
225	#E1	é	E minúscula agudo
226	#E2	í	I minúscula agudo
227	#E3	ó	O minúscula agudo
228	#E4	ú	U minúscula agudo
229	#E5	â	A minúscula circunflejo
230	#E6	ê	E minúscula circunflejo
231	#E7	î	I minúscula circunflejo

232	#E8	ô	O minúscula circunflejo
233	#E9	û	U minúscula circunflejo
234	#EA	à	A minúscula grave
235	#EB	è	E minúscula grave
236	#EC	ì	I minúscula grave
237	#ED	ò	O minúscula grave
238	#EE	ù	U minúscula grave
239	#EF	ÿ	Y minúscula diéresis
240	#F0	ä	A minúscula diéresis
241	#F1	ë	E minúscula diéresis
242	#F2	ï	I minúscula diéresis
243	#F3	ö	O minúscula diéresis
244	#F4	ü	U minúscula diéresis
245	#F5	ç	C minúscula con cedilla
246	#F6	æ	diptongo AE minúscula
247	#F7	å	A minúscula círculo
248	#F8	ø	O minúscula barra
249	#F9	|	barra vertical
250	#FA	ã	A minúscula tilde
251	#FB	õ	O minúscula tilde
252	#FC	⇒	flecha doble a la derecha
253	#FD	⇐	flecha doble a la izquierda
254	#FE	⇔	flecha doble izquierda/derecha
255	#FF	≡	equivalente

HEX	DEC	EE.UU.	FRANCIA	ALEMANIA	R.U.	DINAMARCA	SUECIA	ITALIA	ESPAÑA
23H	35	# £	# £	# £	£ £	# £	# £	# £	₧ ₧
24H	36	$ $	$ $	$ $	$ $	$ $	¤ ¤	$ $	$ $
40H	64	@ @	à à	§ §	@ @	@ @	É É	@ @	@ @
5BH	91	[[	° °	Ä Ä	[[	Æ Æ	Ä Ä	° °	¡ ¡
5CH	92	\ \	ç ç	Ö Ö	\ \	Ø Ø	Ö Ö	\ \	Ñ Ñ
5DH	93	]]	§ §	Ü Ü	]]	Å Å	Å Å	é é	¿ ¿
5EH	94	^ ^	^ ^	^ ^	^ ^	^ ^	Ü Ü	^ ^	^ ^
60H	96	` `	` `	` `	` `	` `	é é	ù ù	` `
7BH	123	{ {	é é	ä ä	{ {	æ æ	ä ä	à à	¨ ¨
7CH	124	| |	ù ù	ö ö	| |	ø ø	ö ö	ò ò	ñ ñ
7DH	125	} }	è è	ü ü	} }	å å	å å	è è	} }
7EH	126	~ ~	¨ ¨	ß ß	~ ~	~ ~	ü ü	ì ì	~ ~

13.8.2. Diferencias en los juegos de caracteres de los distintos idiomas.

```
CP/M Plus  Amstrad Consumer Electronics plc

v 1.2, 61K TPA, 1 disco, 112K disco M:

A)LANGUAGE
Bad number

A)LANGUAGE 0

A)Observemos el simbolo # (EXTRA 4)
OBSERVEMOS?
A)LANGUAGE 7

A)Observemos el mismo simbolo : #
OBSERVEMOS?
A)LANGUAGE 3

A)De nuevo £
DE?
A)LANGUAGE 8

A)Probemos con #
PROBEMOS?
A)LANGUAJE 7
LANGUAJE?
A)
```

13.8.3. Ejecución de muestra de órdenes LANGUAGE.

LIB

l comando LIB se encuentra en estrecha relación con RMAC, del cual haremos mención más adelante y, por tanto, se destina como éste al tratamiento de ficheros que contienen información en código máquina, también denominados OBJETO.

Concretamente, LIB gestiona una biblioteca de módulos objetos, ya sea creándola o seleccionando, añadiendo, reemplazando o borrando módulos de una ya existente, con la restricción de que éstos deben tener la extensión .REL, siguiendo el formato con el que son producidos por RMAC.

La sintaxis de este comando es la siguiente:

LIB fichero {[I|M|P|D]}

LIB fichero{[I|M|P]}=fichero {modificador}
{,fichero{modificador}...}

Así, pues, consideramos una biblioteca (en inglés, *LIBrary)* una colección de módulos de código objeto, que gracias a LIB puede ser completamente gestionada, generando, eliminando o alterando cualquiera de sus módulos.

Gracias al propio LINK 80 de CP/M podremos unir el contenido de varios objetos en una biblioteca u otros ficheros, puesto que actúa seleccionando automáticamente para la biblioteca, sólo aquellos módulos necesarios para que el programa sea conjuntado, formando así un fichero ejecutable tipo comando (.COM).

Las opciones de trabajo de LIB son las siguientes:

I: La operación INDEX crea una biblioteca de ficheros del tipo .IRL. LINK 80 busca antes y más rápidamente sobre este tipo de bibliotecas que sobre las comunes.

M: La opción modular expone el nombre de los módulos que conforman la biblioteca.

P: La opción común *(Public)* presenta el nombre de los módulos y las variables comunes para una nueva biblioteca de ficheros.

D: La opción DUMP presenta el contenido de los objetos modulados en forma de código ASCII.

MODIFICADORES

El manejo de modificadores en el uso de la instrucción LIB se utiliza principalmente para borrar, reemplazar o seleccionar módulos en una biblioteca de ficheros. En su sintaxis, con llaves se encierra el módulo a borrar o reemplazar y con paréntesis los que van a ser seleccionados.

Borrar	{Módulo=}
Reemplazar	{Módulo= nombre fichero.R}
	Si el nombre del módulo y el fichero tienen la misma ortografía puede utilizarse.
	{nombre fichero}
Selección	(Mód PRIMER-mód ULT, mód 1, mód 2, ... mód N)

ALGUNOS EJEMPLOS

Para una mejor comprensión del funcionamiento de este comando, veremos a continuación algunos ejemplos de utilización:

A>LIB PRUEBA[P]

Presenta la totalidad de los módulos y sus comunes en PRUEBA-.REL.

A>LIB PRUEBA[P]=FICH1,FICH2

Crea PRUEBA desde FICH1.REL y FICH2.REL y presenta todos los módulos y sus comunes en PRUEBA.REL.

A>LIB PRUEBA1(MOD1,MOD2),PRUEBA2(A1-A2,A5)

Crea una biblioteca de ficheros desde dos líneas distintas de ficheros.

A>LIB PRUEBA1=PRUEBA<MOD1=>

Crea PRUEBA1.REL desde PRUEBA.REL, omitiendo MOD1, que es un módulo en el fichero PRUEBA.REL.

A>LIB PRUEBA1=PRUEBA<MOD1=FICH1.REL>

Crea PRUEBA1.REL a partir de PRUEBA.REL, reemplazando el módulo de ésta última MOD1 por FICH1.REL.

A>LIB PRUEBA=PRUEBA1<NOMBRE>

Partimos de la base de la existencia de un supuesto módulo NOMBRE dentro de PRUEBA1.REL. Cuando LIB crea PRUEBA.REL a partir de PRUEBA1.REL, un fichero NOMBRE.REL reemplaza el módulo de igual denominación (NOMBRE).

A>LIB PRUEBA[I]=B:PRUEBA1(MOD1,MOD2,MOD3-MOD6)

Rizando el rizo, crearemos una biblioteca indexada PRUEBA.IRL, en la unidad A, con una selección de los módulos MOD1, MOD2 y MOD3 a MOD6 desde la unidad B PRUEBA1.REL.

LINK

omo ya hemos apuntado en el capítulo anterior, el objetivo de LINK es combinar módulos de código objeto, reubicándolos para conformar un fichero ejecutable tipo .COM.

La restricción existente para el correcto desenvolvimiento de su trabajo es que dichos módulos deben distribuirse en forma de ficheros individuales, por ejemplo, como los que genera RMAC, o bien en bibliotecas de ficheros, de cuya construcción ya hablamos al tratar el comando LIB.

OPCIONES DE LINK-80

A ..... Memoria adicional; reduce el espacio de buffer y escribe datos temporales al disco.

B ..... BIOS LINK en soporte CP/M Plus.
1 - Alinea segmento de datos sobre límite de página.
2 - Establece longitud de segmento de código en cabecera.
3 - Fichero tipo .SPR por defecto.

D hhhh ..... Origen de datos; fija origen, dirección de memoria y área de datos.

Gn ..... Ir; establece la etiqueta de comienzo en "n".

L hhhh..... Carga; cambia la dirección por defecto de carga del módulo a hhhh. Valor por defecto 0100H.

M hhhh ..... Tamaño de memoria; define los requerimientos de memoria libre para módulos de MP/M.

NL ..... No lista la tabla de símbolos por consola.

NR ..... No hay fichero de tabla de símbolos.

OC ..... Fichero comando Salida.COM. Tomado por defecto.

OP ..... Fichero reubicable de página Salida.PRL, para ejecución bajo MP/M, en segmentos reubicables.

OR ..... Ficheros de proceso de sistema residente Salida.RSP, para ejecución bajo MP/M.

OS ..... Fichero reubicable de página de sistema Salida.SPR, para ejecución bajo MP/M.

P hhhh..... Program origin (origen de programa); cambia la dirección por defecto de origen del programa a hhhh. Su valor por defecto es 0100H.

Q ..... Lista símbolos que comienzan por interrogación.

S ..... Búsqueda del fichero precedente como una biblioteca.

$Cd ..... El destino de los mensajes de consola "d" pueden ser "X" (consola), "Y" (impresora) o "Z" (sin salida). Por defecto se considera X.

$Id ..... Fuente de ficheros intermedios; "d" es la unidad de disco A-P. Por defecto es la unidad actual.

$Ld..... Fuente de ficheros de biblioteca; "d" es una unidad de disco A-P. Por defecto se considera la unidad actual.

$Od ..... Destino de fichero objeto; "d" puede ser Z o unidad de disco A–P. Por defecto es la misma unidad que el primer fichero en el comando LINK-80.COM.

$Sd ..... Destino de ficheros de símbolo. "d" puede ser Y o Z o una unidad de disco A-P. Por defecto es la misma unidad que el primer fichero en el comando LINK-80.

EJEMPLOS

Veamos algunos ejemplos para colaborar a la comprensión de este comando:

A>LINK b:PRUEBA[NR]

LINK-80 actúa sobre el fichero PRUEBA.REL en la unidad B, produciendo sobre dicha unidad un fichero en código máquina directamente ejecutable, del tipo PRUEBA.COM. La opción [NR] le comunica que no debe generar fichero con la tabla de símbolos.

A>LINK M1,M2,M3

LINK-80 combina separadamente los ficheros compilados M1, M2 y M3, resuelve sus referencias externas y produce el fichero M1 en código máquina ejecutable.

A>LINK M=M1,M2,M3

LINK-80 combina separadamente los ficheros compilados M1, M2 y M3 y produce el M.COM, ejecutable en código máquina.

A>LINK PRUEBA,FICH1[s]

La opción [s] llama a LINK-80 para registrar FICH1 dentro de una biblioteca de ficheros. LINK-80 combina PRUEBA.REL con las referencias contenidas en las subrutinas de FICH1.REL y produce un PRUEBA.COM sobre la unidad de disco por defecto.

MAC

La misión de MAC es generar ficheros con contenido de código objeto, a partir de ficheros fuente en ensamblador. La restricción al fichero de entrada es que debe tener el identificador .ASM, produciéndose salidas del tipo .HEX de contenido hexadecimal, .PRN, adaptado a su edición por la pantalla o impresora y .SYM, con la lista de símbolos definidos por el programa.

Algunos ejemplos de uso serían:

A>MAC PRUEBA

A>MAC PRUEBA $ PB AA HB SX

El uso de las funciones directas de MAC puede ser de OUTPUT o INPUT. Utilizando una marca con la opción a indicar la fuente y destino de unidad de disco, consola o salida cero.

X, P y Z especifican por este orden consola, impresora y salida cero respectivamente.

Estas son las funciones ensamblador que funcionan directamente con OUTPUT / INPUT.

A	Unidad de disco fuente para fichero ASM (A-O).
H	Unidad de disco de destino para fichero HEX. (A-O,Z).
L	Unidad de disco fuente para macrobiblioteca .LIB. Fichero llamado por la instrucción MACLIB.
P	Unidad de disco de destino para .PRN fichero (A-O,X,P,Z).
S	Unidad de disco de destino para .SYM fichero.

Las opciones del ensamblador que modifican el contenido de salida del fichero son:

+1	Produce un listado en primera pasada para macro depuración en ficheros .PRN.
−1	Suprime el listado en primera pasada (por defecto).
+L	Lista líneas de entrada leídas desde ficheros macrobiblioteca .LIB.
−L	Suprime listado (por defecto).
+M	Lista todas las macrolíneas tal como se procesan durante el ensamblado.
−M	Suprime todas las macrolíneas tal como se procesan durante el ensamblado.
*M	Lista sólo los HEX generados por macro expansiones.
+Q	Lista todos los símbolos LOCAL en la lista de símbolos.
−Q	Suprime todos los símbolos LOCAL en el listado de símbolos (por defecto).

```
A>MAC m:PRUEBA
CP/M MACRO ASSEM 2.0
A:PRUEBA.ASM-NO SOURCE FILE PRESENT

A>
```

13.11.1. El primer proceso de MAC es investigar la existencia del fichero fuente en el disco.

+S	Añade fichero de símbolos a fichero de impresión.
−S	Suprime la creación de fichero de símbolos.

En el caso de los parámetros A, H, L, P y S, deben estar seguidos por el nombre de la unidad en la cual obtener los datos o, en su caso, a la que deben pasar los mismos.

ERRORES DE MAC

Al finalizar su labor, puede que el ensamblador emita alguno de los siguientes mensajes de error:

NO SOURCE FILE PRESENT No existe fichero fuente	Normalmente este error se produce de una denominación inapropiada, o bien por carecer de la extensión .MAC.
NO DIRECTORY SPACE No queda espacio en el directorio	Durante la generación de los ficheros tipo .HEX o .PRN, el ensamblador no encuentra espacio suficiente en el directorio.
OUTPUT FILE WRITE ERROR Error de escritura en el fichero de salida	El disco de destino de la información está protegido contra escritura o completo.
CANNOT CLOSE FILE No es posible cerrar el fichero	Probablemente el disco de destino está protegido contra escritura.
SOURCE FILE NAME ERROR Error de nombre del fichero fuente	En la fuente existen caracteres no reconocibles, probablemente debido al uso de ficheros no compatibles.

SOURCE FILE READ ERROR

Error de lectura en el fichero fuente

Generalmente un error físico en el fichero fuente es el que motiva este mensaje.

UNBALANCE MACRO LIBRARY

No existe fin de definición de macro

No encuentra ENDM para una definición de MACRO.

INVALID PARAMETER

Parámetro erróneo

Parámetro erróneo de ensamblado localizado en la línea de entrada.

PALETTE

Los colores iniciales en que CP/M Plus presenta la información en pantalla son el blanco brillante, para el primer término (caracteres) y azul oscuro para el fondo; ello se traduce en un monitor en fósforo verde, obviamente, a las correspondientes gradaciones de este tono.

No obstante, es posible alterar esta configuración inicial, gracias al uso del comando PALETTE, acompañada de diversos parámetros que indican las tintas a emplear, tanto para el fondo como para el primer plano. Inicialmente, dichos valores son el 0 y el 1, respectivamente, aunque CP/M nos permite generar hasta 64 códigos diferentes, numerados de 0 a 63.

En cuanto a las tintas que es posible emplear, son 16 diferentes, aunque sólo las dos primeras son visibles en el modo de 80 columnas.

Así, por ejemplo, el código 1 equivale al color azul y el 63 al blanco intenso, de forma que si quisiéramos alterar los colores de fondo y primer término sólo tendríamos que ejecutar la orden PALETTE 63,1.

Gracias a la siguiente tabla, podremos elegir adecuadamente los colores que más convengan a las aplicaciones concretas. Insistimos que en el caso de un monitor de fósforo verde, podremos apreciar las distintas tonalidades de verde, salvo en el caso del PCW, cuya característica particular en este sentido trataremos más adelante.

COLOR	DEC.
NEGRO	0
AZUL	2
AZUL INTENSO	3
ROJO	8
MAGENTA	10
MALVA	11
ROJO INTENSO	12
MORADO	14
MAGENTA INTENSO	15
VERDE	32
CYAN	34
AZUL CELESTE	35
AMARILLO	40
BLANCO	42
AZUL PASTEL	43
NARANJA	44
ROSA	46
MAGENTA PASTEL	47
VERDE INTENSO	48
VERDE MAR	50
CYAN INTENSO	51
VERDE LIMA	56
VERDE PASTEL	58
CYAN PASTEL	59
AMARILLO INTENSO	60
AMARILLO PASTEL	62
BLANCO INTENSO	63

Como ya hemos anticipado, el comportamiento de esta orden en los PCW es sensiblemente diferente, dado que la pantalla de este aparato no es capaz de generar más que un tono oscuro y otro verde, debido a lo cual sólo es posible obtener dos combinaciones:

PALETTE 1 0 Establece «vídeo inverso» (caracteres oscuros sobre fondo claro).

PALETTE 0 1 Establece «vídeo normal» (caracteres claros sobre fondo oscuro).

PATCH

l comando PATCH incorpora o muestra un «parche» (este es su significado en inglés) en un fichero tipo comando de CP/M, cuyo código estará comprendido entre 1 y 32.

La forma genérica de este comando es:

PATCH nombre-fichero {.EXT} {n}

Así, por ejemplo, la orden

A>PATCH PRUEBA

muestra el parche en el fichero PRUEBA.

A>PATCH PRUEBA.PRL 2

Establece el *patch* número 2 en el fichero PRUEBA.PRL.

```
A>PATCH m:PIP

CP/M 3 PATCH - Version 3.0
Current patches for PIP.COM:
  1 2


A>
```

13.13.1. Visualización de los patch *de PIP.COM en la unidad m:.*

```
A>patch m:pip.com 32

CP/M 3 PATCH - Version 3.0
Do you want to indicate that patch 32
  has been installed for PIP.COM ? y

Patch installed


A>
```

13.13.2. Antes de efectuar su labor, PATCH emite un mensaje para confirmación.

```
A>
A>patch m:pip.com 32

CP/M 3 PATCH - Version 3.0
Do you want to indicate that patch 32
  has been installed for PIP.COM ? n

Patch not installed
```

13.13.3. En caso de confirmación, el patch *queda registrado en el disco.*

```
A>patch m:pip.com

CP/M 3 PATCH - Version 3.0
Current patches for PIP.COM:
  1 2 32


A>
```

13.13.4. La instalación del patch *puede ser comprobada gracias al empleo directo de la misma orden.*

PUT

UT ordena al sistema que en el futuro desvíe a un fichero especificado todas las salidas que normalmente se canalizarían a través de la consola (pantalla) o la impresora, hasta que se indique otra cosa, o bien finalice el programa que se ejecuta a continuación.

Es posible incluir una o varias opciones, situándolas entre corchetes al final de la línea de comando, separadas unas de otras por comas o espacios. Además, si no se especifica otra cosa, los caracteres de control no serán grabados en el fichero y la salida será reproducida en la pantalla, pero no en la impresora.

ESPECIFICADORES

Como ya hemos dicho, son varios los especificadores que pueden modificar la acción concreta de PUT. Estos son: ECHO, NO ECHO, FILTER, NO FILTER, SYSTEM. Veámoslos más en detalle:

ECHO: Especifica que la salida es enviada también a la consola. Esta opción es tomada por defecto.

NO ECHO: Especifica que la salida no debe enviarse a la consola. Esta opción es tomada por defecto en la ejecución de PUT PRINTER.

FILTER: Activa el filtrado de caracteres de control, gracias al cual éstos se sustituyen por los correspondientes códigos representables.

NO FILTER: Esta opción, tomada por defecto, suprime el filtrado de los caracteres de control.

SYSTEM: Especifica que la salida del sistema, así como la del programa, se dirige al fichero especificado en PUT. Esta circunstancia se mantiene hasta que un comando PUT CONSOLE redirecciona la salida de nuevo hacia la misma.

EJEMPLOS

Veamos algunos ejemplos genéricos con la descripción de sus efectos.

PUT CONSOLE OUTPUT TO FILE nombre

Dirige la salida al fichero NOMBRE, al tiempo que envía la misma información a la consola (pantalla).

PUT PRINTER OUTPUT TO FILE nombre

```
A>
A>put console output to file m:teclado
Putting console output to file: M:TECLADO.
A>Esto quedara registrado en el teclado.
ESTO?
A>Si, esto.
SI,?
A>put console output to console
Putting console output to console
A>type m:teclado
ESTO?
A>
```

13.14.1. Muestra del efecto de la desviación de consola a un fichero en disco.

Los datos se dirigen en este caso, simultáneamente, al fichero NOMBRE y a la impresora.

PUT CONSOLE OUTPUT TO FILE nombre [SYSTEM]

Su efecto es dirigir la salida por consola y del sistema al fichero NOMBRE, hasta que recibe orden en contra mediante PUT CONSOLE OUTPUT TO CONSOLE.

PUT PRINTER OUTPUT TO FILE nombre [SYSTEM]

Su efecto es muy similar al anterior, aunque en este caso la información desviada al fichero es la de la impresora. Para volver al estado normal será necesario ejecutar una orden PUT PRINTER OUTPUT TO PRINTER.

PUT CONSOLE OUTPUT TO FILE nombre [NO ECHO]

Suprime las representaciones en la pantalla de la salida por consola, desviando la salida al fichero NOMBRE.

PUT PRINTER OUTPUT TO FILE nombre [ECHO]

Activa la reproducción en la impresora de la salida enviada al fichero NOMBRE.

PUT CONSOLE OUTPUT TO FILE nombre [FILTER]

En este ejemplo, en todo similar a la orden PUT CONSOLE OUTPUT TO FILE nombre, se añade la circunstancia de traducir los códigos de control a su forma representable.

PUT PRINTER OUTPUT TO FILE nombre [FILTER]

Como se observa en el ejemplo, también es posible poner en ac-

```
A>
A>put printer output to file m:nombre ;system,no echo,filter¿
WARNING:
File already exists; Delete it (Y/N)? y
Putting list output to file: M:NOMBRE.
A>put printer output to printer
PUT completed for file: M:NOMBRE.
Putting list output to printer
A>
```

13.14.2. El comando PUT aporta una gran variedad de opciones.

ción la opción FILTER con la salida desde la impresora hacia el fichero NOMBRE.

PUT CONSOLE OUTPUT TO CONSOLE

Consigue que se remita nuevamente a la consola la información que habitualmente se enviaba a ésta.

PUT PRINTER OUTPUT TO PRINTER

Análogamente, los datos enviados a la impresora se ven reflejados nuevamente en ella.

A>PUT CONSOLE OUTPUT TO FILE nombre [FILTER,NO ECHO]

Como podemos comprobar, también es posible emplear varias opciones a un tiempo, en este caso FILTER y NO ECHO gracias a las cuales la salida de consola, enviada al fichero nombre, ve filtrados los códigos de control y no se refleja en la pantalla, como sucedería de no haberse empleado el parámetro NO ECHO.

PROFILE

studiaremos ahora uno de los ficheros de mayor importancia, que usualmente está presente en los discos de trabajo, aunque no se trata ni de un comando ni de un programa de utilidad. Realmente, es un fichero que contiene una serie de órdenes que son ejecutadas en el momento de cargar el CP/M Plus: PROFILE.SUB.

Los ficheros PROFILE contienen habitualmente una serie de órdenes, cuyo contenido podremos visualizar fácilmente (por ejemplo, con TYPE), que son ejecutadas para configurar el sistema de una determinada forma; digamos, para darle un cierto «perfil» al sistema, que tal es su traducción literal del inglés.

En los discos suministrados con nuestro equipo, tendremos la oportunidad de constatar la distribución típica de un fichero de este tipo, gracias al PROFILE.ENG presente en los mismos. Se trata del fichero que otorga al ordenador un «perfil» inglés *(ENGland,* Inglaterra). En concreto, en el caso de los CPC, su misión es adaptar las teclas de edición mediante un comando SETKEYS, cuyo efecto comprobaremos en un capítulo posterior y su fichero auxiliar KEYS.CCP,

así como pasar el juego de caracteres al británico (LANGUAGE 3).

En realidad, la utilidad de este PROFILE, máxime para un español, es bastante dudosa. No obstante, nosotros podemos crear nuestro propio perfil, adaptóndonos a nuestras circunstancias particulares. Así, por ejemplo, la primera línea puede seguir siendo SETKEYS KEYS.CCP, dado que la adaptación de las teclas de edición es siempre necesaria, aunque a todas luces lo conveniente será aplicar un LANGUAGE 7, para disponer del juego de caracteres español.

Por otra parte, cuando en un capítulo posterior veamos la orden SETKEYS podremos personalizar aún más nuestro PROFILE, utilizando a tal fin no el fichero auxiliar KEYS.CCP, sino quizá otro creado por nosotros mismos, que redefina en alguna tecla determinada un carácter u orden que empleemos con gran asiduidad.

ACTIVACIÓN DE UN PROFILE

Automáticamente, nada más cargar CP/M Plus, el sistema busca un fichero denominado PROFILE.SUB, cuyo contenido es el perfil que el usuario desea emplear en esa sesión de trabajo. Si lo encuentra, ejecuta todas las órdenes contenidas en el mismo. De ahí que el perfil inglés nunca llegue a ejecutarse en nuestro ordenador, puesto que se llama PROFILE.ENG y no .SUB.

Así, pues, en el supuesto de que nuestra anglofilia nos lleve a utilizar este perfil, deberíamos efectuar una redenominación del fichero, por ejemplo mediante:

A>REN PROFILE.SUB=PROFILE.ENG.

De esta forma, al cargar el sistema operativo, automáticamente se ejecutarán las órdenes contenidas en PROFILE.SUB (antes .ENG), con las consecuencias a que anteriormente hemos hecho mención:

SETKEYS KEYS.CCP
LANGUAGE 3

```
A>TYPE PROFILE.ENG
setkeys keys.ccp
language 3
```

13.15.1. TYPE del fichero PROFILE.ENG en los CPC.

```
A>type profile.eng
setdef m:,* ¡order = (sub,com) temporary = m:¿
pip
<m:=basic.com¡o¿
<m:=dir.com¡o¿
<m:=erase.com¡o¿
<m:=paper.com¡o¿
<m:=pip.com¡o¿
<m:=rename.com¡o¿
<m:=show.com¡o¿
<m:=submit.com¡o¿
<m:=type.com¡o¿
<
```

13.15.2. TYPE del fichero PROFILE.ENG en los PCW.

Gracias a esto se pueden utilizar las teclas de edición de forma similar a como se hace en BASIC y se obtiene el juego de caracteres británicos en lugar del estadounidense (libra por dólar).

Llegados a este punto, es muy importante hacer mención a que durante la ejecución de cualquier fichero .SUB (veamos en capítulos posteriores la orden SUBMIT) y, por tanto, también de PROFILE .SUB, se crea en el disco un fichero temporal, que es borrado al finalizar la ejecución del perfil. No obstante, para posibilitar la generación de dicho fichero puente es necesario que el disco no se encuentre protegido contra escritura.

De todo esto se desprende que nosotros podemos generar discos de trabajo en los cuales grabar (mediante PIP) el sistema operativo y un perfil adecuado para la tarea concreta a que van a ser destinados.

Así, por ejemplo, a los usuarios de PCW, cuyo intérprete de BASIC se carga desde disco, les puede ser de gran utilidad una copia de trabajo, que además del sistema operativo, imprescindible para el funcionamiento del equipo, incluya un perfil con una adaptación del teclado de función a determinados comandos BASIC de uso frecuente, paso al LANGUAGE 0 (EE.UU.), para no tener problemas con almohadas (#) y la carga del propio intérprete.

Asimismo, podemos disponer de otra copia de trabajo, supongamos para ejecutar un determinado procesador de textos, que antes de la carga del mismo adapte las teclas y el lenguaje al castellano. O bien, una para operar con CP/M, cuyo perfil consista en la copia en memoria virtual de los comandos más comúnmente empleados.

CREACIÓN DE UNA COPIA DE TRABAJO

La creación de esta copia de trabajo es bien simple. En principio, necesitaremos tener el sistema operativo, que es el fichero con distintivo .EMS que se encuentra en los discos del sistema y, obviamente, todos aquellos comandos o programas que consideremos de utilidad.

Para ello disponemos de la orden PIP, que ya fue comentada ampliamente en el primer volumen dedicado a CP/M en esta colección. No obstante, siempre es posible copiar el disco original a través de DISCKIT (o DISCKIT3) y a continuación borrar (ERA) los ficheros sobrantes.

El siguiente paso será la creación de nuestro fichero perfil PROFILE.SUB, que podremos generar a partir de cualquier editor de textos, como puede ser el propio ED.COM de CP/M, de cuyo manejo también tuvimos conocimiento en el volumen 2 de esta biblioteca.

También será posible, desde el propio BASIC, crearlo en forma de fichero secuencial. Por ejemplo:

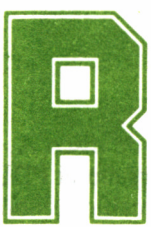
```
10 OPENOUT "PROFILE.SUB"
20 PRINT #9, "SETKEYS KEYS.CCP"
30 PRINT #9, "LANGUAGE 7"
40 CLOSEOUT
```

En cuanto a su contenido, basta con incluir todos los comandos necesarios para nuestro perfil, en forma de órdenes correctas CP/M.

```
M>
M>dir
M: BASIC    COM : DIR      COM : ERASE    COM : KEYS     WP  : PIP      COM
M: PROFILE  ENG : RENAME   COM : SETKEYS  COM
M>ren profile.sub=profile.eng
M>dir
M: BASIC    COM : DIR      COM : ERASE    COM : KEYS     WP  : PIP      COM
M: PROFILE  SUB : RENAME   COM : SETKEYS  COM
M>
```

13.15.3. Simulación de un disco de arranque a partir de PROFILE.SUB.

RMAC



MAC es el ensamblador de macros reubicable de CP/M Plus. La fuente siempre es un programa en ensamblador, con distintivo .ASM, y la salida un código objeto reubicable en forma de fichero .REL, del cual puede obtenerse a su vez un fichero ejecutable como comando, con extensión .COM. Por otra parte, se generan los correspondientes .PRN, para la salida por pantalla o impresora y .SYM, con la lista ordenada de símbolos definidos por el programa.

```
A>RMAC
CP/M RMAC ASSEM 1.1
A:.ASM-NO SOURCE FILE PRESENT

A>
```

13.16.1. RMAC intenta, en primera instancia, la localización en la unidad por defecto de algún fichero .ASM.

La sintaxis general de RMAC es la siguiente:

RMAC nombre-fichero {$Rd | $Sd | $Pd}

Por otra parte, las posibles opciones son las descritas a continuación:

OPCION	Unidad de disco para salida
R- Unidad de disco para fichero REL	(A-O,Z).
S- Unidad de disco para fichero SYM	(A-O,X,P,Z).
P- Unidad de disco para fichero PRN	(A-O,X,P,Z).

donde:

A-O	Especifica la unidad de disco de A a O.
X	Significa salida a la consola o pantalla.
P	Significa salida a la impresora.
Z	Significa sin salida.

Así, por ejemplo:

A>RMAC PRUEBA $PX SB RB

Ensambla el fichero PRUEBA.ASM en la unidad de disco A, pasa el fichero PRUEBA.PRN listándolo a la consola, graba el PRUEBA.SYM en la unidad B y el código objeto reubicable generado (PRUEBA.REL) igualmente en la unidad B.

SAVE

sta orden ejecuta el programa especificado, grabando a continuación el contenido de la memoria, entre dos direcciones hexadecimales dadas en un fichero cuyo nombre también se indica. Tras todo esto se retorna al sistema.

Así, pues, este comando precisa los nombres de los ficheros a ejecutar y grabar, así como las direcciones de memoria que delimitan el área a copiar (en hexadecimal).

La orden se ejecuta escribiendo SAVE. Por ejemplo:

A>SAVE

A>PRUEBA

... salida de PRUEBA

File (or RETURN to Exit)? NUEVOFIC

from?100

to?400

Hace que el contenido de la memoria, entre las direcciones &100 y &400, tras el fin de ejecución de PRUEBA, se grabe en el fichero NUEVOFIC de la unidad de disco por defecto.

```
A>
A>save

CP/M 3 SAVE - Version 3.0
Enter file (type RETURN to exit): palette.com
Delete palette.com? n
Enter file (type RETURN to exit):
CP/M 3 SAVE - Version 3.0
```

13.17.1. El comando SAVE interroga sobre el programa a cargar.

Veamos ahora otro ejemplo, siguiéndolo paso por paso:

A>SAVE

Activa la utilidad SAVE. Ahora se introduce el nombre del programa a cargar y ejecutar desde la memoria.

A>PROG.COM

Al finalizar la ejecución de éste, SAVE intercepta el regreso al sistema y se desvía hacia sí mismo.

SAVE Ver. 3.0

Enter file (type RETURN to exit):VOLCADO

Se indica el nombre del fichero a grabar. Entonces el sistema pregunta por los límites del área de memoria a grabar.

Beginning hex address:100

Ending hex address:500

El contenido de memoria desde 100H (HEXADECIMAL) a 500H es copiado al fichero NUEVOFIC.

SETDEF

Existen determinadas actuaciones del sistema que éste toma por defecto como, por ejemplo, operar sobre la unidad implícita, siempre y cuando no se especifique lo contrario. Para variar este tipo de comportamiento disponemos de la orden SETDEF, con las diferentes opciones que a continuación estudiaremos.

PRIORIDAD DE DISCO

Cuando el comando va seguido de dos especificadores separados por una coma indican el orden de preferencia, por defecto, en el acceso a las unidades de disco. Así, por ejemplo, si tras

SETDEF B:,A:

se intenta localizar un fichero, en primer lugar se accederá a la segunda unidad de disco (B:) y de no encontrarlo se buscará entonces en la A:.

En este tipo de orden, se puede emplear el carácter asterisco (*) para señalar la unidad implícita, sin acompañarlo del signo dos puntos (:). Veamos otro ejemplo al respecto:

SETDEF *,A:

En este caso, en primer lugar se intentará localizar el fichero en la unidad de disco implícita, y de no tener éxito en la búsqueda se procederá a su localización en B:

Análogamente, una orden del tipo:

SETDEF B:,*

Produciría una búsqueda, en primer término, en la unidad B:, para pasar a continuación, de no haberse encontrado el fichero en ella, a su localización en la implícita.

Como podemos comprobar, esta orden puede ser en algunas aplicaciones concretas de gran utilidad, debido a lo cual es frecuente su inclusión en ficheros tipo PROFILE.SUB, para su establecimiento nada más arrancar el disco de trabajo.

Por supuesto, no hay porqué hacer mención siempre a la unidad implícita. Es muy posible que un usuario del PCW vuelque mediante su PROFILE.SUB las órdenes más importantes en M:, como pueden ser, por ejemplo, DIR o PIP, así como la aplicación a utilizar, dejando los ficheros para la misma en la unidad A:. En este caso sería de interés un comando:

SETDEF M:,A:

para no tener problemas en la gestión de los ficheros.

Por otra parte, no es imprescindible especificar dos unidades de disco, sino que es posible indicar una sola:

SETDEF M:

Para completar el efecto de esta orden sobre el disco se encuentra su opción TEMPORARY, que señala la unidad a emplear por defecto para el almacenamiento de los ficheros temporales, que como ya sabemos, debe generar CP/M en algunas ocasiones para su correcto funcionamiento.

SETDEF [TEMPORARY= UNIDAD]

Prioridad En La Búsqueda De Ordenes

Otra de las acciones tomadas por defecto por CP/M es la búsqueda de ficheros .COM al indicarse un nombre tras el *prompt* (A>).

Gracias a la opción ORDER de SETDEF se puede alterar la prioridad en la búsqueda de órdenes, de forma que se intente localizar un fichero .SUB antes que el .COM. Su sintaxis es:

SETDEF [(ORDER)=(SUB,COM)]

La otra forma de utilización de esta orden es:

SETDEF [(ORDEN)=(COM,SUB)]

Esta, por el contrario, selecciona el orden de prioridad .COM y .SUB.

PAGINACIÓN DE PANTALLA

Por otra parte, la orden SETDEF se puede acompañar de una opción que active o inhíba la paginación de pantalla. El término paginación implica que el proceso de escritura en la pantalla se detiene cuando ésta se llena, en espera de la pulsación de una tecla por parte del usuario, para que no se pierda parte de la información vertida en la misma sin que éste la haya leído.

```
A>setdef m:,a:

Drive Search Path:
1st Drive              - M:
2nd Drive              - A:


A>language 0

A>setdef ¡order=(com,sub)¿

Search Order           - COM, SUB

A>setdef ¡temporary=b:¿

Temporary Drive        - B:

A>setdef ¡page¿

Console Page Mode      - On

A>
```

13.18.1. Las opciones de SETDEF son muy diversas.

Su sintaxis es, por tanto:

SET DEF [PAGE]

para la activación del modo de paginación y

SET DEF [NO PAGE]

para su desconexión.

Finalmente, este comando puede utilizarse sin opción alguna, en cuyo caso su efecto es informarnos sobre las actuales prioridades de búsqueda en los discos, unidad empleada para los ficheros temporales y orden de extensiones para ejecución (.SUB o .COM).

EN RESUMEN

Veamos un resumen de las dispares tareas asignadas a la orden que nos ocupa:

SETDEF	Muestra las prioridades actuales de búsqueda en los discos, unidad empleada para guardar ficheros temporales y localización de extensiones para ejecución.
SETDEF unidad: [unidad:]	Establece el orden de búsqueda en las unidades de disco. Para indicar la unidad implícita se utiliza el símbolo *, sin acompañarlo de los habituales dos puntos (:).
SETDEF [TEMPORARY= unidad:]	Establece la unidad que se va a utilizar para el almacén de los ficheros temporales.
SETDEF [ORDER= (COM,SUB)]	Especifica la prioridad de búsqueda para ejecución en ficheros .COM.
SETDEF [ORDER= (SUB,COM)]	Especifica la prioridad de búsqueda para ejecución en ficheros .SUB.
SETDEF [PAGE]	Establece el modo de paginación de la pantalla.
SETDEF [NO PAGE]	Desconecta el modo de paginación de la pantalla.

SETKEYS

Sólo hay un programa capaz de cambiar la estructura prefijada del teclado, al menos en CP/M. Se trata de SETKEYS. Este interpreta los códigos almacenados en un fichero de texto, y asigna nuevos caracteres a las teclas. Para ejecutarlo hay que escribir:

A>SETKEYS NOMBRE

donde NOMBRE es un fichero que contiene los datos para reconfigurar el teclado.

¿Dónde puede ser útil la ayuda de SETKEYS? Siempre que se necesiten ciertos caracteres no accesibles por teclado directamente, como las vocales acentuadas, o quizá el alfabeto griego. También resulta muy cómodo para la ejecución de programas, desde el mismo BASIC hasta otros programas comerciales, donde puede pedirse con frecuencia una pulsación incómoda. Y no digamos nada de sustituir las secuencias de teclas por una sola pulsación:

¿Cuántas veces se pulsa «list» en una sesión de BASIC?

Para escribir el fichero de definición se necesita un editor; en este caso, el más cómodo es ED, dado que se trata de un texto muy corto. El manejo de este editor resulta algo complicado, pero se trata ampliamente en el segundo volumen de esta colección.

CÓMO DEFINIR LAS TECLAS

SETKEYS ofrece tres tipos de resultados para la pulsación de una tecla: generación de un carácter, de un código de control o de una cadena de expansión (conjunto de caracteres y códigos).

Para obtener un carácter la línea ha de ser así:

número-de-tecla estado "carácter" [comentario]

«número-de-tecla» es el código correspondiente a la tecla que deseamos redefinir (en el manual hay un esquema completo); estado indica las teclas auxiliares que han de pulsarse junto con la principal para que se produzca el resultado definido; éstas son *MAYS, EXTRA, ALT* (CONTROL en los CPC) o *MAYS+ALT* (MAYS+CONTROL en los CPC). Los códigos de estado son S, E, A y SA, respectivamente.

En "carácter" debemos entrecomillar el carácter que deseamos como resultado de la definición, evidentemente. El comentario se puede omitir y tiene tan sólo la finalidad de indicar al usuario la explicación correspondiente (recomendamos los comentarios en caso de que se vayan a efectuar modificaciones en adelante).

EJEMPLOS

Pondremos varios ejemplos que, aunque válidos directamente para los PCW, orientarán igualmente a los usuarios de los CPC. En el improbable caso de querer que la tecla *INS* genere una "a" debemos escribir en ED lo siguiente:

```
A>ed tecla.ins
*i
3 N "a" Ejemplo de a en INS
(SAL)
*e
A>setkeys tecla.ins
```

El 3 es el número de la tecla *INS,* "N" indica el estado "normal", en el que no se pulsa ninguna otra tecla a la vez. El resto ya lo podemos suponer.

Si deseamos que la "a" se genere sólo con ALT+INS o EXTRA+INS, pondríamos *3 A E* "a" (el comentario nos lo podemos ahorrar).

Para mencionar un carácter que en ese momento no genera el teclado, hay que acudir a su código (ASCII). Supongamos la "a"

acentuada, que queremos obtener con EXTRA+a. El número de la tecla "a" es 69, el código de la "a" acentuada es 224. La línea de definición sería:

69 E " ↑224' "

Así mismo, se puede indicar el código del carácter generado en hexadecimal: *69 E " ↑#EO'* "el signo " ↑" se obtiene con EXTRA+ú).

Si deseamos obtener un código de control, éste se escribe entrecomillado y «tal cual», precedido del signo " ↑ " como control; si, por ejemplo, queremos que *CAN* genere control+s (interrupción temporal de ejecución), escribiremos:

75 N " ↑S"

Por último, podemos hacer que una sola tecla genere una secuencia de caracteres y códigos de control. Para ello, hay que definir previamente un «código expansible», el cual asignaremos después a una tecla. Los códigos expansibles tienen los números 128 a 159. Los primeros ya están definidos, por lo que conviene tomar los últimos. Asignemos, por ejemplo, la secuencia "list" a la tecla *f1*. El fichero de definición elaborado con *ED* contendría estas líneas:

E 159 "list" código 159 contiene la cadena "list"
2 N ↑'159' se asigna el código 159 a la tecla f1

```
A>TYPE KEYS.CCP
0 N S C "^'R1F'" CCP cursor up
1 N S "^F" cursor right
1 C "^'R9F'"
2 N S C "^'R1E'" cursor down
8 N S "^A" cursor left
8 C "^'R9E'"
9 N S C "^W" copy
16 N S "^G" clr
16 C "^K"
18 C "^E" enter
66 N S "^'27'" esc
66 C "^C"
79 C "^X" del
E R8C "^R" ctrl enter
E R9E "^F^B"
E R9F "^F^B^B"
```

13.19.1. Para el comando SETKEYS es necesario emplear un fichero de apoyo con las definiciones de teclas apropiadas.

También pueden incluirse en la cadena códigos de control. Por ejemplo, podemos añadir a "list" el código CONTROL+M, que equivale a *RETURN,* con lo cual, al pulsar f1, se ejecutará automáticamente un "list".

```
E 159 "list↑M" código 159 incluye un return
2 N ↑'159' y eso se asigna a f1
```

Teniendo en cuenta todas estas posibilidades, podemos aprovechar al completo las teclas de función (efes), utilizando conjuntamente *MAYS, ALT* y *EXTRA.* Así, de la tecla f1 podemos obtener 5 resultados: pulsación normal, con MAYS, con ALT, con EXTRA y con MAYS+ALT.

REDEFINICIÓN AUTOMÁTICA

El proceso de redefinición se reduce, una vez indicados los datos para ésta, a escribir en CP/M la instrucción *A>setkeys fichero.* Esto parece simple y poco molesto, o no; según se mire. Resulta realmente cómodo encender el ordenador, meter el disco y empezar a manejar el programa elegido. La ejecución de SETKEYS se puede incluir en este proceso automático de cargas.

Al cargarse CP/M se buscan algunos ficheros para, si existen, aprovecharlos antes de dar el control al usuario. Estos ficheros son *SUBMIT.COM* y *PROFILE.SUB.* El primero ejecuta todas las órdenes de CP/M que encuentre escritas en el segundo.

Imaginemos ahora que tenemos un fichero con una definición de las teclas f1-f8 para trabajar con BASIC. Para conseguir el proceso automático de carga con un disco, debemos incluir, cómo no, el fichero de CP/M (con distintivo *EMS),* los ficheros *SUBMIT, SETKEYS,* y en el que se encuentren esos datos.

Ahora debemos coger de nuevo el *ED,* y escribir un texto llamado *PROFILE.SUB* con estas líneas:

```
setkeys teclas.def
basic
```

donde suponemos que "teclas.def" es el nombre del fichero de definición.

A partir de este momento, cuando vayamos a trabajar en BASIC, encendemos el ordenador y metemos el disco previamente preparado. Se cargará el CP/M, se detectará a continuación el fichero *PROFILE* y se ejecutará línea a línea lo que en él se indique (o sea: definir teclado y cargar BASIC).

Un detalle importante es que el disco así preparado NO puede estar protegido contra escritura, puesto que *SUBMIT* crea sobre él un fichero temporal (llamado SYSINXXX.$$$, donde XXX es un indicador interno). Este se borra al terminar la ejecución del contenido de PROFILE.

En el caso de que *SUBMIT* encuentre el disco protegido provoca el error correspondiente, con lo que se nos permite desprotegerlo, o "Cancelar". Esto último detiene la ejecución automática del proceso, y también dos pulsaciones de la tecla *STOP*.

Como última recomendación diremos que conviene tener otro fichero de definición para restablecer las características anteriores del teclado, y así evitar complicaciones al utilizar otros programas que reconocen la configuración estándar. En tal caso, nuestro *PROFILE* quedaría así:

```
setkeys nueva.def
basic
setkeys antigua.def
```

y al terminar el trabajo en BASIC tecleando *system, SUBMIT* devolvería el teclado a su estado anterior.

SETLST Y SET 24X80

La inicialización de impresoras se puede conseguir con la orden:

SETLST <nombre>

donde <nombre> es un fichero que contiene la serie de códigos que se debe enviar a la impresora, de modo similar a como los ficheros SETKEYS albergan los de redefinición del teclado. Siendo su representación:

↑ <carácter>

↑ "<valor del carácter>"

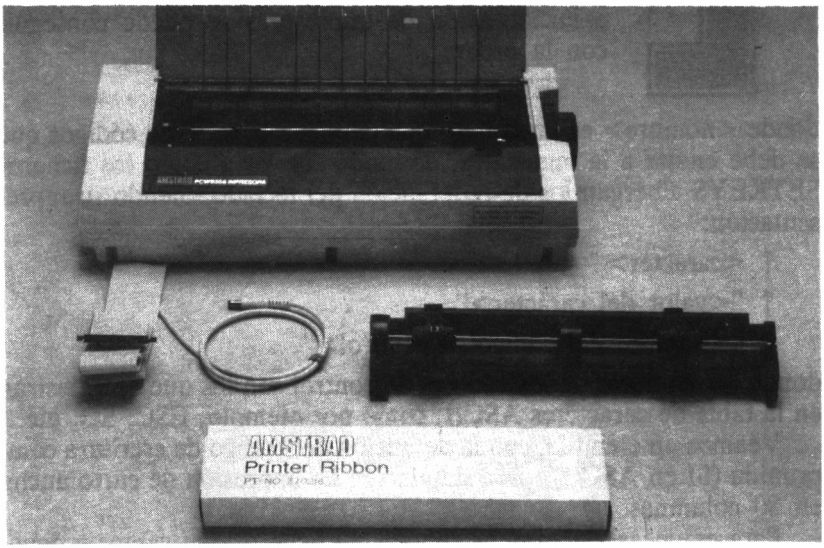
↑ "<nombre del código de control>"

donde los nombres de los códigos de control son los que se muestran en la tabla de caracteres ASCII, como por ejemplo, ESC, SO, etc...

Veamos un ejemplo, con la definición del código de escritura comprimida (SI en ASCII), para simulación de impresión de carro ancho en 80 columnas.

Para empezar, deberemos generar un fichero, que vamos a denominar IMPR, y podremos crearlo con cualquier editor de texto, como

13.20.1. La pantalla es un dispositivo cuyo formato se puede controlar gracias a SET 24X80.



13.20.2. SETLST es el comando que permite fijar el formato de la impresora.

puede ser el propio ED de CP/M. Su contenido será el siguiente:

↑ "SI"

↑ O

↑ "&F"

o también...

↑ "15"

Dado que cualquiera de ellas representan el carácter 15 de escritura comprimida.

FORMATEADO DE LA PANTALLA

Al igual que con el carácter de escritura comprimida podemos simular impresoras de carro ancho sobre 80 columnas, es posible que algunas aplicaciones vengan preparadas para la utilización de la pantalla en el formato de 24 filas por 80 columnas. Aunque esta configuración se adopta por defecto en los CPC, en los PCW se inicia el trabajo con otra bastante más extraña (90 columnas).

Si deseamos adoptar el formato 24 × 80 deberemos escribir:

SET 24X80

o bien

SET 24X80 ON

Mientras que la orden para anular su efecto será:

SET 24X80 Off

SETSIO

Como otro dispositivo más, al igual que el disco, la pantalla o la impresora, CP/M puede controlar un interface serie monocanal de entrada/salida (RS232). La orden SETSIO se implementa a tal fin, pudiendo establecer los distintos parámetros determinantes de la actuación de este tipo de interface.

La forma más sencilla de utilización de este comando es en solitario, sin ningún tipo de parámetros, gracias a lo cual el sistema nos informa de la configuración actual de la salida RS-232:

```
A>SETSIO
```

PARÁMETROS DE SETSIO

Por otra parte, el comando puede ejercer como establecedor de formato, pudiendo actuar sobre los siguientes parámetros, por el orden en que a continuación se relacionan:

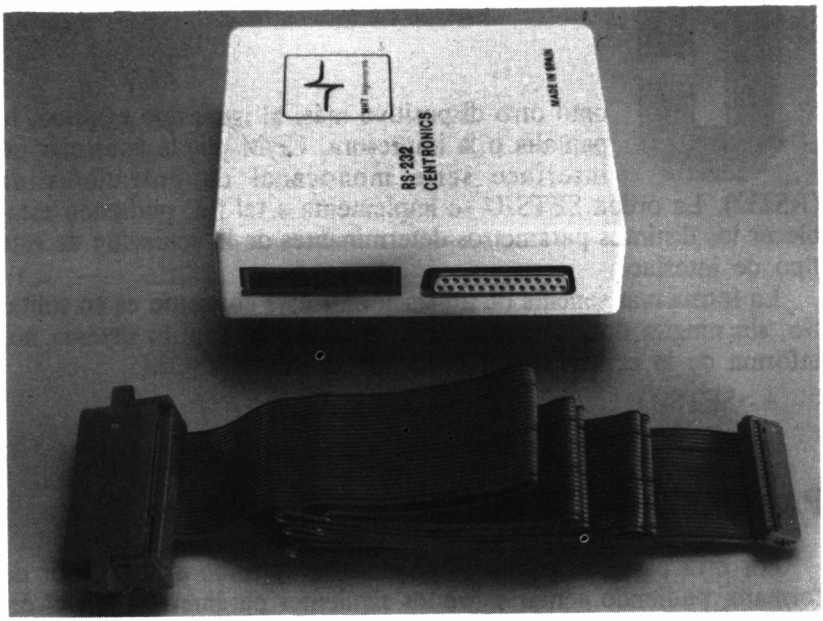
— Velocidad de transmisión.
— Velocidad de recepción.
— Paridad.
— Bit de stop.
— Transmisión de 8 bits.
— Handshake.
— Protocolo XON/XOFF.

Así, por ejemplo, un comando SETSIO podría adoptar la siguiente forma:

SETSIO,Rx1200,Tx75,PARITY NONE,STOP 1,BITS 8,HANDSHAKE=ON, XOFF=OFF

Por otra parte, esta orden se halla íntimamente relacionada con DEVICE, pues algunas de las características de velocidad de transmisión (RX) y de estado XON/XOFF pueden resultar afectadas por esta última.

Como ya tuvimos ocasión de comprobar, DEVICE controla los dispositivos lógicos y físicos, de forma que es posible actuar sobre la velocidad de transmisión, como por ejemplo:



13.21.1. El canal serie (RS-232) se controla gracias a la orden SETSIO.

DEVICE SIO [1200]

o bien sobre el protocolo XON/XOFF:

DEVICE SIO [XON]

o con

DEVICE SIO [NOXON]

En cuanto a lo referente a los dispositivos físicos, es posible asignar alguno de ellos al interface serie, cuyo identificador es SIO. Como, por ejemplo, al auxiliar:

DEVICE AUXOUT=SIO

o bien a la impresora:

DEVICE LST=SIO

Con lo que la salida que normalmente se bifurcaría a la impresora pasaría ahora a través del dispositivo serie.

SID

ID es el depurador de instrucciones simbólicas de CP/M Plus: *Symbolic Instructions Debugger*. Su función es depurar programas en código máquina, para lo cual los va ejecutando paso a paso, mostrándonos en todo momento las transformaciones que se producen en la memoria, por defecto del desarrollo de la rutina, así como el contenido de los registros de la CPU, permitiéndonos intercalar puntos de interrupción en el programa.

En relación directa con esta orden, se encuentran las utilidades TRACE.UTL e HIST.ULT, proporcionadas en los discos del sistema, cuyo cometido es, respectivamente, seguir la pista, en orden inverso de ejecución, de las instrucciones que han conducido a un punto de interrupción y dar un histograma (diagrama de barras) de la frecuencia con que se ha accedido a los códigos en determinados sectores del programa.

COMANDOS DE SID

A continuación se relacionan los comandos del SID, con su correspondiente significación:

COMANDO:		SIGNIFICADO:
As	(Assemble)	Introduce órdenes en lenguaje ensamblador. — s es la dirección de inicio.
Cs{b{,d}}	(Call)	Hace una llamada desde SID a una dirección de memoria. — s es la dirección de llamada. — b es el valor del par BC. — d es el valor del par DE.
D{W}{s}{,f}	(display)	Muestra la memoria en hexadecimal y ASCII. — W es una palabra con formato de 16 bit. — s es la dirección de inicio. — f es la dirección de final.
Epgm-fichero {,sim-fichero}	(load)	Carga para ejecución de un programa y su tabla de símbolos.
E * sim-fichero	(Load)	Carga una tabla de símbolos.
Fs,f,d	(Fill)	Llena la memoria con un valor constante. — s es la dirección de comienzo. — f es la dirección de final. — d es un dato de bits.
G{p}{,a{,b}}	(Go)	Comienzo de ejecución. — p es la dirección de comienzo. — a es un punto de ruptura temporal.
H	(Hex)	Muestra todos los símbolos con las direcciones en hexadecimal.
H.a		Muestra los valores hexadecimal, decimal y ASCII de a; donde a es una expresión simbólica.
Ha,b		Calcula la suma y diferencia hexadecimal de a y b; donde a y b son expresiones simbólicas.
I command tail	(Input)	Introduce una línea de comando CCP.
L{s}{,f}	(List)	Lista instrucciones mnemónicas del 8080.

— s dirección de inicio.
— f dirección de final.

Ms,h,d (Move) Desplaza un bloque de memoria.
— s es la dirección de inicio.
— h es la dirección final del bloque.
— d es la dirección de destino del bloque.

P{p{,c}} (Pass) Establece, inicializa y muestra el puntero de programa.
— p es la dirección de un punto de ruptura permanente.
— c es el valor inicial del puntero.

Rfichero {,d} (Read) Lee código/símbolos.
— d es un desplazamiento a cualquier dirección.

S{W}s (Set) Fija valores de memoria.
— s es la dirección a la cual se envía el valor.
— W es una palabra de 16 bit.

T{n{,c}} (Trace) Tracea la ejecución del programa.
— n es el número de pasos del programa.
— c es la dirección de entrada de la utilidad.

T{W}{n{,c}} (Trace) Traceo sin llamada.
— W indica a SID no tracear subrutinas.
— n número de pasos del programa.
— c es la dirección de entrada de la utilidad.

U{W}{n{,c}} (Untrace) Ejecución desde monitor sin traceo.
— n es el número de pasos del programa.
— c es el punto de entrada de la utilidad.
— W indica a SID no tracear subrutinas.

V (Value) Muestra el valor de la próxima dirección de memoria disponible (NEXT), la siguiente tras el fichero leído más

largo (MSZE), el valor actual del contador de programa (PC), y la dirección de final de la memoria disponible (END).

Wfichero,s,f	(Write)	Escribe el contenido de un bloque de memoria contiguo a fichero. — f es la dirección de final.
x{f}{r}	(Examine)	Examina/altera el estado de CPU. — f es el bit de bandera C, Z, M, E o I. — r es el registro A, B, D, H, S o P.

EJEMPLOS

A>SID

CP/M Plus carga SID desde drive A en la memoria. SID muestra el *prompt* # cuando está leyendo los comandos aceptados.

A>B:SID PROGRAMA.HEX

CP/M Plus carga SID y el fichero programa PROGRAMA.HEX en la memoria desde la unidad de disco B.

```
A>sid
CP/M 3 SID - Version 3.0
#
#^C
A>
```

13.22.1. Comienzo de ejecución en el SID de CP/M.

SUBMIT

omo ya vimos al tratar la ejecución de PROFILE-.SUB, es posible ejecutar las líneas contenidas en un determinado fichero, como si se tratasen de órdenes y datos para CP/M introducidos directamente desde el teclado.

Este sistema se revela de gran utilidad al confeccionar un disco de trabajo; no obstante, CP/M pone a nuestro alcance la posibilidad de ejecutar ficheros, de forma similar al PROFILE.SUB, en cualquier momento, sin necesidad que sea al arrancar la copia de trabajo.

Esta misión corre a cargo del comando SUBMIT, que se puede abreviar a SUB, cuya función es la ejecución del fichero de texto que se le indique, considerando que cada línea de éste va a ser entendida como procedente del teclado y teniendo siempre en cuenta que dicho fichero, obligatoriamente, debe tener la extensión .SUB.

Así, pues, podríamos afirmar que la ejecución de PROFILE.SUB durante el arranque del sistema, se trata de una particularización del caso general de SUBMIT; es como si CP/M, en el momento de su inicio, ejecutara una orden del tipo:

SUBMIT PROFILE.SUB

Es muy importante dejar desprotegido contra escritura el disco en el cual se encuentra el fichero .SUB, dado que este comando necesita

crear un fichero temporal que graba en la unidad de disco, borrándolo automáticamente al finalizar la ejecución de las órdenes del fichero.

FICHEROS .SUB

El fichero de tipo .SUB puede contener tanto órdenes de CP/M Plus, como más comandos SUBMIT anidados y datos de entrada para las órdenes de CP/M utilizadas o para los propios programas.

Por otra parte, existe la restricción de que las líneas de órdenes no pueden superar los 128 caracteres y las líneas de datos de entrada tienen que ir precedidas del signo menor que (<). Además, obviamente, es absolutamente imprescindible que el fichero grabado tenga la extensión .SUB.

Además, si un programa ejecutado por una orden SUBMIT finaliza sin consumir todos los datos de entrada asociados, se emite el siguiente mensaje:

Warning: Program input ignored

cuya traducción es: Atención: entrada para programa ignorada; procediéndose a continuación con el resto de las órdenes del fichero. Por el contrario, si no se han incluido los datos suficientes, SUBMIT espera su entrada por el teclado.

Una de las características que proporcionan gran flexibilidad a SUBMIT es la posibilidad de inclusión de variables en sus ficheros. Concretamente, podrán ser nueve datos, que habrán de ser escritos tras la orden SUBMIT. Dentro del fichero, dichos datos se sustituyen por $1 ... $9, lo que provoca que si, por cualquier motivo, es necesario incluir un símbolo $, deberá introducirse para evitar el error de sintaxis como "$$".

```
A>submit
CP/M 3 SUBMIT Version 3.0
Enter File to SUBMIT: profile
ERROR: No 'SUB' File Found
A>
```

13.23.1. SUBMIT sólo admite ficheros con la extensión .SUB.

OPCIONES DE SUBMIT

Es posible emplear la orden SUBMIT de varias formas:

SUBMIT fichero	Produce la ejecución de las órdenes grabadas en el fichero especificado, sin indicar su extensión, pues ésta es siempre .SUB.
SUBMIT fichero entrada1[entradaN]	Se ejecutan las órdenes grabadas en el fichero especificado, dándole los valores de entrada1 a entradaN a las correspondientes variables $1 a $N.
SUBMIT	Espera que se introduzca por el teclado el nombre del fichero cuyas líneas van a ser ejecutadas, así como otras posibles entradas.

Veamos un ejemplo:

A>SUBMIT FICHERO ENTRADA1 ENTRADA2

ejecuta las órdenes grabadas en FICHERO, dando los valores ENTRADA1 y ENTRADA2 a las variables $1 y $2, respectivamente.

XREF

l comando XREF produce un fichero que contiene un resumen, en referencias cruzadas, del uso de variables en un programa de ensamblador a partir de los ficheros PRN y SYM generados por la actuación previa de MAC o RMAC sobre ese programa.

XREF es una utilidad para programadores en ensamblador. Ha sido diseñado para registrar el uso de símbolos en un programa escrito en ensamblador y dejar constancia del lugar del programa en que estos aparecen. Es útil como referencia durante el mantenimiento de un programa y también para el apoyo de rutinas que, a lo largo de la modificación del programa, no se utilizan muy frecuentemente.

XREF también proporciona un índice de referencias cruzadas de las variables utilizadas en un programa. Este comando precisa para sí la existencia de los ficheros: .PRN y .SYM, producidos por MAC o RMAC. Además, aceptará también los ficheros .PRN producidos por el ensamblador M80 de Microsoft y .SYM del LINKER LINK. Los

ficheros .SYM y .PRN deben tener el mismo nombre que el que acompaña o se suministra para XREF. Finalmente, añadir que este comando da como salida un fichero del tipo .XRF.

Así, por ejemplo:

XREF a: NOMBRE	Efectúa las referencias cruzadas de nombre .PRN y nombre .SYM generados por MAC, produciendo como resultado un fichero XREF.
XREF b: NOMBRE $P	Efectúa la referencia cruzada de nombre .PRN y Nombre .SYM al dispositivo de listado.

Un sistema operativo se hace imprescindible para obtener de una forma cómoda y rápida el máximo rendimiento de nuestro ordenador. En el caso concreto de los Amstrad CPC 6128 y PCW ha sido el CP/M Plus el escogido para tal tarea. Nos encontramos ahora ante una profundización en el estudio de las herramientas que éste pone a nuestra disposición, tratando todas aquellas órdenes y utilidades que quedaron por tocar en el segundo volumen de esta Gran Biblioteca Amstrad.

GRAN BIBLIOTECA AMSTRAD

450 ptas.
(incluido IVA)
Precio en Canarias, Ceuta y Melilla: 435 ptas.