

160 Ptas.

Canarias 165 pts.

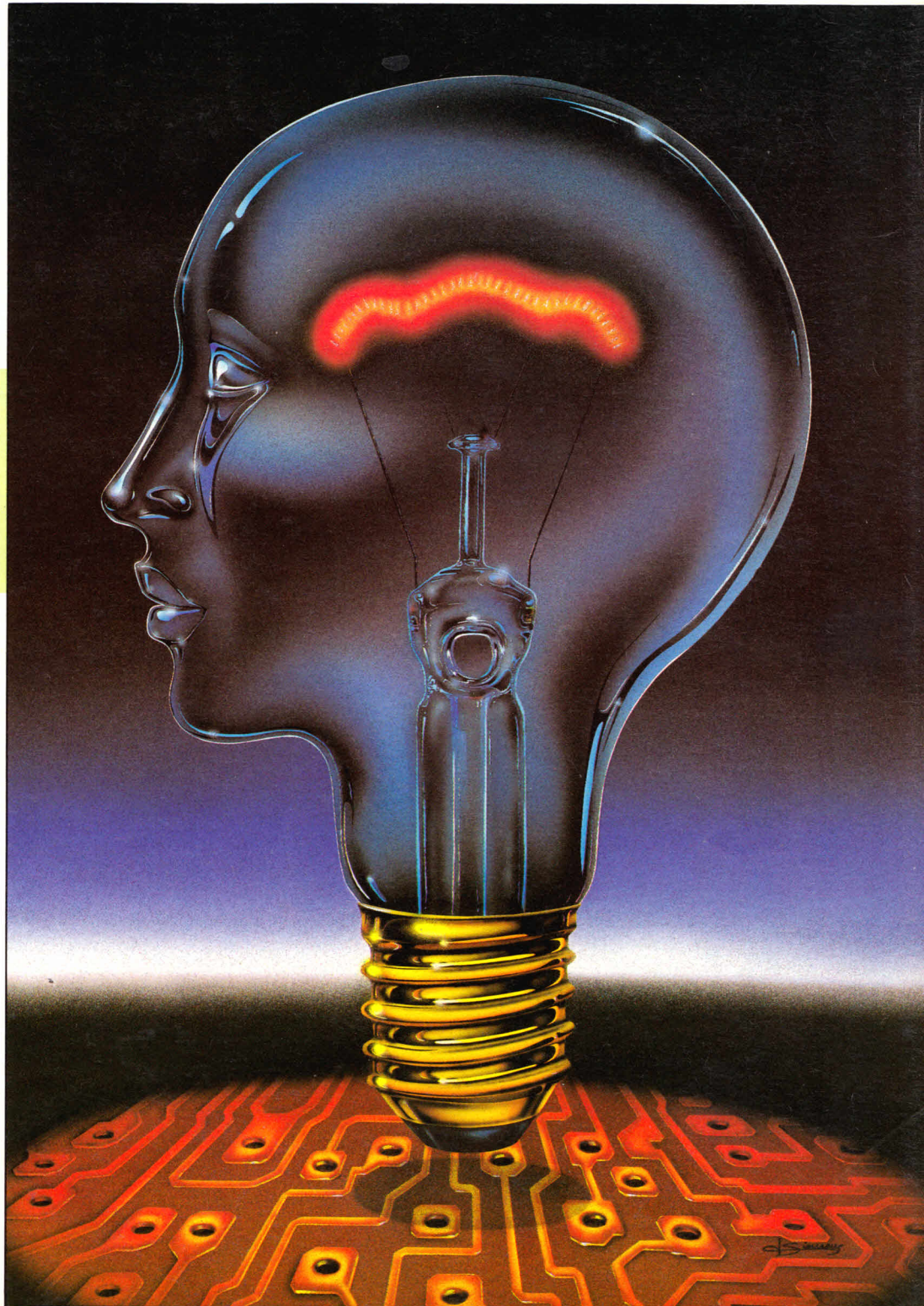
**Así
se organiza
la pantalla**

**Lector
de ROM
en Código
Máquina**

Pon al descubierto todo el contenido de la ROM de tu AMSTRAD, con el lector-copiador que te ofrecemos

**LOAD-RUN:
A LA CARGA
CON LOS
PROGRAMAS**

**MAS
COLORES
PARA TU
AMSTRAD**



SOBREDOSIS DE ACCION

ARMY MOVES



GAME OVER



4 SUPER 4

4 SUPER JUEGOS EN UN ESTUCHE DE DOS CINTAS



ARMY MOVES

SPECTRUM 48 K, +2 • AMSTRAD

DERDHAL es un miembro del C.O.E., Cuerpo de Operaciones Especiales. Ha sido entrenado durante largos años para convertirse en un especialista y ahora es el primero de su promoción.

Puede atravesar las líneas enemigas por tierra, mar o aire, domina todas las técnicas de la guerra en la selva, conoce todas las armas y es un experto en explosivos.

ARMY MOVES, tres sistemas de combate distintos:

- JEEP, equipado con misiles tierra-aire.
- HELICOPTERO COBRA para la lucha en la jungla.
- SOLDADO COE, miembro de un cuerpo de élite, entrenado en todas las técnicas conocidas para la guerra.

GAME OVER

SPECTRUM 48 K, +2 • AMSTRAD

En una lejanísima Galaxia perdida en la inmensidad del Universo, una bellísima y malvada mujer había sometido con su inteligencia y un inmenso ejército de TERMINATORS, a las cinco confederaciones de planetas situadas más allá de ALFA CENTAURI. GAME OVER, dos mundos diferentes, más de 60K de gráficos, adictividad asegurada. GAME OVER, una video-aventura cargada de toda la acción del mejor arcade. GAME OVER, la ilusión de lo desconocido.

4 SUPER 4

SPECTRUM 48 K, +2 • AMSTRAD

4 SUPER 4 es una recopilación de juegos DINAMIC caracterizada por la variedad de programas que incluye. Si deseas sumergirte en el mundo mágico de CAMELOT, si tu curiosidad te obliga a conocer cómo han sido llevados los toros al ordenador, si prefieres vivir la aventura del Viejo Oeste o el reto de SCRIZAM, no lo dudes, 4 SUPER 4 es justo lo que andabas buscando.

DINAMIC

MICROHOBBY

AMSTRAD

Sumario

Director Editorial

José I. Gómez-Centurión

Director Ejecutivo

José M.ª Díaz

Redactor Jefe

Juan José Martínez

Diseño gráfico

José Flores

Colaboradores

Eduardo Ruiz

Javier Barceló

David Sopena

Robert Chatwin

Francisco Portalo

Pedro Sudón

Miguel Sepúlveda

Francisco Martín

Jesús Alonso

Pedro S. Pérez

Amalio Gómez

Alberto Suñer

Secretaría Redacción

Carmen Elías

Fotografía

Carlos Candel

Chema Sacristán

Portada

Pejo

Ilustradores

J. Igual, J. Pons, F. L. Frontán,

J. Septien, Pejo, J. J. Mora

Edita

HOBBY PRESS, S.A.

Presidente

María Andrino

Consejero Delegado

José I. Gómez-Centurión

Jefe de Producción

Carlos Peropadre

Marketing

Marta García

Jefe de Publicidad

Concha Gutiérrez

Secretaría de Dirección

Pilar Arestizábal

Suscripciones

M.ª Rosa González

M.ª del Mar Calzada

Redacción, Administración y Publicidad

Ctra. de Irún km 12,400

(Fuencarral) 28049 Madrid

Pedidos y suscripciones:

734 65 00

Redacción: 734 70 12

Dto. Circulación

Paulino Blanco

Distribución

Coedis, S. A. Valencia, 245

Barcelona

Imprime

ROTEDEC, S. A. Ctra. de Irún.

Km. 12,450 (MADRID)

Fotocomposición

Novocomp, S.A.

Nicolás Morales, 38-40

Fotomecánica

GROF

Ezequiel Solana, 16

Depósito Legal:

M-28468-1985

Derechos exclusivos de la revista

COMPUTING with the AMSTRAD

Representante para Argentina, Chile, Uruguay y Paraguay, Cia.

Americana de Ediciones, S.R.L. Sud América 1.532. Tel.: 21 24 64. 1209 BUENOS AIRES (Argentina).

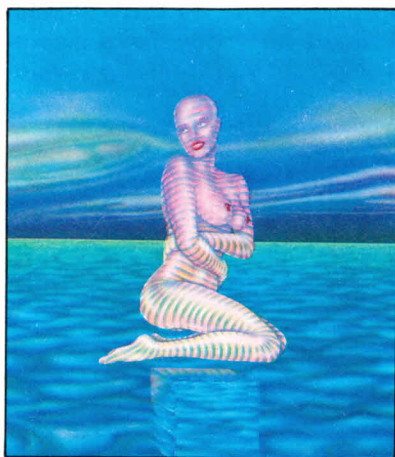
M. H. AMSTRAD no se hace necesariamente solidaria de las opiniones vertidas por sus colaboradores en los artículos firmados. Reservados todos los derechos.

Año II • Número 64 • 2 al 8 de Diciembre
160 ptas. (incluido I.V.A.)
Canarias, 155 ptas. + 10 ptas. sobretasa aérea
Ceuta y Melilla, 155 ptas.

Código máquina



Para muchos, no es extraño dada su dificultad, la pantalla del **Amstrad** continúa siendo un misterio. En la primera parte de este artículo de nuestro curso de Código Máquina, analizamos los conceptos básicos de la intrincada organización de la pantalla de nuestro **Amstrad**.



12 Análogo

Uno de los pilares fundamentales en el que se apoya la Inteligencia artificial, es el reconocedor de modelos. Las posibilidades que se pueden sacar de esta aplicación podrían calificarse de infinitas, sobre todo, allí donde es necesario un acercamiento al lenguaje natural.

18 Primeros pasos

Comenzamos con este artículo la exploración de las formas de almacenamiento exterior del **Amstrad**, explicando en estas líneas, profundos conceptos de este importante tema.



Programación 22

Hasta ahora, al menos el manual del **Amstrad** así lo indicaba, tener más de cuatro colores en pantalla en modo 1, por ejemplo, no era posible. Descubre cómo no sólo esto es falso, sino que es posible tener hasta ocho colores distintos en modo 1.

Util al programador

30

Con el monitor de Roms que ofrecemos podréis investigar tanto en las dos **ROMS** que el **Amstrad** trae de fábrica, como en cualquiera de las **ROMS** que podáis acoplar a su bus trasero.

PROCESO CONTABLE BOOKKEEPER V.3.0 AMSTRAD PCW8256/PCW8512

CARACTERISTICAS TECNICAS

Aplicación: Contabilidad General + Vencimientos.

Equipo: Amstrad PCW8256 y PCW8512.

Almacenamiento externo: 1 ó 2 unidades de disco.

Sistema operativo: CP/M PLUS.

Capacidad: (orientativo). PCW8256: 5.000 apuntes. 256 cuentas (nivel 3 dígitos). 500 subcuentas (nivel 7 dígitos). 500 vencimientos. 100 conceptos automáticos.

PCW8512: 10.000 apuntes. 256 cuentas (nivel 3 dígitos). 1.000 subcuentas (nivel 7 dígitos). 500 vencimientos. 100 conceptos automáticos.

Informes. Monitor e impresora: consulta del Diario. Extracto de subcuentas con posibilidad de selección por conceptos. Balance de saldos iniciales. Balance de sumas y saldos de subcuentas. Balance de situación. Cuenta de resultados. Vencimientos pendientes. Vencimientos pagados. Listado del Plan General Contable. Listado de subcuentas. Listado de conceptos automáticos.

Otras características:

—Introducción de apuntes con editor de pantalla. Comandos para borrado de un carácter o de todo el texto, opción inserción/sobreimpresión, movimiento del cursor.

—Posibilidad de introducir apuntes de contrapartida automáticamente.

—Alta de vencimientos a través del Diario.

—Posibilidad de modificar apuntes en cualquiera de sus campos.

—Módulo de introducción de saldos iniciales.

—Módulo de introducción de saldos iniciales y finales en las existencias.

—Pago de vencimientos.

—Alta, baja y modificación de cuentas, subcuentas y conceptos.

—Actualización del fichero de vencimientos.

—Cierre automático del ejercicio contable.

—Cierre del período.

—Remuneración de apuntes por fechas.

DESCRIPCION

El proceso Contable Bookkeeper V.3.0 en su versión para el ordenador PCW ha sido desarrollado a partir de la experiencia de las sucesivas versiones para el CPC6128. Como en los anteriores programas se ha buscado proporcionar al usuario **una herramienta útil, de fácil manejo, capacidad para pequeñas y medianas empresas y presentación cuidada.**

La organización del programa se ha simplificado al máximo, creándose **un solo directorio general desde el cual salta directamente a cualquiera de las 36 opciones disponibles.** La selección se realiza a través de las teclas de movimiento del cursor. La tecla «SAL» permite en cualquier momento el retorno al directorio general.

El programa incluye el Plan General Contable para Pequeñas y Medianas Empresas al objeto de facilitar la labor al usuario con poca experiencia contable. Dicho plan es modificable y ampliable.

En la página 3 se reproducen por «hard-copy» tres de las pantallas del programa: el directorio general, el módulo de introducción de apuntes y el extracto de subcuentas.

P.V.P. **16.500** + IVA.



KONAMI ABRE MERCADO EN MADRID

La prestigiosa firma Konami, una de las grandes líderes del mercado, en el sector de los videojuegos, abrió el día 14 del pasado mes un local comercial en **Madrid.**

Lo más importante de esta noticia es que **Shop Konami**, no sólo va a ser el único comercio abierto en **España** por esta compañía, sino el primer centro comercial de Europa de la prestigiosa firma japonesa.

Con el asentamiento definitivo de la microinformática en España, el mercado de juegos en nuestro país es una de las bases más sólidas del sector, razón que parece estar convenciendo a más de uno, para venirse a nuestras cálidas tierras mediterráneas.

Divagaciones aparte, lo importante es eso, **Konami** ha abierto su primera tienda en Europa y para su ubicación ha elegido a España, más concretamente en:

**Francisco Navacerrada, 19
Madrid**



FORTH COMPILER

Forth Compiler es una implementación del popular lenguaje **Forth-79**, potenciado con un conjunto de ampliaciones orientadas a la explotación del sistema sobre el que trabajamos. **Forth Compiler** es un lenguaje que combina la facilidad de programación de un lenguaje de alto nivel, al ofrecer al programador estructuras de control (*BEGIN/UNTIL, WHILE/REPEAT, IF/THEN/ELSE*), con la potencia y rapidez de ejecución del lenguaje de bajo nivel, dotándole así de excepcionales características.

Forth Compiler no utiliza instrucciones o comandos como el lenguaje Basic. En su lugar utiliza palabras. **Forth Compiler** posee un juego de esas palabras, las cuales constituyen el núcleo del lenguaje, constituyendo el denominado diccionario residente. Como en Basic, pero de una forma más flexible, un programa en **Forth Compiler** consta de un número de combinaciones de las mencionadas palabras. Es posible operar en dos modos de trabajo: modo directo o por programa. La opción de modo directo permite ejecutar palabras sin tener que recurrir a su posterior compilación, facilitando así el aprendizaje del lenguaje.

La velocidad de ejecución, al ser un lenguaje compilado, es comparable al Código Máquina. Posee amplios comandos gráficos y de sonido, así como tratamiento de sprites. Todo ello facilita la realización de juegos, sin el inconveniente encontrado en Basic de la lenta velocidad de ejecución.

ESPECIFICACIONES TECNICAS

Forth Compiler posee su propio editor. Este reúne un conjunto de funciones que van desde el borrado del área de trabajo, listado del programa, edición de líneas, modo de inserción automática (*AUTO*), hasta la reenumeración y borrado. Una característica importante que dota a **Forth Compiler** de una gran operatividad es la visualización de las teclas de función en la parte inferior del área de trabajo, referenciando todos los comandos del editor y compilador para una mayor facilidad de ejecución.

Posee otras funciones más específicas, tales como PRONT-PROFF, para la activación-desactivación de la impresora; W para actualizar la última entrada en modo directo, y ESC que nos permite borrar la línea actual y/o desactivar el modo AUTO.

Forth Compiler trabaja con aritmética entera, permitiendo el manejo de enteros en el rango de -32768 a 32767 (*enteros con signo*), o de 0 a 65535 (*enteros sin signo*).

Una de sus opciones más interesantes es la posibilidad de salvaguardar el Código Máquina generado, como un fichero binario autoejecutable por sí solo, en ausencia del compilador.

EQUIPO NECESARIO

Ordenadores **Amstrad** CPC-464, o CPC-664, o CPC-6128.

SUMINISTRO

Cassette o disco de 3", conteniendo el programa **Forth Compiler**.
Manual de instrucciones.

ACTIVIDADES COMERCIALES ELECTRONICAS, S. A.

ACE, empresa que desde su inicio, se ha caracterizado por dos constantes: productos exclusivamente de primera línea y servicio.

Previendo el crecimiento experimentado por **Amstrad** en el resto de Europa, y como representación de los intereses del grupo Indescomp en Cataluña, **ACE, S. A.**, modifica su estructura, contando para ello con la apertura de sus nuevas instalaciones, que actualmente ocupan un total de 1.000 m², un almacén central con una superficie de 400 m², y su ya operativa Sala de Demostraciones «**Amstrad**», que ofrece un servicio desinteresado a fin de facilitar toda la información y soporte que precise el distribuidor, profesional y usuario.

ACE, S. A., posee en la actualidad una plantilla de 30 personas, habiéndose incorporado recientemente a la misma don Alberto Prada como director financiero.

En este aspecto, y como hecho significativo, cabe destacar el acuerdo establecido entre **ACE, S. A.**, y la firma **Arthur Young** para su asesoramiento general y auditoría.

En distinto orden **ACE, S. A.**, ha dotado a sus instalaciones con equipos de gestión **Secoinsa I-500**, desarrollando para ello software de gestión en sistema PICK en tiempo real.

En definitiva, éstas son las primeras direcciones hacia donde han ido encaminadas las inversiones que **ACE, S. A.**, Delegación Grupo Indescomp y empresa especializada de distribución de productos electrónicos de consumo, ha iniciado dentro del atractivo y complejo mundo de la informática.

Primera PLANA

La casa inglesa **Ocean** parece que se está especializando en juegos de temas cinematográficos y televisivos. Siguiendo en esa línea, está a punto de aparecer en el mercado la versión en bytes de la película **Top-Gun** que cuenta la historia de los mejores pilotos de F-14 de la US Navy.

• Con el PCW parece que se confirma el dicho de que **no se puede tomar nada en serio**.

Tenemos noticias del estreno de Alligata Software (*autor del popular «Mercenario» para CPC*) en este ordenador con dos títulos muy sugestivos: '**Guardián**' y '**Blogger**'. El primero fue originalmente diseñado para el Commodore 64 y se trata de un clásico de marcianos. Del segundo sólo sabemos que se basa en laberintos muy complicados.

• Lo que será inminente será la aparición en España de la versión para el PCW del **Tomahawk**, un simulador de vuelo basado en el helicóptero Apache AH-64 de la US Army. Como novedad admite para su manejo el nuevo interface de Joystick que acaba de comercializar, tanto en nuestro país como en el mercado inglés, la casa DK'Tronic.

• También en España se hacen juegos '**Arcade**' para el PCW, y el primero ha sido obra de la producción de ACE Software con la distribución de Micro Byte. Su nombre es **Troglo** y como da a indicar, trata de ambientes trogloditas con versiones paralelas en disco y cassette dirigidas a los CPC 464/472/664 y 6128.

• Fruto del mismo dúo, ACE y Micro, es **El enigma de Acepts** que pasará a nuestra historia como la primera aventura gráfica con voz en castellano. Al estar dirigida a los usuarios de cualquier CPC dotado de disco se recurre a una capacidad total de nada menos **¡400 K de memoria!**, consultando continuamente ficheros de datos. En cuanto el programa esté en nuestras manos, os prometemos cumplida información de su funcionamiento.



FE

INTERN
AMSTRAD

¡Ven a conocer el apasionante mundo de

Un mundo que comienza con el espectacular SPECTRUM+2 y se cierra con el revolucionario AMSTRAD PC 1512, pasando por toda la gama de los increíbles CPC 464, CPC 6128 y los procesadores de textos PCW 8256 y PCW 8512. Desde el ordenador de iniciación a la informática al más completo equipo profesional, reunidos en una ocasión única y singular: la Feria Internacional AMSTRAD-SINCLAIR.

Las más importantes empresas europeas se dan cita en Madrid para presentar sus más recientes productos:

Programas de acción, juego, aventuras... Programas educativos, de utilidades, lenguajes... Programas de gestión y profesionales...

Periféricos, ampliaciones de memoria, tabletas gráficas, digitalizadores, impresoras, redes de comunicación, sintetizadores de voz, correo electrónico, proceso de textos, tratamiento de imagen y gráficos...

Libros, revistas, cursos...

Una oportunidad extraordinaria para "estar al día".

¡¡Ven de compras a la Feria Internacional AMSTRAD-SINCLAIR!!

RIA

ACIONAL

sinclair

los ordenadores **AMSTRAD** y **SINCLAIR!**

- Patrocinada y organizada por AMSTRAD ESPAÑA.
- Horario continuo de 10,00 a 19,30
- Entrada: 250 Ptas.
- Sorteo de Ordenadores AMSTRAD y SINCLAIR ante los visitantes.

FERIA
INTERNACIONAL
AMSTRAD

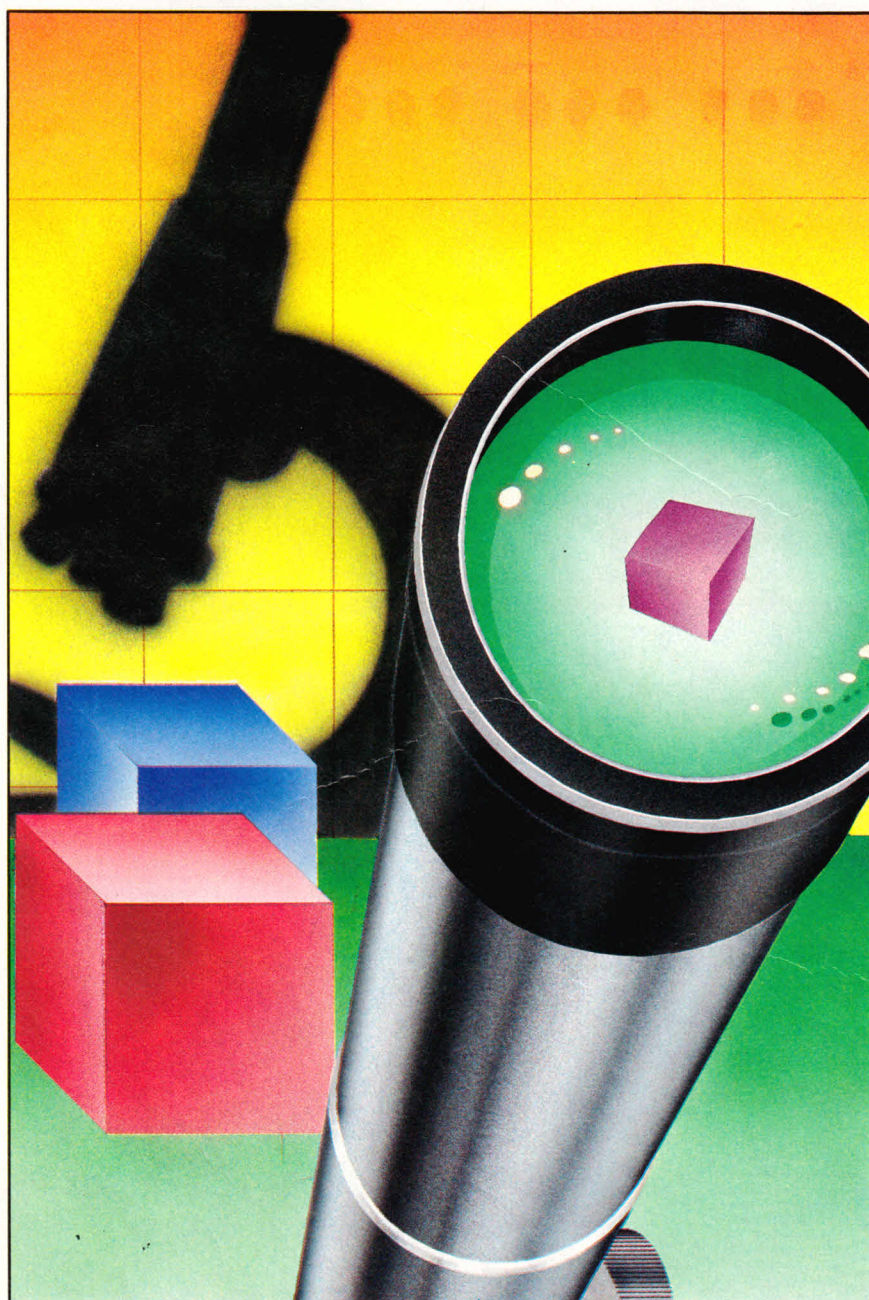
sinclair

12,13 y 14 de Diciembre
Palacio de Exposiciones y Congresos
P.º Castellana, 99. 28046 MADRID

ESTA ES TU FERIA

LA PANTALLA DEL AMSTRAD AL MICROSCOPIO (I)

Empezaremos esta semana con el estudio de la pantalla de nuestro Amstrad. Veremos cómo se puede pokear directamente para obtener gráficos, las direcciones de memoria que utiliza, cómo las utiliza y los valores que se deben dar para obtener los diferentes colores que existen.



o primero que se debe conocer para imprimir directamente en pantalla, son las posiciones de memoria que son usadas y la forma en que se utilizan.

Así pues, la memoria de pantalla empieza a partir de la dirección hexadecimal &C000. A dicha dirección corresponde el primer byte del carácter superior izquierdo de la pantalla.

En el gráfico que se muestra a continuación, podemos ver cómo está organizada la pantalla:

	Columna 1	Columna 2	Columna 80
FILA 1	&C000	&C001	&C04F
	&C800	&C801	&C84F
	&D000	&D001	&D04F
	&D800	&D801	&D84F
	&E000	&E001	&E04F
	&E800	&E801	&E84F
	&F000	&F001	&F04F
	&F800	&F801	&F84F
FILA 2	&C050	&C051	&C09F
	&C850	&C851	&C89F
	&D050	&D051	&D09F
	&D850	&D851	&D89F
	&E050	&E051	&E09F
	&E850	&E851	&E89F
	&F050	&F051	&F09F
	&F850	&F851	&FF9F
FILA 25	&C780	&C781	&C7CF
	&CF80	&CF81	&CFCF
	&D780	&D781	&D7CF
	&DF80	&DF81	&DFCF
	&E780	&E781	&E7CF
	&EF80	&EF81	&EFCF
	&F780	&F781	&F7CF
	&FF80	&FF81	&FFCF

Estas posiciones de pantalla son válidas mientras no se produzca un scroll de pantalla, ya que éste produce un deslizamiento de dichas direcciones. Por lo tanto, siempre que deseemos trabajar, deberemos inicializar dichas direcciones con el comando Basic:

MODE X

donde «X» debe ser el modo de pantalla en el que se desea trabajar.

Igualmente, trabajando en código máquina deberemos inicializar la pantalla con la siguiente rutina:

```
LD A,X
CALL #BC0E
```

donde «X» deberá ser también el modo de pantalla.

Como se puede comprobar viendo el gráfico anterior, para pasar de un carácter a otro dentro de la misma fila, únicamente hace falta incrementar en una unidad la memoria de pantalla.

Para pasar de un carácter a otro dentro de dos filas distintas contiguas, debemos sumar 80 a la dirección de memoria anterior, ya que éste es el número de columnas que forman la pantalla.

Y, por último, para movernos de un byte a otro contiguo dentro del mismo carácter, deberemos sumar a la dirección anterior el valor 2048.

Debemos tener siempre presente que el primer byte del carácter situado en la esquina superior izquierda de la pantalla corresponde a la dirección de memoria &C000.

Por lo tanto, para calcular la dirección de pantalla correspondiente a un carácter cualquiera, deberá procederse de la siguiente manera:

Dirección = &C000 + 80 * Y + X

donde «Y» será el número de fila donde se encuentra dicho carácter, y «X» el número correspondiente de columna.

Veamos, por ejemplo, cómo debería calcularse la dirección de pantalla correspondiente al carácter situado en la fila 10, columna 20.

Desde Basic efectuaríamos el siguiente programa:

```
10 MODE 2
20 FILA = 10
30 COLUMNA = 20
40 DIRECCION = &C000 + 80 * FILA
+ COLUMNA
```

El mismo programa en código máquina, lo realizaríamos de la siguiente forma:

```
LD HL,#C000 ;Dirección inicial
LD DE,50 ;Dif. entre filas
LD B,10 ;Número de fila
BUCLE: ADD HL,DE
DJNZ BUCLE
LD D,0
LD E,20 ;Núm. de columna
ADD HL,DE
RET
```

Así pues, una vez ejecutado el programa anterior, obtendríamos en el registro doble HL, la dirección donde se encuentra situado nuestro carácter.

Estamos, pues, en condiciones de calcular la dirección de pantalla para cualquier carácter que se encuentre dentro de ella, para imprimir algo en esa dirección, únicamente tendremos que pokear un valor en dicha dirección, con lo cual conseguiremos iluminar los píxel correspondientes al valor introducido.

De esta forma, para iluminar todos los bits correspondientes al primer byte del carácter situado en la esquina superior izquierda de la pantalla, deberemos ejecutar el siguiente programa:

```
10 DIR = &C000
20 POKE DIR,255
```

Para efectuar el mismo trabajo desde código máquina, bastará con efectuar:

```
LD HL,#C000
LD (HL),255
```

La forma en que se han calculado las direcciones correspondientes al carácter elegido, hacen referencia al modo de pantalla en alta resolución, es decir, en modo 2, ya que en el mismo coinciden el número de carácter por fila con el número de bytes horizontales que posee la pantalla.

Código MAQUINA

Por lo tanto, si nos encontramos en el modo 1, para calcular la dirección de memoria de un carácter, deberemos dividir o multiplicar el número correspondiente de fila por 2, ya que dicho modo únicamente posee 40 caracteres por fila.

Del mismo modo, para calcular la dirección de pantalla de un carácter cualquiera en modo 0, deberemos multiplicar el número de fila correspondiente por 4, ya que en este modo únicamente existen 20 caracteres por fila.

Vamos a ver ahora cómo podemos conocer las direcciones de pantalla de los 8 bytes correspondientes a un carácter, sabiendo cuál es la dirección que ocupa el primero de ellos.

Realmente la cuestión es bien sencilla, ya que para pasar de uno a otro, únicamente hace falta sumar al primero el valor 2048.

```

Hi soft GENA3.1 Assembler. Page 1.

Pass 1 errors: 00

A000          10          ORG #A000
A000 3E02       20          LD A,2
A002 CD0EBC    30          CALL #BC0E
A005 2100C0    40          LD HL,#C000
A008 115000    50          LD DE,80
A00B 060A      60          LD B,10
A00D 19        70          BUC: ADD HL,DE
A00E 10FD      80          DJNZ BUC
A010 1600      90          LD D,0
A012 1E28     100         LD E,40
A014 19       110         ADD HL,DE
A015 1126A0   120         LD DE,DAT0
A018 0608     130         LD B,8
A01A 05       140         BUC1: PUSH BC
A01B 1A       150         LD A,(DE)
A01C 77       160         LD (HL),A
A01D 13       170         INC DE
A01E 010008   180         LD BC,2048
A021 09       190         ADD HL,BC
A022 C1       200         POP BC
A023 10F5     210         DJNZ BUC1
A025 C9       220         RET
                230
A026 183C6666 240 DAT0: DEFB 24,60,102,102,126,102,102,0

Pass 2 errors: 00

BUC  A00D  BUC1  A01A  DAT0  A026

Table used: 45 from 136

```

Así pues, si por ejemplo la dirección correspondiente al primer byte de un carácter fuera la &C000, las direcciones correspondientes a los restantes bytes de ese mismo carácter, serían las siguientes:

byte 2...&C000 + 2048
 byte 3...&C000 + 2*2048
 byte 4...&C000 + 3*2048

byte 8...&C000 + 7*2048

Conocidas, pues, las direcciones correspondientes a todos los bytes que forman un carácter, estamos en condiciones de imprimir en pantalla el carácter que nosotros deseemos, cargando en dichas direcciones los valores correspondientes al carácter elegido.

En el programa ejemplo que os ofrecemos esta semana, podremos ver cómo imprimimos el carácter «A» en la posición correspondiente a la fila 10, columna 40.

Como podremos comprobar, hemos elegido como pantalla de trabajo el modo 2, ya que en dicho modo los bits correspondientes a cada byte únicamente informan de los píxeles que se deben o no, iluminar

En los otros dos modos de pantalla dichos bits llevan además la información del color con el que se deben imprimir en pantalla.

Esto último se estudiará en el próximo capítulo de código máquina, donde se indicará cuáles son los bits que informan del color en cada uno de los modos de pantalla.



```

10 MODE 2
20 FILA=10:COLUMNA=40
30 DIRECCION=&C000+10*80+40
40 FOR N=0 TO 7
50 READ A:POKE DIRECCION,A
60 DIRECCION=DIRECCION+2048
70 NEXT
80 REM *DATAS PARA OBTENER EL CARAC
TER 'A'*
90 DATA 24,60,102,102,126,102,102,0
10 FOR N=&A000 TO &A02E
20 READ A:SUMA=SUMA+A
30 POKE N,A
40 NEXT
50 IF SUMA<>&D1C THEN PRINT "ERROR
EN DATAS"
60 DATA 62,2,205,14,188,33,0
70 DATA 192,17,80,0,6,10,25
80 DATA 16,253,22,0,30,40,25
90 DATA 17,38,160,6,8,197,26
100 DATA 119,19,1,0,8,9,193
110 DATA 16,245,201,24,60,102,102
120 DATA 126,102,102,0,255,73,78
  
```

IMPOSSIBLE MISSION.™

YOUR MISSION-TO SAVE THE WORLD.



EPYX
COMPUTER SOFTWARE

Strategy Games for the Action-Game Player



Como miembro del Escuadrón Antiterrorista de Computadores (EAC), su misión es encontrar y coger al infame Elvin, quien tiene a la población mundial como rehén bajo la amenaza de aniquilación nuclear. Debe pasar por un pasillo a través de habitaciones y túneles en su cuartel general tratando de evitar los robots protectores de Elvin.

Deberá correr más o saltar por encima del siguiente robot o jugar seguro y tomar el tiempo necesario para unir los códigos para desactivar los robots, encontrar a Elvin y pararle.

P. V. P.: COMMODORE 64, 2.300 ptas.

P. V. P.: SPECTRUM 48 K, 2.300 ptas.

Utilice su cámara para fotografiar tantas pistas como sea posible para encontrar la contraseña que le permitirá penetrar en la habitación de control de Elvin.

Su misión: Salvar al mundo, ¡pero deprisa!

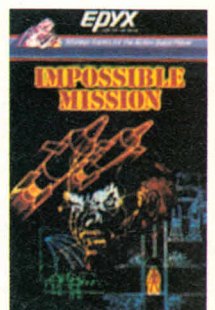
Un jugador: control joystick o teclado (sólo SPECTRUM).



Fabricado y distribuido bajo licencia por:

Santa Cruz de Marcenado, 31 - 28015 Madrid - Teléf. 241 1063

Distribuido en CATALUÑA y BALEARES por: DISCLUB, S.A.
Balmes, 58 08007 BARCELONA Teléfono (93) 302 39 08



RECONOCEDORES DE MODELOS (II)

Hace unos números comentamos, con bastante detalle, el propósito y funcionamiento de los reconocedores de modelos, además de explicar un pequeño programa en Logo que lo implementa. Nos quedamos a punto de explicar el procedimiento «principal», y de darle a nuestro reconocedor de modelos un poco más de potencia e «inteligencia».

por José María Díaz



Como puede verse en los listados, hemos añadido dos nuevos procedimientos: «iguales3» y «listp». Antes de explicar su funcionamiento, junto con el «principal», que dejaremos para el final, vamos a tratar de comprender por qué son necesarios y qué nuevas facilidades añaden al reconocedor de modelos.

En el artículo anterior, ya habíamos conseguido que el programa, dado un modelo por complejo que sea, fuera reconocido.

El siguiente paso es poder «entresacar» de la frase el trozo que deseemos, para así poder manipularlo de la manera en que se necesite.

Para poder hacer esto, se ha ampliado la «sintaxis» de la frase modelo, de tal forma que admita la nueva «palabra clave» [bind "xx"].

Esta expresión, tan rara, responde a la necesidad de que las palabras clave elegidas no sea fácil que aparezcan en nuestro idioma, por lo que se ha recurrido al inglés.

Así, «bind» puede traducirse, bastante libremente, como «asignar», y «xx», en este caso, representa la variable en la cual el programa nos devolverá la parte de la frase input seleccionada.

Por ejemplo, si el modelo es:

me any mucho [bind "xx"]

y como frase de entrada decimos:

me gusta mucho mi Amstrad

el programa diría que ambas frases son iguales, y el contenido de «xx» sería:

mi Amstrad

En definitiva, la palabra clave «bind» le está diciendo al ordenador que algo especial ocurre. Los «especial» es asignar a «xx» las palabras que vengan detrás, respetando los condicionamientos respecto a las claves «any» y «many» que explicamos en el artículo anterior.

De esta forma hemos aumentado considerablemente la potencia del reconocedor de modelos, sin perder su flexibilidad a la hora de reconocerlos.

Vamos a estudiar los nuevos procedimientos, comenzando por «listp».

Los usuarios del 6128 y del PCW verán que esta orden está implementada como primitiva en su versión del Logo. El resto no son tan afortunados. Pensando en ellos, hemos escrito esta orden que funcionará sin cambios en los 6128 y PCW. Su función es decirnos si un «algo» es o no una lista.

Definimos el nombre del procedimiento y su argumento.

TO LISTP :LISTA

Ahora viene el meollo del procedimiento:
OP (NOT (WORDP :LISTA))

La orden OP ya es conocida: le indica a la máquina que salga del procedimiento y retorne al nivel superior.

La orden

WORP: LISTA

devolverá un resultado cierto o falso, según que :LISTA sea una palabra o no lo sea. Por tanto, la secuencia

NOT (WORDP :LISTA)

invierte el resultado de WORDP, es decir, si lista es una palabra, WORDP dará verdadero, y el NOT invertirá el resultado, con lo que el procedimiento dará el resultado falso.

Si :LISTA es efectivamente una lista, WORDP dará falso, y NOT WORDP verdadero.

De esta forma tortuosa, pero útil, podemos averiguar si un objeto es una lista o no lo es.

END

Fin del procedimiento.

LISTP será muy necesario, por razones que veremos luego, al procedimiento «iguales», en la última línea que hemos introducido para reflejar la aparición de «iguales3», la nueva estrella del reconocedor de modelos.

Por tanto, antes de estudiar en profundidad «iguales3», echemos una ojeada a la nueva línea:

a) IF AND (LISTP FIRST :MOD) (FIRST FIRST :MOD="bind")

b) [OP IGUALES3 :MOD :INP]

c) [OP 1=0]

La estructura de la línea es un típico IF...THEN...ELSE del Basic.

Si la condición del IF se cumple, THEN se ejecuta b), ELSE se ejecuta c).

En a), existe una condición doble, que debe cumplirse simultáneamente, como expresa el AND.

Lo primero que tiene que ocurrir es que sea cierto

LISTP FIRST :MOD

esto es, que en la frase modelo haya una lista, que «casualmente» será nuestro [bind "xx"].

Si esto no aparece, la acción b) no se ejecuta. Ahora bien, ¿qué pasa si nosotros metemos una lista en el modelo porque nos interese, o nos dé la gana hacerlo?

Bien, entonces entra en acción la segunda condición de la línea a):

(FIRST FIRST :MOD="bind")

que expresa que, una vez localizada la lista [bind "xx"], o la que sea, su primer elemento debe ser precisamente «bind»; por eso hemos escogido una palabra tan rara. Si, a pesar de todo, usted introduce una lista que empiece por «bind», pues..., el programa no se bloquearía. Simplemente se le asignaría al segundo elemento de la lista el trozo seleccionado.

Resumiendo; supongamos que el modelo es

me gusta [bind "xx"] Amstrad

y el input

me gusta mi Amstrad

tarde o temprano, si no se corta la luz, el programa habría comparado «me» y «gusta» con el input, decidido que son iguales y se enfrentaría con el famoso [bind "xx"].

Ambas condiciones de a) se cumplirán, por que el objeto es una lista y su primer elemento es «bind», por lo cual se ejecutaría b) y entraría en acción el procedimiento «iguales3», que veremos luego.

Examinemos qué pasaría si tratamos de engañar al programa y ponemos como modelo:

me gusta [bind "nisesabe"] Amstrad

con el mismo input.

Bien, el programa daría como resultado iguales, y si salimos de él y le damos al logo la orden

:nisesabe

se imprimiría en pantalla

mi

El resultado es el mismo, pero el trozo seleccionado se asigna al segundo elemento de la lista que comienza con «bind».

Las líneas b) y c) simplemente llaman a «iguales3» si a) se cumple y si no, c) da un resultado de falso, y el procedimiento «iguales» acaba.



ANALOGO

Vamos a emprenderla ahora con «iguales3». Primero, como de costumbre, el nombre y los argumentos:

TO IGUALES3 :MOD :INP

Las líneas siguientes son algo difíciles de comprender, a pesar de que su lógica es muy simple. Es necesaria una máxima atención.

1) IF (IGUALES (BF :MOD) (BF :INP))

2) [MAKE RUN BF FIRST :MOD :FIRST :INP
OP 1=1]

Recordemos que este procedimiento aparece cuando el modelo y el input, por acción de iguales se han visto reducidos a lo siguiente:

[bind "xx"] Amstrad
mi Amstrad

suponiendo el mismo modelo e idéntico input.

En 1) se vuelve a llamar a «iguales» para manejar la posibilidad de que la asignación sea de un solo elemento, como sería el caso en este ejemplo; «xx», al final, contiene «mi».

Como la condición del IF se cumple, se ejecuta 2), que vamos a descomponer en elementos más sencillos:

2A) [MAKE

2B) RUN BF FIRST :MOD :FIRST :INP

2C) OP 1=1

Bueno, en 2A usamos la sentencia de asignación del Logo, equivalente al LET del Basic, la cual necesita dos argumentos, la variable y el contenido.

En 2B es donde se encuentra el meollo de la cuestión. La orden RUN es una de las más potentes del Logo, y ejecuta una lista de objetos como si fueran órdenes del lenguaje. Por ejemplo, es equivalente teclear:

pr "HOLA

run [pr "HOLA]

como se puede comprobar. Para ver lo que pasa en 2B, vamos a descomponerlo de nuevo:

2B1) BF FIRST :MOD

2B2) FIRST :INP

y a ver sus valores, según la situación inicial descrita un poco más arriba.

El valor de FIRST :MOD es [bind "xx"]

El valor de BF FIRST :MOD es JUSTAMENTE [xx]

El valor de FIRST :INP es "mi

Por lo que RUN se encuentra que tiene que ejecutar [[xx] "mi] y después de hecho esto, en 2A) queda

MAKE "xx "mi

con lo cual hemos conseguido el objetivo deseado, asignar a una variable un trozo pre-seleccionado de la frase input.

PROCEDIMIENTO PRINCIPAL

```
to principal
(local "11 "12)
pr [ ] PULSA 'ENTER' PARA TERMINAR]
label "loop
pr [DIME LA PRIMERA FRASE]
type [ ]
make "11 r1
pr [DIME LA SEGUNDA FRASE]
type [ ]
make "12 r1
if (or (empty :11) (empty :12)) [op "adios]
if (iguales :11 :12) [pr "iguales] [pr "distintos]
pr [EL TROZO SELECCIONADO ES]
if (empty :xx) [pr [NINGUNO]] [pr :xx]
make "xx [ ]
go "loop
end
```

PROCEDIMIENTO PRIMERO

```
to iguales :mod :inp
if (and (:mod = [ ]) (:inp = [ ])) [op 1 = 1]
if (or (:mod = [ ]) (:inp = [ ])) [op 1 = 0]
if ((first :mod) = (first :inp)) [op iguales (bf :mod) (bf :inp)]
if (first :mod) = "any [op iguales (bf :mod) (bf :inp)]
if (first :mod) = "many [op iguales2 :mod :inp]
if and (listp first :mod) (first first :mod = "bind) [op iguales3 :mod :inp] [op 1 = 0],
end
```

PROCEDIMIENTO SEGUNDO

```
to iguales2 :mod :inp
if (iguales :mod (bf :inp)) [op 1 = 1] [op iguales (bf :mod) (bf :inp)]
end
```

PROCEDIMIENTO TERCERO

```
to iguales3 :mod :inp
if (iguales (bf :mod) (bf :inp)) [make run bf first :mod first :inp op 1 = 1]
if (iguales :mod (bf :inp)) [make run bf first :mod :inp op 1 = 1] [op 1 = 0]
end
```

PROCEDIMIENTO CUARTO

```
to listp :lista
op (not (wordp :lista))
end
```

Por último, las líneas:

- 3) IF (IGUALES :MOD (BF :INP))
- 4) [MAKE RUN BF FIRST :MOD :INP OP 1=1]
- 5) [OP 1=0]

hacen exactamente lo mismo de análoga manera, pero manejan el caso en que hay que asignar a "xx MAS DE UNA PALABRA. De ahí la diferencia de las líneas 4) y 2).

Un vez traspasado este «muro», queda lo más simple: el procedimiento «principal». Vamos a ver brevemente las líneas más importantes.

```
TO PRINCIPAL
(LOCAL "1 "2)
```

Nombre del procedimiento y definición de dos variables locales a él.

```
LABEL "LOOP
```

Etiqueta de comienzo del bucle principal.

```
MAKE "L1 RL
MAKE "L2 RL
```

El resto de las líneas del procedimiento son evidentes, y hemos visto clones suyos en otros artículos.

Bien, el reconocedor de modelos está completo. Yo creo que merece la pena experimentar con él para ver todas sus posibilidades y descubrir sus fallos, si es que algo se nos ha pasado por alto.

En un próximo artículo veremos una aplicación práctica, consistente en un programa extremadamente sencillo que nos permitirá conversar en lenguaje natural con la computadora.

Antes de despedirme, me gustaría llamar la atención sobre la inmensa potencia que tiene el Logo en el tratamiento de listas y en la construcción de programas en los que unos procedimientos se llaman a otros sucesivamente. Obsérvese que «iguales», «iguales2», e «iguales3» se comportan así. Lo importante es que el programador no tiene que ocuparse de todo el follón personalmente. Para eso está el ordenador y el Logo.



¿TE GUSTARIA SER UN HEROE?

RECORTA ESTA ESTRELLA
Y ENVIALA A SERMA CON
TU NOMBRE Y DIRECCION.
RECIBIRAS UN POSTER DE
BIGGLES COMPLETAMENTE
¡GRATIS!

VIAJA EN EL TIEMPO POR
ESTA MULTIAVENTURA GRAFICA
PARA ENCONTRAR Y DESTRUIR
EL ARMA SECRETA ALEMANA.

AMSTRAD cassette - 2000 pts
disco - 4000 pts
SPECTRUM 2000 pts
COMMODORE 2000 pts

Biggles

VIVE LA PELICULA DESDE TU ORDENADOR!

EN TODOS LOS CINES DE ESPAÑA
DISTRIBUIDA POR C.B. FILMS.

Proximamente a la venta en
la nueva Konami shop
Cl Francisco Navacerrada 19

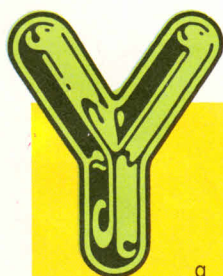
MIRROSOFT

SERMA

LAS TRES LUCES DE GLAURUNG

El sudor se perla en la frente bajo tu casco. No puedes ver el peligro pero lo sientes, sientes el aura de una maldad del más allá, de lo desconocido. Cogiendo aire y haciendo acopio de valor, te dispones a abrir la puerta por la que entrarás en los dominios del señor del mal. ¿Qué te aguardará tras esa puerta...?

Compatible: CPC/464, CPC/664 y CPC/6128



Ya estamos listos para sacar nuestra ajada armadura del desván, y enfrentarnos a otros cien mil peligros para salvar a nuestra damisela de las terribles garras del tirano de turno.

Otro juego, en este caso de ERBE, que pasará a engrosar las listas de los destinados a salvar doncellas secuestradas, mediante el sistema de recoger artilugios, gemas, aparatos, etc. Pero vayamos al grano.

Al comenzar nos encontramos situados en una sala del castillo, y a partir de aquí, deberemos hacer un vasto recorrido por todas sus dependencias, incluyendo dentro de éstas las mazmorras y tenebrosos pasadizos secretos, para recoger las tres Piedras Estelares que necesitamos para vencer a Zwohlan que tiene retenida a la fuerza a nuestra amada.

Como es evidente, este recorrido está plagado de peligros a los que deberemos enfrentarnos con nuestro arco y flechas —en esta ocasión no utilizamos espada—, así como dando saltos y cayendo encima de nuestros oponentes. Pero hay que llevar cuidado, pues ellos, también nos pueden matar con sus armas, además de si caen encima de nosotros.

A quien no podemos enfrentarnos sin nin-

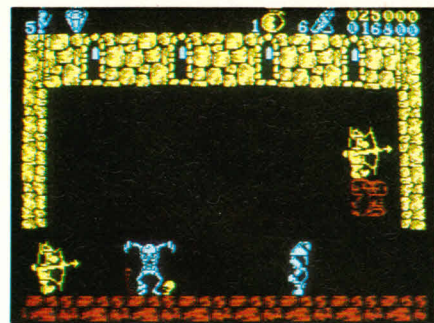
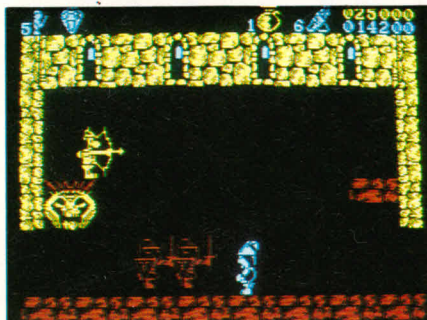
guna de las tres gemas, es al mago del castillo y a su dragón, pues sin éstas no les afectan las flechas. Hablando de flechas, sólo disponemos de seis, o sea que hay que ahorrarlas procurando acabar con nuestros enemigos saltando y cayendo encima de ellos.

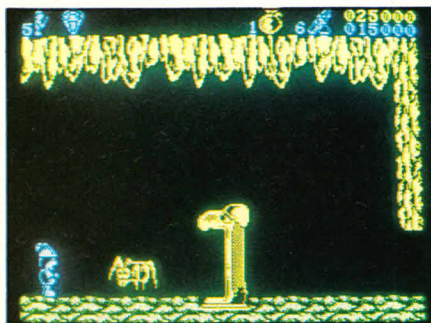
También debemos procurar evitar que nos maten, pues sólo disponemos de seis vidas y el camino a recorrer es largo.

Y para conservarlas, tendremos que vencer a una legión de arqueros, espadachines y lanceros, además de acabar con los murciélagos, vampiros y arañas gigantes que nos encontramos en mazmorras y corredores.

Cada uno de estos enemigos que matemos nos da una puntuación y ésta es:

ARQUEROS	Puntos 600
LANCEROS	800
ESPADACHINES	1.000
MURCIELAGOS	400
ARAÑAS	1.200





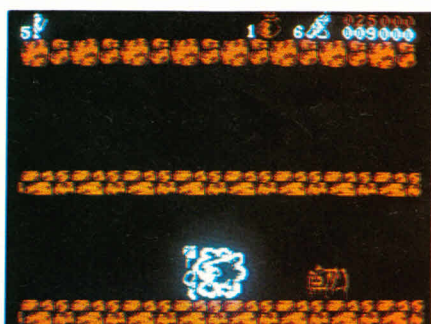
Como podemos conseguir muchos puntos es matando al mago y a su dragón, aunque esto no es imprescindible para conseguir culminar nuestra misión con éxito.

Lo que sí debemos lograr, una vez que tengamos en nuestro poder las tres gemas, es la llave de la puerta de salida del castillo. Pero el mago, que es un astuto zorro, ha puesto otra en uno de los cofres para confundirnos, así que para evitarnos problemas será interesante conseguir las dos.

Distribuidos por el inmenso castillo se encuentran los cofres donde están guardadas las Piedras Estelares y la llave de la salida.

Lo que no vamos a decir es la **«animal»** sorpresa que se oculta en alguno de los cofres. Realmente graciosa. Los magos de la época debían tener su estado mental un poquito alterado.

Pero aunque es una sorpresa simpática, entraña cierto peligro ya que nos deja indefen-



zos al no poder utilizar nuestras flechas mientras dura el hechizo del cofre y esto hace que tengamos que redoblar nuestros cuidados.

Con los ojos bien abiertos tendremos que caminar a través de pasadizos con suelo de hirvientes burbujas, que al aparecer sin previo aviso, pueden originarnos la muerte.

Una cosa que requiere habilidad y precisión es manejar a nuestro caballero cuando debemos salir de las galerías de la parte baja del castillo. Esto lo tenemos que hacer saltando a una especie de cornisas o aleros, lo que resulta un tanto difícil debido a que no podemos manejar al guerrero una vez iniciado el salto.

Pero como no todo va a ser sufrir, tendremos la ocasión de relajarnos cuando en algunas habitaciones veamos a unos graciosos elefantes sentados en una banqueta, o cuando para pasar de alguna sala a otra tengamos que entrar por una puerta en forma de boca-

Mister JOYSTICK

za de guerrero abierta, cuyas hojas están formadas por los propios dientes.

Dentro de algunos cofres encontraremos oro y pócimas que podremos utilizar. Si es oro, para matar las arañas y en el caso de las pócimas para conseguir la invisibilidad o la inmunidad durante un período de tiempo. Cuando somos invisibles los enemigos no nos ven y, por lo tanto, no se dirigen a matarnos, pero si por casualidad una flecha se cruzara en nuestra trayectoria podría matarnos.

Hay dos llaves; una para abrir la puerta de salida, y la otra, aunque no es absolutamente necesaria, puede darnos entrada a alguna estancia donde esté una gema.

Si tenemos alguna, o todas estas cosas, podemos seleccionarlas pulsando la letra N.

Nosotros hemos observado que una o dos de las joyas están colocadas en cofres que se encuentran en las zonas bajas del castillo, donde el peligro es mayor, dando emoción al juego.

Bueno, como se puede ver, el juego es entretenido y entraña cierta emoción, aunque su tema está ya muy trabajado.

Creemos que los programadores que hay, tanto aquí como fuera del país, deberían ir pensando en buscar nuevos temas y planteamientos de juegos.

Pero volviendo a nuestro juego, que pertenece al grupo de los arcades, hemos visto que a nivel gráfico está bien, salvo en el caso del salto del caballero que hubiéramos preferido poder manejarle en el aire.

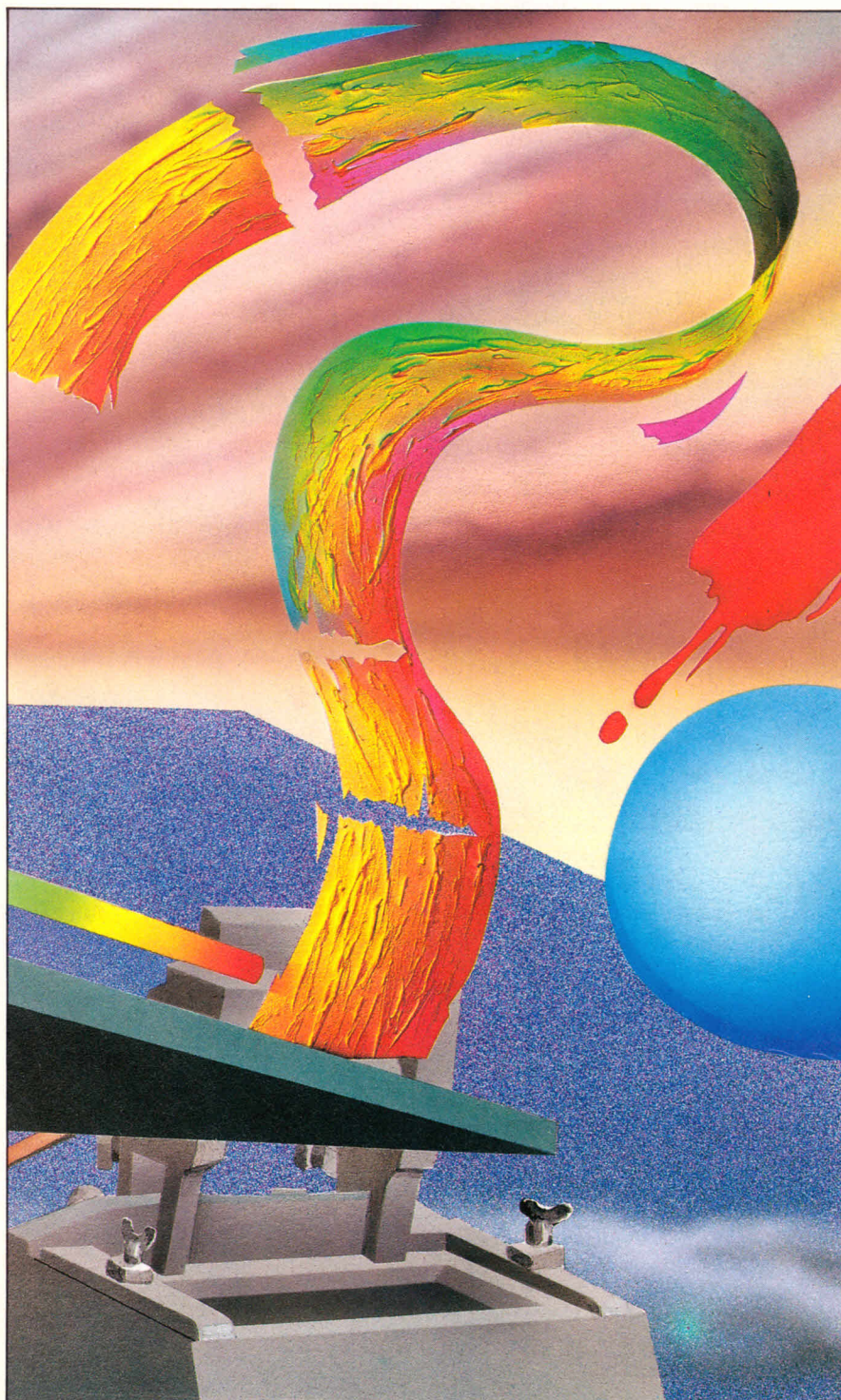
Lo que sí llama la atención es el logradísimo movimiento de las arañas, éste es de un realismo total y de una rapidez importante.

Pues nada, esperaremos con ansiedad que los buenos programadores —de los que sin duda disponemos—, encuentren nuevos aires que nos conduzcan a otra era en que veamos juegos donde nos sintamos partícipes de la acción y nos integremos plenamente.



¡MEMORION!

Tenemos ya escrito nuestro programa. Hemos realizado todas las pruebas unitarias imaginables comprobando los caminos por los que, pensamos, va a recorrer la ejecución del mismo con éxito. Ya podemos, entonces, utilizarle sin miedo. El programa está almacenado en la memoria «central» del ordenador.



ero se trata de un tipo de memoria que hemos llamado «**vólatil**», o sea que todo lo que esté guardado allí desaparece en el mismo instante que demos por finalizada la sesión de trabajo y apaguemos el ordenador, ¡incluido el programa que tanto trabajo nos ha costado terminar!

Toda la información almacenada en la memoria **RAM** del **Amstrad** se borra al cortar la corriente eléctrica que la alimenta. ¿Se supone, entonces, que perdemos todo el trabajo que realizamos durante la última semana, por ejemplo? ¿Tendremos que volver a teclear el largo listado, que resultó de la codificación de un buen programa, cada vez que necesitemos utilizarlo? La solución en las próximas líneas.

Si partimos de su definición una memoria es un «**dispositivo**» que permite almacenar información. Pero además, dicha información ha de estar disponible para que nosotros, los usuarios, podamos acceder a ella a través del ordenador en el momento que necesitemos algún dato de los que guarda.

Cuando hicimos una breve descripción de las partes más importantes de un ordenador descubrimos que no existía un único tipo de memoria.

Nos encontrábamos que uno de ellos era el lugar físico donde teníamos toda la información que manejaba el ordenador directamente. Se encontraba en su interior y nos dio por llamarlo «memoria central».

Otro de los tipos de memoria es el que se encuentra fuera del **Amstrad**, o de cualquier otro ordenador, y por eso las bautizamos con el nombre de «**memorias externas**». A ellas nos vamos a referir en este artículo.

Volvamos con nuestro primitivo programa; ya lo tenemos escrito y perfectamente probado y nos acecha el peligro de verlo desaparecer cuando desconectemos el ordenador. Antes que así ocurra, ¿por qué no guardarlo en una de estas memorias externas donde el mantenimiento de información en su interior no dependerá para nada del estado de nuestro ordenador? «Salvamos» el programa en ellas y podemos, al menos en teoría, apagar la fuente de alimentación del **Amstrad** sin ningún miedo.

Otro día nos apetece volver a ejecutar de nuevo el programa: no hay ningún problema. Cogemos nuestra «**memoria externa**», guardada cuidadosamente en un lugar adecuado; tomamos la información que nos interese, la pasamos a la «memoria central» del ordenador y una vez allí nos encontraremos que el programa que necesitamos ya está a nuestra disposición en el interior del **Amstrad**. De ahí a «**correrlo**» (o ejecutarlo) sólo hay un paso: **RUN**.

¿Está claro?

Clarifiquemos un poquito los conceptos. Hemos decidido utilizar unas llamadas «**memorias externas**» para guardar datos y programas permanentemente hasta que necesitemos usarlos. Por otra parte estos dispositivos de almacenamiento, «**permanente**» (*repetimos*), mantendrán intacta la información independientemente de si el ordenador está encendido o apagado. Además será muy conveniente que tengan una capacidad de almacenamiento muy elevada, o muchos programas y datos en poco espacio, por motivos económicos y de comodidad.

Poniendo a trabajar un poco la imaginación nos podrá responder a esta pregunta: ¿De qué dispositivos estamos hablando?

Seguramente si le hablamos de una «**memoria externa**» todavía ponga una cara de asombro ante estas dos nuevas «palabrejas», pero si oye «**disco magnético**» o «**cinta cassette**» todo le parecerá familiar.

Pues sí, esas son las dos memorias externas, o auxiliares, que habitualmente emplea su ordenador, el **Amstrad**, para guardar su juego favorito, una agenda repleta de direcciones o cualquier otro programa que tengamos por casa.

Nuestro problema inicial ya parece resuelto. Desde ahora almacenaremos todos los programas de fabricación casera así como los que compremos en las tiendas en una cinta cassette o en un disco de material magnético.

Aparte las particularidades físicas de cada uno de ellos, ¿cuál es la principal diferencia existente entre un disco y una cinta? Expliquemos brevemente como se almacenan los datos en uno y otro dispositivo y seguramente podrá obtener la respuesta a la pregunta por sus propios medios.

En una cinta magnética la información se almacena «**secuencialmente**», es decir un dato detrás de otro. Para poder acceder a uno de ellos es necesario que recorramos todos los anteriores así que el tiempo empleado para llegar hasta él dependerá de la posición que ocupe dentro de la cinta.

Podemos deducir de todo esto que la velocidad con la que se localizan y transfieren datos a la memoria central será bastante lenta y el tiempo empleado para «**guardar**» un programa o bien «**cargarlo**» en la memoria será bastante largo.

Pero no todo van a ser inconvenientes; este tipo de dispositivos de «**almacenamiento**» (*cassette*) también tiene sus ventajas: su precio es mucho más económico que el de cualquier otra clase de memorias externas. Lo habrá comprobado en su propio bolsillo, ¿verdad?

Los discos, sin embargo, están divididos en zonas perfectamente delimitadas por algo que se llaman «**pistas**» y «**sectores**». La Unidad Central de Proceso podrá llegar directamente a la información contenida en la zona 3, sector 6, por ejemplo, sin tener que pasar por to-

do lo que habíamos almacenado anteriormente. Por eso a estas memorias se las llama de «**acceso directo**».

Un programa está guardado en la zona 7, sector 9 del mismo modo que una determinada persona vive en la Gran Vía número 64: ambos pueden ser localizados fácilmente sin tener que recorrer el disco desde su comienzo o «**partear**» la Gran Vía desde el número 1.

De ello se desprende que la velocidad con la que el ordenador se sitúa en el lugar exacto donde se encuentra un dato es bastante más rápida que la necesaria para hacer lo propio en una cinta cassette (*acceso secuencial*) así que, el tiempo que dura la transferencia de datos será también mucho más corto.

Ya conocemos algo más: podemos guardar un programa en una cinta o disco magnético si no queremos dejar el ordenador encendido para que no se pierda el contenido de la memoria RAM o no nos apetezca teclear el programa cada vez que necesitemos ejecutarlo.

Todo este proceso lleva consigo un paso o travase de información entre la memoria central y las memorias externas o auxiliares en los dos sentidos.

Guárdelo bien

Cuando guardamos el programa en un disco o cinta, los datos pasan del interior del ordenador a la «unidad» encargada de grabarlos sobre el soporte magnético donde quedan recogidos: llamaremos a esta acción «**almacenar**» o «**salvar**» el programa.

Llega otro día y necesitamos utilizar de nuevo nuestra «**agenda**» de direcciones para escribir felicitaciones de Navidad, por ejemplo. Cogemos nuestra cinta o disco y «cargamos» o recuperamos el programa que necesitamos en la memoria central del ordenador. ¿**Para qué hacer esto?** Toda la información pasará inicialmente a la memoria principal del **Amstrad** para que la UCP pueda tratarla directamente.

Ahora, que ya sabemos en qué consiste el paso de información de un sitio a otro, necesitamos conocer la manera de hacerlo, puesto que si no estaríamos en la misma situación que al comienzo del artículo.

Vayamos por partes. Una vez que tengamos codificado y teclado el programa correctamente, y estemos seguros que funciona tal como lo habíamos diseñado, pasaremos a «salvarlo» en algún soporte físico. Para efectuar la grabación emplearemos el comando SAVE. Su estructura general es:

SAVE «NOMBRE, TIPO», parámetros.

Podíamos explicar en qué consiste cada una de estas cosas. «**Nombre**» es justamente eso, el nombre con el que el programa que tenemos entre manos va a quedar grabado en el disco o la cinta.

«**Tipo**» es el carácter de los datos que quere-

Primeros PASOS

mos grabar: programa Basic, ficheros binarios, ficheros de texto, etc...

Ambos conceptos «**NOMBRE**» y «**TIPO**» han de ir encerrados entre comillas y separados por un punto.

Ha llegado el momento de hacer una separación entre los usuarios de uno y otro soporte físico y contar qué condiciones necesitan cumplir el nombre y el tipo de fichero para cada uno de ellos.

Ocho como máximo

Si hablamos de disco, el «nombre» tendrá como máximo 8 caracteres que pueden ser letras y números pero no los llamados especiales ni espacios en blanco. Teclee el siguiente programa:

```
10 CLS
20 PRINT «ESTO ES UN PRUEBA»
```

y sávelo con el comando:

```
SAVE «PROBANDOLO»
```

Sin duda le aparecerá en la pantalla el mensaje de error:

```
BAD COMMAND
```

que le indica que el nombre con el que pretende guardar el programa no es buen (*tiene más de 8 caracteres*).

El «**tipo**», siempre y cuando estemos utilizando unidad de disco, podrá llevar 3 caracteres como mucho. Si el usuario no especifica este atributo, el sistema generalmente asigna al programa el que le corresponda según la naturaleza de los datos: BAS si es un programa escrito en Basic o BIN si se trata de Código Máquina.

Vayamos con la cinta. Para salvar un programa en ella no es necesario especificar el nombre del mismo. Con:

```
SAVE ""
```

grabaremos el programa sin nombre. Puede tener su aplicación. ¡Ah!, tampoco nos encontraremos la limitación de tener que inventarnos uno que esté compuesto sólo por letras y números. Podemos emplear los signos que queramos.

Queremos salvar en una cinta nuestro «**programín**» anterior y damos al ordenador el comando.

```
SAVE «PRUEBA»
```

seguido de RETURN. En la pantalla aparece el mensaje:

```
Press REC and PLAY then any key
```

Le hacemos caso pulsando los botones REC

y PLAY del cassette y a continuación cualquier tecla del ordenador.

Al comenzar a grabarse cada uno de los bloques que forman nuestro programa, un mensaje semejante a:

Saving PRUEBA blok 1

nos informa de lo que está ocurriendo. Suponga que queremos eliminar de un plumazo todos estos mensajes. Escriba el siguiente comando:

SAVE "¡PRUEBA"

y lo conseguirá. El carácter "!" añadido al principio del nombre hace que el ordenador suprima todos los mensajes anteriormente citados.

Pero volvamos a cosas más generales. En lugar de un programa Basic queremos grabar otro que esté escrito en Código Máquina, que es un lenguaje directamente ejecutable por el ordenador sin necesidad de intérprete. **¿Qué necesitamos?**

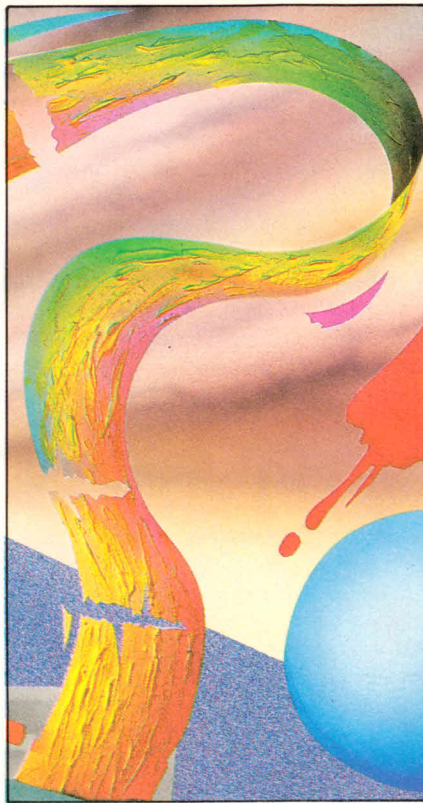
Lo primero es indicárselo al **Amstrad** añadiendo una «B» (*de binario*) detrás de las últimas comillas que encierran el nombre y el tipo y separada por una coma. Además el ordenador necesitará conocer la dirección de memoria a partir de la que queremos grabar así como la longitud del programa expresada en «bytes». ¿Recuerda lo que era un byte? Sí, ¿verdad?

Resumiendo:

SAVE «PRUEBA»,B,30000, 600

salvará en cinta o disco el programa «PRUEBA» o «PRUEBA.BIN» compuesto por 600 bytes tomados a partir de la dirección 30000 de la memoria **RAM** del ordenador.

Una aplicación muy clásica es salvar los datos correspondientes a la memoria de la pantalla. Mediante PLOT, DRAW y otras instruccio-



nes gráficas dibuje un bonito cuadro. Después páselo a una cinta o disco tecleando:

SAVE «CUADRO»,B,&C000,&4000

y podrá tener almacenada una copia imperecedera de su obra maestra para poder presumir delante de sus amigos.

Y no sólo eso. Vuelva a escribir el pequeño programa de prueba. Ahora sávelo con:

SAVE «PRUEBA»,P

A continuación ejecútelo: funciona. Cargue-

lo de nuevo de la cinta o de el disco e intente sacar el listado: no aparece en la pantalla. Hemos salvado un programa que trabaja correctamente pero parece que no existe.

El parámetro P añadido detrás de las comillas indica al ordenador que el programa Basic está protegido y que, o bien lo destruya después de su carga (*con LOAD*) o bien lo haga tras su ejecución (*con RUN «PRUEBA»*). Ya veremos estos comandos.

Es una sencilla forma de cuidar e impedir que algún avisado «aprendiz» de pirata informático visualice los listados de nuestros programas favoritos, que tanto trabajo nos han costado hacer, y los emplee con fines «poco recomendables».

Antes de utilizar esta protección permita que le demos un consejo: primero saque una copia del programa sin proteger y después haga todas las que quiera protegidos. Así evitará perder el listado de su obra.

También podemos guardar un fichero que contenga una serie de caracteres ASCII (*un texto, por ejemplo*). Para ello teclearíamos:

SAVE «TEXTO», A

indicando al ordenador con este parámetro, «A», el tipo de datos que estamos manejando.

Y antes de dejarlo, un pequeño truco. ¿Quiere verificar que la grabación ha sido correcta? Si utiliza disco no hay ningún problema ya que el sistema operativo del ordenador se encarga de ello, pero si todavía emplea la unidad de cinta, rebobínela hasta el comienzo del programa a verificar. A continuación tecle **CAT** y si todos los bloques obtienen la calificación «OK» es que la grabación ha ido bien.

Se acabó. Continuaremos viendo detenidamente el resto de comandos que hagan pasar la información de la memoria auxiliar a la central o viceversa. Hasta ese día.

MERCA COMPUTER

Tienda n.º 1
en Amstrad

TAMBIEN COMPATIBLE PC

TODO A LOS MEJORES PRECIOS

464 CPC FN 50.999
6128 F/V 72.999
8256 115.999

MAS IVA

Comandante Zorita, 13 (tienda)
Telf. 253 57 93. 28020 MADRID

OPERACION CAMBIO

— Valoramos tu
AMSTRAD 464 FV en 45.000 Ptas.
AMSTRAD 664 FV en 58.000 Ptas.
en la compra de un nuevo ordenador.

— Se cambia monitor verde por
monitor color nuevo 40.000 Ptas.

TEL. (91) 419 82 72

SERMICRO

SERVICIO TECNICO Amstrad

Barcelona
(93) 307 13 91

Sevilla
(954) 52 56 24

Valencia
(96) 154 11 43

La Coruña
(981) 22 10 12

Bilbao
(94) 444 60 87

Zaragoza
(976) 34 57 22

Málaga
(952) 32 09 44

P. de Mallorca
(971) 23 07 37

SEDE CENTRAL:

Avda. Ramón y Cajal, 107
Tel. 416 80 85

28043 MADRID

SOMOS MAYORISTAS

MICRO-1

EL IVA
LO PAGA MICRO-1

C/ Duque de Sesto, 50. 28009 MADRID
Tel. (91) 275 96 16 - 274 75 02
Metro O'Donnell o Goya (aparcamiento gratuito en Felipe II)

**POR CADA PROGRAMA QUE COMPRES ¡¡GRATIS!!
UNOS CASCOS DE MUSICA ESTEREO
SI TU COMPRA ES SUPERIOR A 800 PTAS.**



SOFTWARE SUPER-REBAJADO
DECATHLON 495 PTAS., BEACH HEAD 495 PTAS.,
SOUTHERN BELLE 495 PTAS., DRAGONTORC 495 PTAS.

	PTAS.
STREET HAWK	2.300
KNIGHT RIDER	2.300
MIAMI VICE	2.300
MOVIE	2.300
EQUINOX	2.100
«V»	2.100
MILLION II DISCO	3.300
GREEN BERET DISCO	3.000
LAS 3 LUCES G DISCO	3.000
DESERT FOX DISCO	3.000
KNIGHT GAMES	2.300
STAINLESS STEEL	2.300
DESERT FOX	2.600
JAK THE NIPPER	2.300
CAULDRON II	2.300
NEXUS DISCO	3.000
RAMBO+MATCH DAY DISCO	3.300
STEINLESS STEEL DISCO	3.000
KNIGHT RIDER DISCO	3.000
KNIGHT GAMES DISCO	3.000

LAPIZ OPTICO
3.295 PTAS.

CASSETTE ESPECIAL ORDENADOR
3.595 PTAS.

OFERTAS EN JOYSTICKS

	PTAS.
QUICK SHOT I	1.395
QUICK SHOT II	1.695
QUICK SHOT IX	2.395

COMPATIBLE PC IBM
MONITOR FOSFORO VERDE
DOS UNIDADES DE DISCO
360 K UNIDAD
UNIDAD CENTRAL CON 256 K
TECLADO EN CASTELLANO
169.900 PTAS.
INCLUIDO PAQUETE DE
SOFTWARE DE GESTION

IMPRESORAS
20% DE
DESCUENTO

PRECIOS EXCEPCIONALES PARA TU AMSTRAD
CPC-464, CPC-6128, PCW-8256, PCW-8512

RATON DE PANTALLA
CON SOFTWARE 7.900 PTAS.
GRATIS UN SUPLETORIO
TELEFONICO



	PTAS.
TAPA DE METACRILATO 464	895
CABLE CENTRONICS	3.175
CABLE SEGUNDA UNIDAD D.	1.790
CABLES SEPARADORES 6128	1.975
INTERFACE RS232	9.265
CINTA VIRGEN C15	69
CABLE AUDIO	795
CABLE ADAPTADOR 2 JOYSTICK	2.390
CABLES SEPARADORES 464	1.390
CABLE SEPARADOR 8256	2.900
CABLE RS232	2.500
DISKETTES 3"	735

PEDIDOS CONTRA REEMBOLSO SIN NINGUN GASTO DE ENVIO. TEL. (91) 275 96 16 - 274 75 02 O ESCRIBIENDO A: MICRO-1. C/ DUQUE DE SESTO, 50. 28009 MADRID.

Tiendas y distribuidores grandes descuentos.
Dirigirse a Dipromisa. C/ Galatea, 25. Tel. (91) 274 75 03.

MEZCLA DE COLORES EN PANTALLA

El Amstrad posee una gran diversidad de colores. Ahora bien, cada uno de ellos se debe utilizar por separado. Os proponemos, pues, intentar mezclar dichos colores para conseguir de esta forma espectaculares efectos.



La forma en que el sistema operativo del ordenador maneja los colores no permite ningún tipo de mezcla con ellos, ya que, a cada interrupción actualiza los colores que se encuentran almacenados en las variables que se utilizan para este propósito.

Así pues, para provocar el efecto de mezcla de colores, deberemos eludir el sistema operativo y activar los colores directamente, enviando los valores correspondientes al periférico encargado de generar el color.

Por lo tanto, deberemos utilizar las instrucciones de entrada y salida que posee el microprocesador Z80, en este caso particular se deberán utilizar instrucciones 'OUT', ya que se desea enviar valores distintos a dicho periférico.

Para ello, en primer lugar debemos averiguar la dirección en la cual se encuentra situado el periférico deseado. Dicha dirección es la &7FXX.

Donde 'XX' será en este caso el número de pluma que se desea activar, o bien poseerá el valor 16 si lo que queremos es dar color al borde de la pantalla.

Veamos cómo podríamos activar un color para una de las plumas, desde Basic:

```
10 OUT &7F00,X
20 OUT &7F00,Y
```

donde 'X' será el número de pluma que se desea activar (16 para el borde), e 'Y' será el color deseado para dicha pluma.

Escribiremos este mismo programa en código máquina, para aquellos que estén interesados en este tema:

```
LD B,#7F
LD C,X
OUT (C),C
LD B,#7F
LD A,Y
OUT (C),A
```

en este caso los parámetros 'X' e 'Y', tendrán el mismo significado que hemos indicado anteriormente.

Debemos tener en cuenta que estos dos programas únicamente producirán un parpadeo del color que hayamos elegido, ya que a la siguiente interrupción que provoque el sistema, éste volverá a actualizar los colores que estaban almacenados anteriormente, que son los que se habían introducido mediante los comandos PEN, PAPER o BORDER.

Para que esto no ocurriera, deberíamos desactivar las interrupciones, lo cual únicamente se podría realizar desde código máquina.

A continuación aparecen listados los distintos valores que se deben introducir en los comandos 'OUT', para conseguir los diferentes colores que posee el Amstrad.

Color	N.º de color	Valor para OUT
Negro	0	&54
Azul	1	&44
Azul brillante	2	&55
Rojo	3	&5C
Magenta	4	&58
Malva	5	&5D
Rojo brillante	6	&4C
Púrpura	7	&45
Mag. brillante	8	&4D
Verde	9	&56
Ciano	10	&46
Azul cielo	11	&57
Amarillo	12	&5E
Blanco	13	&41
Azul pastel	14	&5F
Naranja	15	&4E
Rosa	16	&47
Mag. pastel	17	&4F
Verde brillante	18	&5A
Verde marino	19	&42
Ciano brillante	20	&53
Verde lima	21	&52
Verde pastel	22	&51
Ciano pastel	23	&5B
Amar. brillante	24	&4A
Amar. pastel	25	&43
Blanco brillante	26	&4B





Una vez vista la correspondencia de valores para cada una de las tintas, estamos en condiciones de poder colocar el papel, la tinta y el borde con el color deseado.

De esta forma si deseáramos poner el borde de color negro, la pluma 0 de color rojo y la pluma 1 de color blanco, deberíamos ejecutar lo siguiente:

```
LD BC,#7F00 ;PLUMA 0
OUT (C),C
LD A,#5C ;COLOR ROJO
OUT (C),A
LD C,1; PLUMA 1
OUT (C),C
LD A,# ;COLOR BLANCO
OUT (C),A
LD C,16 ;BORDE
OUT (C),C
LD A,# ;COLOR NEGRO
OUT (C),A
RET
```

Pero como hemos dicho anteriormente, estos colores no permanecerán en pantalla durante largo tiempo, ya que en la próxima interrupción que provoque el sistema, se sustituirán por los almacenados en las variables del sistema correspondiente.

Únicamente lograríamos unos colores permanentes utilizando este sistema, si se desactivarían las interrupciones con la instrucción DI, impidiendo de esta forma que el sistema operativo del ordenador actualizara dichos colores. Hemos visto, hasta el momento, la forma en que podemos utilizar directamente el periférico que posee el **Amstrad** para poner en pantalla los colores deseados.

Veamos ahora cómo provocar una mezcla de colores en cualquier lugar de la pantalla, ya sea en el borde, en el papel o bien en la tinta. Para ello hemos preparado tres programas en código máquina que realizan dichas funciones.

El primero de ellos actúa sobre la pluma 0, que en este caso será el papel o color de fondo de la pantalla, ejecutando previamente el comando:

PAPER 0

Este programa se encarga de mezclar dos colores en pantalla, para ello toma un valor inicial, el #54, correspondiente al color negro, y cada vez que se produce el bucle se produce sobre ese valor la instrucción:

XOR 12

lo que hace que ese mismo valor se convierta en #58; así pues, obtenemos en pantalla los colores negro, correspondiente al valor &54, y magenta, correspondiente al valor &58.

De esta forma se observan en pantalla unas franjas correspondientes a los dos colores mencionados.

La anchura de dichas franjas dependerá únicamente del valor que hayamos introducido

PROGRAMACION

en la pausa. Cuanto mayor sea la pausa, mayor será la anchura de dichas franjas.

Debemos advertir que los tres programas que hemos propuesto, desactivan las interrupciones cuando se entra en ellos, y las habilitan para retornar otra vez al Basic, ya que de otra forma el ordenador se quedaría bloqueado. El programa siguiente, actúa sobre la pluma número 1, por lo tanto para poder observar el efecto que provoca, deberemos ejecutar en primer lugar, el comando:

PEN 1

En este caso se han definido tres valores, los cuales proporcionarán un color distinto cada uno.

El programa toma dichos valores dentro del bucle, y cuando llega al valor 255, vuelve a tomar el primer valor. Así pues, se obtendrán en pantalla tres colores distintos para la pluma 1.

Estos colores serán en este caso, el rojo, amarillo y el rosa, formando franjas. La anchura de estas franjas dependerá, como hemos dicho anteriormente, del valor de la pausa que se haya introducido.

El último de los programas que listamos al final de este artículo, se encarga de proporcionarnos distintos colores para el borde de la pantalla.

Este último programa actúa de una forma similar al programa anterior, ya que toma los valores para los colores, de una tabla, que en este caso también contiene tres colores.

Dichos colores son el negro, amarillo y rojo, formando franjas a través del borde de la pantalla.

Como hemos dicho en los anteriores casos, la anchura de estas franjas dependerá únicamente del valor que se haya indicado en la pausa.

Si deseáramos obtener más de tres colores en cualquiera de los programas mencionados anteriormente, únicamente deberíamos aumentar la tabla de donde se toman dichos valores, teniendo en cuenta que el último valor siempre deberá ser 255, ya que este será el indicador de fin de tabla.

Por último diremos que dando diferentes valores a la pausa y a los colores, podemos conseguir efectos sorprendentes. Actuando sobre la primera, conseguiremos que las franjas sean más estrechas o más anchas, y actuando sobre los segundos, podremos conseguir mezclas de colores realmente asombrosas.

Sólo nos resta decir que para aquellos que no posean ensamblador, ofrecemos un cargador Basic para cada uno de los programas, por lo que únicamente deberéis copiarlo procurando que no dé ningún error.

Una vez hecho esto, podréis salvarlo en cinta o disco para ejecutarlo cuando se desee.

PROGRAMACION

Hisoft GENA3.1 Assembler. Page 1.

Pass 1 errors: 00

```

A000      10      ORG  #A000
A000 F3          20      DI
A001 011027     30      LD  BC,10000
A004 C5          40      BUC: PUSH BC
A005 067F       50      LD  B,#7F
A007 0E00       60      LD  C,0
A009 ED4F       70      OUT (C),C
A00B 01007F    80      LD  BC,#7F00
A00E 3A2CA0    90      LD  A,(COLOR)
A011 EE0C      100     XOR  12
A013 322CA0   110     LD  (COLOR),A
A016 ED79     120     OUT (C),A
A018 CD29A0   130     CALL PAUSA
A01B C1        140     POP  BC
A01D 0B       150     DEC  BC
A01D 7E       160     LD  A,B
A01E B1       170     OR  C
A01F 20E0    180     JR  NZ,BUC
A021 7E       190     EI
A022 C9       200     RET
A023 010000  210     PAUSA: LD  BC,200
A026 0B       220     PAUS: DEC  BC
A027 7E       230     LD  A,B
A028 B1       240     OR  C
A029 20FE    250     JR  NZ,PAUS
A02B C9       260     RET
A02D 54       270     COLOR: DEFB #54
    
```

Pass 2 errors: 00

BUC A004 COLOR A02C PAUS A02E
PAUSA A023

Table used: 58 from 137

Hisoft GENA3.1 Assembler. Page 1.

Pass 1 errors: 00

```

A000      10      ORG  #A000
A000 F3          20      DI
A001 2132A0    30      LD  HL,COLOR
A004 011027     40      LD  BC,10000
A007 C5          50      BUC: PUSH BC
A008 01017F    60      LD  BC,#7F01
A00B ED4F       70      OUT (C),C
A00D 01017F    80      LD  BC,#7F01
A010 7E          90      LD  A,(HL)
A011 FEFF      100     CP  255
    
```

```

A013 2004      110     JR  NZ,SIGUE
A015 2132A0    120     LD  HL,COLOR
A018 7E          130     LD  A,(HL)
A019 F640      140     SIGUE: OR  #40
A01B ED79       150     OUT (C),A
A01D CD29A0    160     CALL PAUSA
A020 23         170     INC  HL
A021 C1         180     POP  BC
A022 0B         190     DEC  BC
A023 7E         200     LD  A,B
A024 B1         210     OR  C
A025 20E0      220     JR  NZ,BUC
A027 FB         230     EI
A028 C9         240     RET
A029 019600    250     PAUSA: LD  BC,150
A02C 0B         260     PAUS: DEC  BC
A02D 7E         270     LD  A,B
A02E B1         280     OR  C
A02F 20FB      290     JR  NZ,PAUS
A031 C9         300     RET
A032 0A070CFF  310     COLOR: DEFB 10,7,12,255
    
```

Pass 2 errors: 00

BUC A007 COLOR A032 PAUS A02C
PAUSA A029 SIGUE A019

Table used: 70 from 145

Hisoft GENA3.1 Assembler. Page 1.

Pass 1 errors: 00

```

A000      10      ORG  #A000
A000 F3          20      DI
A001 2132A0    30      LD  HL,COLOR
A004 011027     40      LD  BC,10000
A007 C5          50      BUC: PUSH BC
A008 01107F    60      LD  BC,#7F10
A00B ED4F       70      OUT (C),C
A00D 01107F    80      LD  BC,#7F10
A010 7E          90      LD  A,(HL)
A011 FEFF      100     CP  255
A013 2004      110     JR  NZ,SIGUE
A015 2132A0    120     LD  HL,COLOR
A018 7E          130     LD  A,(HL)
A019 F640      140     SIGUE: OR  #40
A01B ED79       150     OUT (C),A
A01D CD29A0    160     CALL PAUSA
A020 23         170     INC  HL
A021 C1         180     POP  BC
A022 0B         190     DEC  BC
A023 7E         200     LD  A,B
A024 B1         210     OR  C
A025 20E0      220     JR  NZ,BUC
A027 FB         230     EI
A028 C9         240     RET
A029 016400    250     PAUSA: LD  BC,100
A02C 0B         260     PAUS: DEC  BC
A02D 7E         270     LD  A,B
A02E B1         280     OR  C
    
```

```

A02F 20FB      290     JR  NZ,PAUS
A031 C9         300     RET
A032 140C0AFF  320     COLOR: DEFB 20,12,10,255
    
```

Pass 2 errors: 00

BUC A007 COLOR A032 PAUS A02C
PAUSA A029 SIGUE A019

Table used: 70 from 145

```

10 FOR N=&A000 TO &A02D
20 READ A:SUMA=SUMA+A
30 POKE N,A
40 NEXT
50 IF SUMA<&131E THEN PRINT "ERROR
  EN DATAS"
60 DATA 243,1,16,39,197,6,127
70 DATA 14,8,237,73,1,0,127
80 DATA 58,44,168,238,12,50,44
90 DATA 160,237,121,205,35,160,193
100 DATA 11,120,177,32,227,251,201
110 DATA 1,200,0,11,120,177,32
120 DATA 251,201,84,0,255,84,88
    
```

```

10 FOR N=&A000 TO &A035
20 READ A:SUMA=SUMA+A
30 POKE N,A
40 NEXT
50 IF SUMA<&1704 THEN PRINT "ERROR
  EN DATAS"
60 DATA 243,33,50,160,1,16,39
70 DATA 197,1,1,127,237,73,1
80 DATA 1,127,126,254,255,32,4
90 DATA 33,50,160,126,246,64,237
100 DATA 121,205,41,160,35,193,11
110 DATA 120,177,32,224,251,201,1
120 DATA 150,0,11,120,177,32,251
130 DATA 201,10,7,12,255,75,32
    
```

```

10 FOR N=&A000 TO &A035
20 READ A:SUMA=SUMA+A
30 POKE N,A
40 NEXT
50 IF SUMA<&16FD THEN PRINT "ERROR
  EN DATAS"
60 DATA 243,33,50,160,1,16,39
70 DATA 197,1,16,127,237,73,1
80 DATA 16,127,126,254,255,32,4
90 DATA 33,50,160,126,246,64,237
100 DATA 121,205,41,160,35,193,11
110 DATA 120,177,32,224,251,201,1
120 DATA 100,0,11,120,177,32,251
130 DATA 201,20,12,10,255,75,32
    
```

Correo... más rápido...



Con el fin de acelerar lo más posible el **correo**, y poder resolver o contestar a todas las dudas y sugerencias que llegan a nuestra redacción, a partir de esta semana os rogamos, en beneficio de todos, consignar en el sobre, en lugar bien visible, una de las denominaciones siguientes:

- **Suscripciones AMSTRAD.** Para todos aquellos casos relacionados con petición de cintas, números atrasados, formalización de suscripciones, devoluciones etc...
- **Mercado Común AMSTRAD.** Compras, ventas, intercambios, clubs...
- **Sin duda alguna AMSTRAD.** Para que nos enviéis todas vuestras dudas.
- **Serie Oro AMSTRAD.** Para los programas que nos enviéis para su publicación.
- **Sugerencias AMSTRAD.** Para vuestras críticas, sugerencias o cualquier opinión que queráis vertir sobre la revista.

TU PROGRAMA DE RADIO claro!



ALDISO2

- Entrevistas a fondo
- Exitos en Soft
- Noticias en Hard
- Concursos

Programáteles: Sábados tarde de 5 a 7 horas.
En directo y con tu participación.

LA COPE A TOPE.

— RADIO POPULAR 54 EMISORAS O.M.—

En Barcelona Radio Miramar



PROCESO CONTABLE

DISTRIBUIDOR ZELIG SOFTWARE
AMSTRAD CPC 6128/664/464 DISCO

Víctor Prieto

**El Amstrad 6128
debe tener como una
docena de
programas de
contabilidad en el
mercado, en esta
ocasión echamos un
vistazo al Proceso
contable de Zelig.**

RECIBI





Para... PCW

La contabilidad informatizada es una de las aplicaciones más difundidas en el uso profesional del ordenador, con ella el trabajo de arduas jorandas de lápiz y calculadora, quedan reducidas a una actividad cómoda y ordenada, en la que el ordenador corre con el peso de la ingente cantidad de cálculos rutinarios que requiere el control de una contabilidad.

Nos encontramos ante un programa adaptado al **Plan Contable**, con el cual nuestra contabilidad debe marchar como una seda. El funcionamiento y uso del programa ha sido diseñado pensando en usuarios no expertos en informática, haciendo que este software pueda ser manejado por cualquier experto en contabilidad, sin que tenga necesariamente por qué poseer conocimiento alguno del funcionamiento de los ordenadores.

La versión ante la que nos hallamos, 2.0 Plus, tiene su precedente en el programa de control contable realizado por la misma casa en el año 1985. El ordenador para el cual se realizó fue el **CPC 664**, con la aparición del modelo 128, **Zelig** decidió revisar su software y realizar un producto con mejoras, de esta forma nace este nuevo **Proceso Contable**.

La versión actualizada, aunque realizada para el 128 no presenta ningún tipo de incompatibilidad con el 664, debido principalmente a que este nuevo proceso contable no utiliza el segundo banco de memoria, factor que hace posible dicha compatibilidad.

El texto de presentación con que Zelig nos obsequia en las instrucciones reza: «*La filosofía que inspiró el programa Bookeeper fue la de ofrecer un paquete contable de prestaciones profesionales, cuidada presentación y facilidad de manejo. Confiamos haber cumplido los objetivos*».

Ante tales palabras, no nos queda más que poner en marcha el programa y ver que se encuentra oculto entre los sectores del disco.

Características

- Capacidad:** 200 subcuentas.
- 1.000 apuntes mensuales.
- 100 conceptos automáticos.
- Tipos de informes
- Extractos del diario.
- Extractos de subcuentas.
- Extractos de subcuentas por conceptos.
- Balance de saldos iniciales.
- Balance de situación.
- Cuenta de resultados.
- Listado del Plan Contable.
- Listado de conceptos.

Sólo monitor:

Consulta gráfica de subcuentas por meses.

Iniciación del programa

Como todos los programas de la casa, el **Proceso Contable** ha de ser personalizado con los datos de la empresa o sujeto encargado de la contabilidad, hecho que posibilita la aparición del nombre en los listados, y que es imprescindible para obtener el código de cliente que Zelig exige para la utilización de sus sesiones de trabajo haciendo imposible su uso.

Cumplidos los requisitos de inicialización, ponemos en marcha el programa principal y procedemos a introducir la fecha del día de trabajo, en este momento se nos abren las puertas del proceso contable.

Menú principal

La selección de las distintas opciones se realiza mediante las teclas numéricas, o las de función en la parte derecha del ordenador, presentando la siguiente lista de funciones.

- F1:** *Introducción de apuntes.*
- F2:** *Consultas: por pantalla.*
- F3:** *Consultas: por impresora.*
- F4:** *Representación gráfica.*
- F5:** *Modificación de apuntes.*
- F6:** *Procesos especiales.*

Para proceder a utilizar la opción 1, hemos de dar de alta con anterioridad a las subcuentas que hemos de utilizar, para esta operación ha de usarse la F6.

En ella están contenidas las siguientes operaciones:

- F1:** *Alta de subcuentas.*
- F2:** *Baja de subcuentas.*
- F3:** *Mantenimiento de conceptos.*
- F4:** *Procesos periódicos.*



En el alta de subcuentas, tenemos que introducir el código y nombre de cada una, para el código disponemos de hasta cinco dígitos, de los cuales los tres primeros deben adaptarse al Plan Contable.

Para obtener mayor información acerca de los códigos según los capítulos, podemos consultar el Plan Contable dentro de la opción F2 o F3 que nos dará el listado de todas las subcuentas.

En el campo de nombre, podemos introducir hasta 18 caracteres.

Mediante el mantenimiento de conceptos, tenemos facultad para dar de alta hasta 100 conceptos automáticos, con los cuales operaremos luego en nuestras cuentas.

Los Procesos periódicos permiten realizar:

Cierre anual.

Inicialización del diario.

Inicialización contabilidad.

La posibilidad de Inicialización del diario, viene dada por el límite de capacidad de apuntes, al ser de 1.000 esta cifra no puede ser sobrepasada y hemos de realizar una puesta a cero de todos los apuntes, la cual se realiza por medio de esta opción, una vez realizada la puesta a cero, el libro de apuntes queda vacío, pero manteniendo los saldos de acumulados en las subcuentas.

Introducción de apuntes

La confección de apuntes se realiza utilizando el número de cada subcuenta, de forma que, una vez tecleado éste aparecen en los demás apartados del fichero los datos de nombre de la subcuenta y el concepto.

Con este formato solamente hemos de teclear los dígitos correspondientes al código y luego pasar directamente al Debe o Haber según los casos, en el cual podemos introducir

cifras con el límite máximo de 99 millones de pesetas.

Para efectuar cualquier rectificación, siempre podemos volver al campo anterior hasta que hallamos pulsado el ENTER, en cuyo caso la rectificación ha de realizarse por la opción F5, *Modificación de apuntes*.

Obtención de informes

Los distintos tipos de informes sobre las actividades contables pueden reflejarse o bien por pantalla o por impresora, permitiendo el programa la consulta de los siguientes apartados:

Extractos del diario.

Extractos de subcuentas.

Balance de situación.

Balance de saldos iniciales.

Balance de saldos acumulados.

Cuenta de resultados.

Listado del Plan Contable.

Listado de conceptos.

Abanico de opciones que permite un exhaustivo control escrito de todas las operaciones realizadas, con lo cual se facilita enormemente la introducción de apuntes y demás operaciones contables.

Existe además una posibilidad de consulta sólo por pantalla, con la que podemos observar en un gráfico de barras la evolución por meses de una cuenta determinada.

Ficheros en disco

La grabación de los datos en disco se realiza al final de la utilización de cada opción del menú, de esta forma, se graban en bloque las

distintas subcuentas, por otro lado los conceptos y por otro los apuntes del diario. No es necesario realizar una grabación al final del programa, lo que evita pérdida de datos por olvido o precipitación.

La obtención de copias del programa es posible con utilización del CPM, por lo cual no hemos de temer la pérdida del mismo en caso de accidente.

Documentación

El programa viene acompañado de un completo manual de 40 páginas en el que están reflejadas todas las particularidades del mismo y las operaciones que se han de realizar para su buen aprovechamiento.

En él se describe de forma clara y precisa desde el proceso la carga del programa, hasta la obtención de informes por impresora, pasando por un detallado periplo a través de todas las opciones y submenús contenidos en el mismo.

Conclusiones

Estamos ante un programa de contabilidad ajustado al Plan Contable, un programa de fácil manejo y capacidad suficiente para los usos comunes, con la posibilidad de ser limpiado una vez completado el número tope de apuntes.

La toma de datos del disco hace que al trabajar con un elevado número de subcuentas y apuntes, se observe una ralentización de las operaciones debida al considerable volumen de datos a procesar.

El proceso de inicialización y demás requisitos para su puesta en marcha, hace que sólo se pueda utilizar con un nombre de entidad o profesional, evitando de esta forma y gracias al código secreto que nos es remitido por correo, la circulación de copias piratas.

Como otros programas de este tipo al principio de la carga se nos solicita la fecha del día de trabajo, y sin posibilidad de volver a esta fecha mediante el menú principal, lo cual nos obliga a realizar diversas cargas del programa en el caso en que tengamos que actualizar en una sola sesión el proceso contable de varios días.

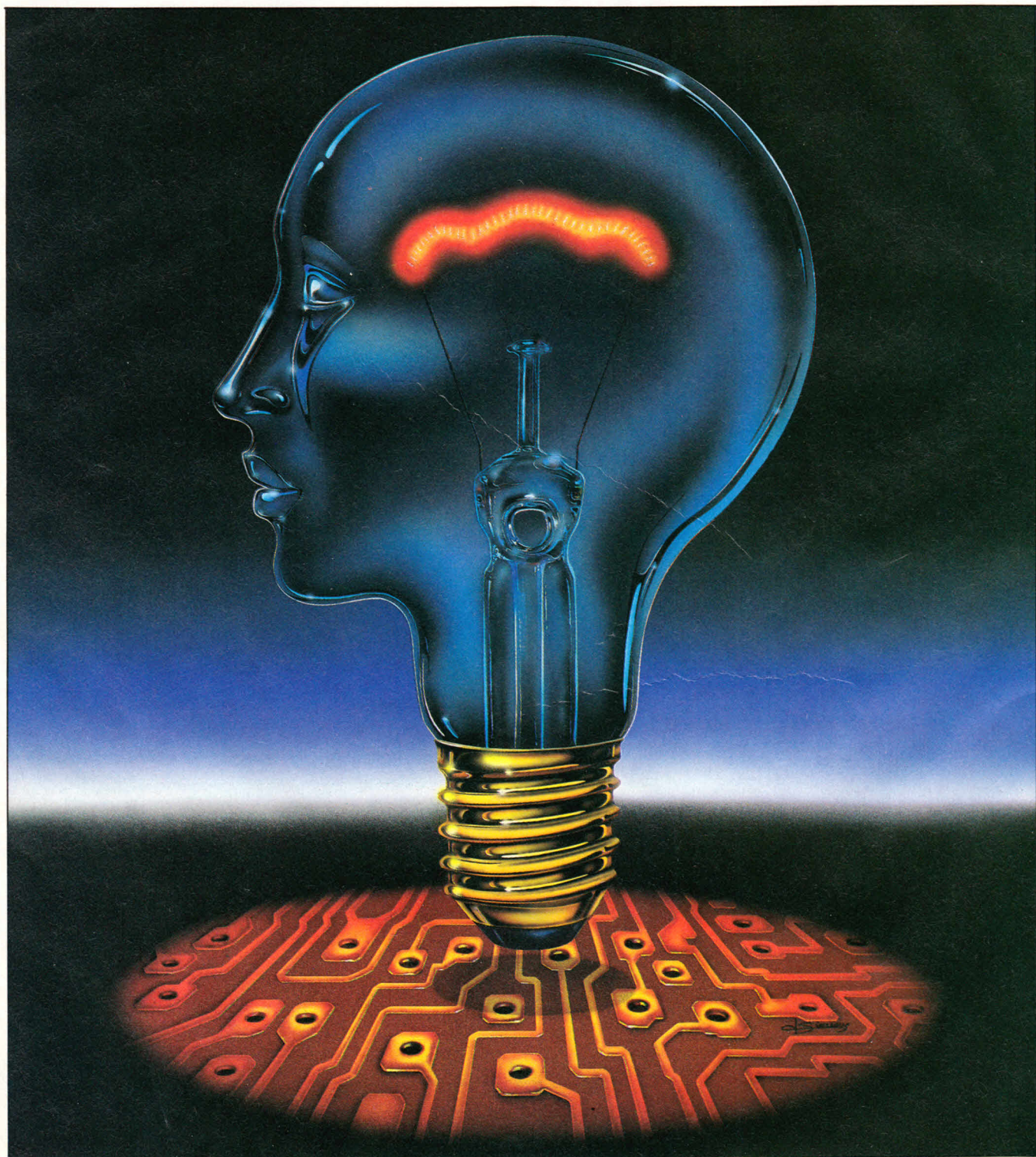
La gama de informes que podemos sacar por impresora, es amplia y no deja tema por tocar, constituyendo, sin duda la mejor calidad de esta pieza de software con capacidad de 200 cuentas y 1.000 apuntes.

Este programa viene acompañado de otro para llevar el IVA, con el que podemos manejar hasta un máximo de 150 cuentas entre clientes y proveedores, permitiendo realizar un total de 2.000 apuntes.



MONITOR DE ROM's

Mediante el pequeño, pero potente programa, que hoy ofrecemos a los programadores, tenemos la posibilidad de curiosear en el interior de las ROM's incorporadas a nuestro CPC. Ya sea en las que vienen de fábrica, la del BASIC y CPM, o en las que hayamos incorporado nosotros mediante el bus trasero de nuestro ordenador.





O sé si os pasa igual a vosotros, pero a mí los manuales de las «cosas» que se conectan al ordenador me son inútiles. Nunca los necesito excepto cuando no los encuentro; y claro, nunca los encuentro excepto cuando no los necesito.

Conectados al pequeño CPC 464 de nuestra casa tenemos un módem, un sintetizador de voz, alarmas, un ensamblador, bancos de memoria, y..., vaya, no recuerdo cómo se llama porque he perdido el manual.

Cada uno tiene su ROM con sus comandos externos, que como DIR, se escriben empezando con la barra vertical.

En total 50 ó 60 comandos de entre los que alguno hace que me pierda en detalles de ortografía como SETFILEEND.

Muy bien. Por lo menos no he perdido el manual del Firmware, ya que éste tiene las rutinas para seleccionar y encender las ROM.

Y es aquí donde el presente programita aparece para ayudarnos a desvelar los secretos de Read Only Memory (*Memoria*). Sólo leer en indio.

La familia de los CPC está dotada de un mínimo de 32 Ks de ROM. Una parte, 16 Ks de esta memoria, solapa la parte inferior de la memoria. También es llamada firmware, y su dirección va de la 0 a la 3FFF.

Su contenido lo forman las rutinas pregrabadas disponibles para todos los demás programas, incluyendo el Basic. De esta parte no hablaremos más. La que aquí veremos es la otra, que solapa en memoria los 16 Ks más altos (*upper, superior*). Aquí se puede expandir la memoria hasta 4 Mbytes (*si, eso es 4 Megabytes*) en 250-y-pico bloques de 16 Ks.

Si tienes un miembro de la familia CPC con disco, por ejemplo, ya hay uno de estos bloques ocupado por el programa que contiene los sistemas operativos AMSDOS y CPM.

Todo el mundo tiene su Basic (*que no es más que un simple programa pregrabado —pero ¡vaya programa!—*) en una ROM dentro del ordenador. Nuestra máquina tiene siete ROMs conectados, unas 100 Ks.

Utilidades del PROGRAMADOR

MONITOR D CB

```
1 ; *****
1 ; *****
2 ; *
3 ; * ROM Monitor *
4 ; *
5 ; *****
6
7 ORG #3D00 ;Microhobby Amstrad 4.xi.1986
8
9 comand: CALL SETROM ;call comand,@string$,rom
10 JR RDCMD ;devuelve chr(255) si acabado
11
12
13 SETROM: LD C,(IX+0) ;rom seleccion
14 CALL #B90F ;kl_rom_select
15 JP #B900 ;kl_u_rom_enable
16
17 title: CALL SETROM ;call routine,@string$,rom
18 LD HL,(#C004) ;copiar titulo de ROM a (DE)
19 LD (pointr),HL ;inicializa puntero
20 RDCMD: CALL STRING ;DE=@string$
21 LD HL,(pointr) ;puntero al proximo nombre
22 LD A,(HL) ;busca null-terminator
23 AND A ;es cero?
24 JR NZ,RDCMD1 ;sigue
25 LD A,255 ;bandera de terminacion
26 LD (DE),A ;resultado para Basic
27 JR romoff ;hecho
28 RDCMD1: LD A,(HL)
29 AND #7F ;ignorar bit-7
30 CP #20 ;ignorar codigos de ctrl
31 JR C,RDCMD2
32 LD (DE),A ;construir string$
33 INC DE
34 RDCMD2: BIT 7,(HL) ;ultimo tiene bit-7 alzado
35 INC HL
36 JR Z,RDCMD1 ;bit-7 no alzado
37 LD (pointr),HL ;para la proxima vez
38 romoff: XOR A
39 LD C,A ;selecciona Basic
40 JP #B90F ;kl_rom_select
41
42 STRING: LD L,(IX+2) ;devuelve DE=puntero a string$
43 LD H,(IX+3)
44 INC HL ;puntero
45 LD E,(HL)
46 INC HL
47 LD D,(HL)
48 RET
49
50 copyit: CALL SETROM ;call copyit,rom [activar rom]
51 XOR A ;copiar #4000 bytes del banco
52 LD C,A ;#C000 al banco #4000
53 LD E,A
54 LD L,A
55 LD H,#C0
56 LD D,#40
57 LD B,D
58 LDIR
59 JR romoff
60
61 pointr: DEFB "*","*" ;puntero al proximo nombre
62
63 ; ***** fin *****
```

Teclea el programa en Basic. La primera parte es Código Máquina con un chequeo al final de cada línea para captar los fatales errores de tecleo.

No es necesario teclear el código ensamblador fuente. Es más rápido y más seguro teclear las líneas DATA en el programa Basic y usar el listado de tu revista como referencia.

Si tienes 4 Mbytes de ROM conectados o, por lo menos, si tienes más de 7 ROMs, tendrás que cambiar la línea 110, poniendo el número del último ROM conectado. (En verdad, no conocemos a nadie con más de siete).

Ahora RUN. Primero el programa proporciona un catálogo de las ROMs con sus títulos internos, que son comandos externos normalmente no accesibles desde Basic. Ahora elige uno, por ejemplo el número cero, el Basic. Esta ROM está encendida y copiada a RAM (memoria normal de escritura/lectura) en las direcciones 4000-8000 hexadecimales. Se muestra un directorio de los comandos externos disponibles en esta ROM. El directorio de Basic estará vacío, porque los comandos de Basic son todos internos a este programa. El de CPM-AMSDOS tendrá ITAPE, IDISC, IERA, etcétera.

Finalmente, existe la opción de grabar los 16 Kbytes de la ROM en disco o cinta. Será un fichero binario. Este fichero se puede estudiar trozo a trozo con un programa monitor, para ver, por ejemplo, cómo funciona el Basic, o cómo es el bloque de datos de los discos en el BIOS de AMSDOS/CPM. Adiós misterios.

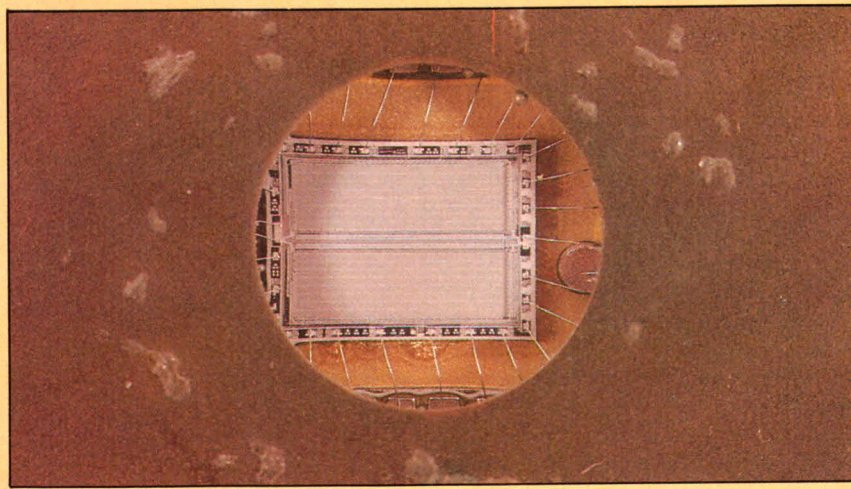
Yo tengo normalmente residente en memoria el programa HOJEAMEN que publicaba **AMSTRAD SEMANAL** en su edición especial n.º 2. Con este, y después de ejecutar el presente programa, se puede hojear rápidamente los 16 K de la ROM.

Cuando lo hago con ROM cero, encuentro el famoso READY del Basic ya en la tercera línea, y en la segunda página los mensajes de error, empezando con «Unknown error» (Error desconocido). Disfruta.

```

10 DATA C0853D1812DD4E00CD0FB9C300B9C
D053D2A04C022573DCD3C0931
15 DATA 3D2A573D7EA720053EFF1218117EE
67FFE2038021213CB7E230889
20 DATA 28F222573DAF4FC30FB9DD6E02DD6
603235E2356C9CD053DAF0A6D
25 DATA 4F5F6F26C0164042EDB018E02A2A0
584
30  COMMAND=&3D00
32  DIREC=COMMAND-1: IF HIMEM>DIREC TH
EN MEMORY DIREC
35  'ROUTINA CARGA DATOS CON CHECKSUM
40  FOR LINEA=10 TO 25 STEP 5
45  READ POSTIZO$
50  CHECK.SUM=VAL("&" + RIGHT$(POSTIZ
0$,4))
55  FOR PUNTERO=1 TO LEN(POSTIZO$)-
5 STEP 2
60  DUMMY=VAL("&" + MID$(POSTIZO$,P
UNTERO,2))
65  CHECK.SUM=CHECK.SUM-DUMMY
70  DIREC=DIREC+1
75  POKE DIREC,DUMMY
80  NEXT PUNTERO
85  IF CHECK.SUM<>0 THEN GOTO 10000
90  NEXT LINEA
95  'CATALOGO DE ROMS
100 'CATALOGO DE ROMS
102 TITLE=COMMAND+&E
103 COPYIT=COMMAND+&47
105 COPYIT=COMMAND+&47
110 ULT.ROM=7
115 MODE 2
120 FOR R=0 TO ULT.ROM
130 GOSUB 1000'TITLES
170 NEXT R:PRINT
180 GOSUB 2000'ELIGE
190 CALL COPYIT,R'ROM TO &4000
200 GOSUB 3000'PIDE FILENAME Y SAVE
210 GOSUB 4000'ADIOS
999 END
1000 'PRINT ROM TITLES
1004 A$=SPACE$(16)
1005 CALL TITLE,2A$,R
1006 IF R<>0 AND A$="BASIC"+SPACE$(16
-5) THEN RETURN
1010 PRINT "-ROM NUMERO" USING "###
&";R,A$
1020 RETURN
2000 'ELIGE ROM
2010 LINE INPUT "ELIGE NUMERO DE ROM
[ENTER PARA TERMINAR PROGRAMA YA ] --
>";R$
2020 IF R$="" THEN END
2030 R=VAL(R$):IF R<0 OR R>ULT.ROM GO
TO 2010
2040 GOSUB 1000'REPITE TITLE
2045 GOSUB 5000'PRINT EXT COMMANDS
2999 RETURN
3000 'SAVE
3010 LINE INPUT "FILENAME PARA GRABAR
17K EN DISC/CINTA [ENTER = TERMINAR
] -->";R$
3020 IF R$="" THEN RETURN
3030 R$=R$
3040 SAVE R$,B,&4000,&4000
3050 RETURN
4000 'BYE
4010 PRINT "LOS BYTES DEL ROM TODAVIA
ESTA EN &4000-&7FFF"
4050 RETURN
5000 'PRINT ROM EXT COMMANDS ( SALTAN
DO CODIGOS CTRL )
5001 PRINT "-COMANDOS EXTERNOS DE ESE
ROM:"
5003 GOSUB 5060'LINEA
5005 WHILE 1=1
5010 A$=SPACE$(20)
5020 CALL COMMAND,2A$,R
5030 IF LEFT$(A$,1)=CHR$(255) GOTO
5060'RETURN
5040 IF A$<>SPACE$(20) THEN PRINT
A$;
5050 WEND
5060 PRINT STRING$(80,"-")
5070 RETURN
10000 MODE 2:PRINT CHR$(7) "ERROR EN"
LINEA:LIST-25

```



Mercado común

Con el objeto de fomentar las relaciones entre los usuarios de AMSTRAD, **MERCADO COMUN** te ofrece sus páginas para publicar los pequeños anuncios que relacionados con el ordenador y su mundo se ajusten al formato indicado a continuación.

En **MERCADO COMUN** tienen cabida, anuncios de ventas, compras, clubs de usuarios de AMSTRAD, programadores, y en general cualquier clase de anuncio que pueda servir de utilidad a nuestros lectores.

Envíanos tu anuncio mecanografiado a: **HOBBY PRESS, S.A.**
AMSTRAD SEMANAL.

Apartado de correos 54.062
28080 MADRID

¡ABSTENERSE PIRATAS!

Vendo monitor color **Amstrad** CTM 640 por 40.000 pts. Tel. 242 04 29. Llamar de 9 a 12. Preguntar por Angel. (Barcelona).

Vendo Amstrad 464 con monitor a color y 25.000 pts. en programas. Todo por 75.000 pts. José L. Rojo Matarranz. C/ Tórtola, 9, izq. 47012 (Valladolid). Tel. (983) 29 41 04 de 2 a 5 de la tarde.

Vendo ratón AMX MOUSE para **Amstrad** CPC-464 comprado en Semana Santa, precio: 12.500 pts. Alexis Gutiérrez. C/ Gutiérrez Rada, Laredo (Cantabria). Tel. (942) 60 62 25.

Vendo Amstrad CPC-464 (¡Oye, está nuevo!) monitor fósforo verde, con sus 8 programas de obsequio, al igual que dos manuales. Precio 45.000 pts. Interesados llamar al Tel. (96) 278 31 49. Liria (Valencia), a partir de las 19,00 h. Preguntar por Miguel Bori.

Urge vender **Amstrad** 464 monitor color en perfecto estado por cambio de ordenador. También estarían en el lote unos 25 ó 30 programas como procesador de textos, Sorcery, Roland ropes, La pulga y un joystick Quick Shot. Todo por 70.000 pts. Julián o Javier Rus. C/ Pío Gullón, 17. Astorga (León). Tel. (987) 61 53 74, llamar de 14,00 a 15,00.

SOFTWARE GESTION COMERCIAL

- **ALSICONT** (Contabilidad general adaptada P.G.C.) Disponible para Amstrad PCW 8512 y PCW 8256, IBM PC/XT/AT y compatibles, INVES PC, ATARI ST 520 y 1040, SINCLAIR QL.
- **COMERCIAL 6** (Facturación, Control de stocks, Fichero de direcciones) Disponible para AMSTRAD PCW 8512 y PCW 8256, IBM PC/XT/AT y compatibles, INVES PC, SINCLAIR QL.
- **ALSISTOCKS** (Control de stocks) Disponible para AMSTRAD PCW 8512 y PCW 8256, IBM PC/XT/AT y compatibles, INVES PC, SINCLAIR QL.
- **ALSIMAIL** (Fichero de direcciones, impresión de recibos) Disponible para IBM PC/XT/AT y compatibles, INVES PC y SINCLAIR QL.
- **ALSIFIN** (Simulador financiero, Estadística financiera, Matemática financiera) Disponible para Amstrad PCW 8512 y PCW 8256, IBM PC/XT/AT y compatibles, INVES PC y SINCLAIR QL.

ALSI COMERCIAL, S. A.

NICOLAS USERA, 10
TELF. 475 43 39. 28026 MADRID

Desearía comprar modulador de televisión color para **Amstrad** 464. Interesados llamar al Tel. 33 12 40 de Málaga. Preguntar por José Bernardo.

Vendo juegos compatibles para el **Amstrad** CPC-472. Yie ar Kung Fu, Roland Ropes, Ajedrez 3D voice, Kinght Lore, etc. Interesados llamar al tel. (954) 64 04 74. Preguntar por Antonio o escribir a Antonio Navas Caro. C/ Santuario de las Cabezas, 83, 3-E. 41006 (Sevilla). Desearía contactar con usuarios del **Amstrad** para cambiar juegos etc.

Desearía contactar con usuarios de **Amstrad** CPC-464, CPC-6128 para el intercambio de juegos y utilidades. Enviad lista. Contestaré todas las cartas. Javier T. Tejada. C/ V. Garamendi, 5, 7-B. 48006 Bilbao (Vizcaya).

Intercambio juegos del CPC-464 con usuarios de dicho ordenador. Poseo juegos y más o menos son 150 como los que siguen: Fred, Alien. Simulador de Vuelo, Fighter Pilot, Knight Lore y muchísimos más. Me gustaría que alguien me dejara Decathlon. Os aseguro que os los dejaré mucho tiempo. Llamar los martes y jueves de 6 a 9 h. Preguntar por Andrés Andrade. C/ Meliana, Patio 5ª pta. 13. (Valencia) 46019. Tel. (96) 365 50 36.

Vendo n.º 1 al 30 de Microhobby Spectrum por 2.700 pts. No ejempleres suelto. Sin gastos de envío. Llamar a partir de las 7 h. de la tarde. Miguel Angel Godoy Rubio C/ Itálica, 163. Santiponce (Sevilla). Tel. (954) 39 64 25.

Vendo Amstrad 464, color, en perfecto estado. Tel. (974) 20 53 32. Preguntar por Alfonso, de 9 a 10,30 noche. (Burgos).

Programador y poseedor de un **Amstrad** 8256. desearía contactar con usuarios de dicho equipo, para intercambio de programas. Realizo programas a la medida. Por encargo, (programs de gestión). Contactar con Francisco Moreno Monteverde. Avda. José Fariñas, 68 5.º B. 21006 (Huelva).

Sin duda alguna

A través de esta sección se pretende resolver, en la medida de lo posible, todas las posibles dudas que «atormenten» a todas las personas interesadas en el mundo del AMSTRAD, sean o no poseedores de uno y, si lo son, se encuentren en cualquier nivel de destreza en su manejo.

Semanalmente, aparecen en estas páginas las consultas de la mayor cantidad de usuarios posible; ello redundará en un mejor servicio y en un contacto más estrecho entre todos nosotros a través de la revista.

SIN DUDA ALGUNA está abierta a todos.

¿OTRO SALUDO?

Me dirijo a Vds. con el fin de aclarar «una terrible duda» relacionada con el ordenador **Amstrad CPC 464**.

Se trata de lo siguiente: Al yo comprar el ordenador, recibí un par de manuales; pero en ninguno de ellos habla de cómo se puede cambiar en la memoria interna el **saludo** que aparece al encender el ordenador.

Quisiera que me dijeran si es posible cambiarlo, y si así es, la forma de hacerlo.

Carlos Abehsera Davó

Un ordenador tiene dos «tipos» de memoria: la RAM y la ROM. En la primera, usted puede modificarla escribiendo y leyendo datos en ella, mediante, por ejemplo, las órdenes PEEK y POKE (vea manual). La segunda es de una manipulación más comprometida. Por los métodos normales, sólo se puede averiguar lo que tiene (PEEK), pero no se puede alterar su contenido, escribir en ella

(POKE). Como ya se habrá imaginado, el mensaje del que usted nos habla está en ROM, y no se puede cambiar, como hemos dicho antes, por métodos normales.

Los métodos «anormales», que sí le permitirían alterar la memoria ROM del **Amstrad**, requieren una serie de aparatos relativamente complejos, y caros: un grabador de EPROM y las memorias EPROM.

EPROM significa ROM programable: usted puede leer y escribir en ella. Para poder hacerlo, debe:

— Escribir un programa que sustituya, en todo o en parte, a la propia ROM del Amstrad.

— Sacar la memoria ROM del **Amstrad**, la que viene de fábrica, y sustituirla por la suya.

Ni que decir tiene que realizar esta tarea requiere unos conocimientos muy fuertes de lenguaje máquina.

CINTA DEFECTUOSA

Soy propietario desde hace poco de un **Amstrad CPC 6128** y debo de advertirles que estoy empezando en esto de la informática.

Recientemente he adquirido un cassette de la serie Oro de Microhobby **Amstrad** el n.º 10, y tengo que hacerles la siguiente consulta a raíz de unos problemas que me han surgido.

He seguido correctamente las instrucciones de carga y no he tenido problemas con los programas de una de las caras de la cinta que son: «Amscal», «Ejecutivo», «Ram disc 128», «Compant», «Análisis» 37-39-40, «Ideas» 37-38-39. Sin embargo con los restantes programas de la otra cara, no he podido cargarlos porque enseguida me salía en pantalla «Read error b». He probado con diversos volúmenes y no ha habido manera.

Les rogaría me dijeran si hay alguna solución o es que es problema de la cinta, ya que en ese caso se la cambiaría para que me la cambiara.

Angel Alarcón Villanueva

Lo más probable es que la cara B de la cinta se encuentre mal grabada. La solución más corta es que, en efecto, nos la mande por correo.

Nosotros le enviaremos otra completamente gratis a la mayor brevedad. Le rogamos que disculpe las posibles molestias.

¿QUE CONTROLA EL CONTROLADOR?

Me dirijo a Vd. para que me resuelva unas dudas que tengo: Poseo un **CPC 464** y se que se le puede conectar una unidad de disco, pero para ello, ¿es necesario utilizar otro interface distinto al que ya tiene el ordenador? ¿Para qué sirve un controlador de la unidad de disco? ¿Se pueden utilizar los lenguajes LOGO y COBOL en mi máquina, y de ser así cómo se podría hacer?

Gracias.

Daniel Ramos Ruiz

a) La pregunta es a medias. El 464 posee un bus de expansión», esto es, un lugar donde pueden conectarse todo tipo de periféricos. El disco entra dentro de ellos, pero necesita un interface que se conecta al ordenador a través del bus de expansión y de él al disco. En cualquier caso, la unidad de disco y la interface vienen juntas; al comprar una le dan la otra. ¡Ha habido suerte!

c) El controlador de la unidad de disco es precisamente la interface de que le hablábamos en el apartado a).

Su función es dominar al disco de forma que el ordenador se entere lo menos posible, o al menos esa debiera ser. En el caso del Amstrad, sin embargo, el controlador se limita a incorporar un programa, escrito en ROM (memoria de sólo lectura) que permite al disco cumplir su compleja misión.

c) En su máquina se puede usar cualquier lenguaje que se ejecute bajo su sistema operativo, en este caso el CP/M.

Los lenguajes por los que pregunta existen en versión comercial: el Logo se lo dan al comprar el disco, y un compilador de Cobol lo comercializa Microbyte.

En el caso de este último, mucho nos tememos que el 464 se le quede algo corto, pero la gente que se lo venda sin duda le podrá asesorar con todas las garantías.

COMPUTER ENGLISH SYSTEM

EL METODO INTEGRAL MAS FACIL Y DIVERTIDO PARA APRENDER INGLES

CONTROLADO POR PROFESOR



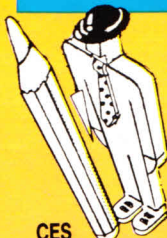
El Curso se compone de cinco discos **FULL-MEMORY**, que incluyen un Test de Nivel para evaluar los conocimientos del alumno, sesenta unidades completas (con tres tipos distintos de ejercicios) de dificultad progresiva, diez unidades especiales que ahondan en los aspectos más esenciales del idioma (con un juego cada una), un programa de Redacción controlada y libre, y un disco para prácticas de conversación que permite dialogar con el ordenador sobre diversos temas en lengua inglesa.

Todos los discos poseen un **PLANNING** para que el alumno organice su propio ritmo de trabajo y puede controlar sus progresos.

En resumen, **1 MEGABYTE** de contenidos y programas configuran **COMPUTER ENGLISH SYSTEM**, enfocados para ayudarnos a aprender o reforzar nuestro **INGLES** de forma estimulante y sugestiva.

CENTRAL: ZARAGOZA: CAMBRIDGE SCHOOL. Zurita, 21, 1.º izq. - Tel. 23 91 92 ● **CENTROS DISTRIBUIDORES:** BARCELONA: BARNA HOUSE. Rambla Catalunya, 112, 1.º Tel. 93-2187 846 ● PAMPLONA: THE ENGLISH FILM CENTRE. Duque de Ahumada, 3, 2.º Tel. 22 41 84
CENTROS COLABORADORES: BARCELONA: NOVALINGUA. Avda. Diagonal, 600 ● SAXON SCHOOL. Gran Vía de las Cortes Catalanas, 437

BOLETIN DE PEDIDO



CES
COMPUTER ENGLISH SYSTEM
ZURITA 21, 1º izq. • ZARAGOZA

NOMBRE _____ EDAD _____
DIRECCION _____ TEL. _____ C. POSTAL _____
POBLACION _____ PROV. _____ PROFESION _____
 DISCO 1 - 4.900 Pts. I.V.A. Incluido DISCO 2 - 4.900 Pts. I.V.A. Incluido DISCO 3 - 4.900 Pts. I.V.A. Incluido
 DISCO 4 - 4.900 Pts. I.V.A. Incluido DISCO 5 - 4.900 Pts. I.V.A. Incluido CURSO COMPLETO - 21.600 Pts. I.V.A. Incluido

FORMA DE PAGO: TALON BANCARIO A CES. COMPUTER ENGLISH SYSTEM ●
 GIRO POSTAL A CES. COMPUTER ENGLISH SYSTEM ● CONTRA REEMBOLSO MAS GASTOS

DESEO RECIBIR EN MI DOMICILIO LOS PROGRAMAS SEÑALADOS

DRAGON'S LAIR[®]

OWNED BY MAGICOM INC. AND USED BY PERMISSION

¡¡ EL JUEGO DE LAS MAQUINAS QUE SORPRENDIO A TODOS!!

... Por su parecido a una película de dibujos animados. En tu intento de rescatar a la bella Princesa de las garras del diabólico Dragón, tú, nuestro héroe, has de abrirte paso a través del castillo encantado por el Mago Negro.

Muchos son los peligros que te acechan y pocas tus defensas, sólo tu espada y tu habilidad...

¡ADELANTE AVENTURERO, EL DESAFIO TE ESPERA!



OTRO EXITO DE

ERBE
Software

SOFTWARE
PROJECTS



DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA: ERBE SOFTWARE. C/. STA. ENGRACIA, 17
28010 MADRID. TEL. (91) 447 34 10 - DELEGACION BARCELONA, AVDA. MISTRAL, N.º 10 - TEL. (93) 432 07 31