



LA PROTECCION DEL SOFTWARE



ICE: Iconos, ventanas y ratones



La protección del software

Debido a que la entrada y salida de datos bajo el sistema operativo QDOS es un dispositivo independiente, la protección de programas resulta algo problemática. Cualquier archivo, esté grabado en *diskette* o *microdrive*, puede copiarse fácilmente en una impresora, pantalla o cualquier otro *diskette* o *microdrive*. Esto es particularmente cierto para los programas SuperBASIC que se han almacenado como archivos en forma de caracteres ASCII, ya que el comando COPY *mdvLnombre_de_archivo* TO scr revelará al instante toda la información de forma totalmente legible.

En cuanto al código máquina, aunque éste necesita desensamblarse, el hecho de que el QDOS permita extraer *bytes* de cualquier lugar de la memoria mediante el comando SBYTES y volverlos a cargar con LBYTES en otra localización totalmente diferente, facilita la desprotección.

Por lo tanto, para proteger un programa de QL deben idearse métodos especiales, para que solamente funcione cuanto esté en su soporte original o para hacer el proceso de copia lo más complejo posible. En este artículo pasaremos revista a algunos métodos disponibles para el usuario.

RUTINA DE CARGA AUTOMÁTICA

La carga de programas comerciales desde el soporte en que han sido grabados (cartuchos *microdrive*) a la memoria de modo que funcionen automáticamente, se realiza mediante un archivo BOOT.

En el QL, inmediatamente después de pulsar una de las teclas de función para seleccionar el forma-

La protección de programas tiene un significado distinto para cada uno de los diferentes tipos de usuarios existentes. En la escala superior, las casas de software emplean medidas de protección para mantener el fruto de costosas inversiones fuera del alcance de la explotación criminal de terceras personas. Para el programador, es un medio de mantener en secreto el resultado de muchas horas de duro trabajo desarrollando revolucionarios algoritmos. En cuanto al hombre de negocios, significa impedir el acceso a información reservada. Finalmente, puede ser también, para el simple aficionado, un medio eficiente de cobertura de una mala programación. En este artículo, vamos a revisar algunos métodos para proteger sus propios programas.

to de pantalla, la máquina intenta abrir un canal para un dispositivo llamado BOOT, consistente en una lista de comandos almacenados en ROM. Si falla el intento, se prueba a abrir un archivo de entrada BOOT que esté almacenado en el *microdrive* 1. Este archivo, formado por una serie de líneas BASIC en ASCII, se cargará normalmente en la mayoría de los programas SuperBASIC y/o código máquina (incluyendo una sentencia CALL para estos últimos).

Este mecanismo gira en torno a la variable del sistema BV_COMCH, en la dirección

S0084(A6), la cual contiene una palabra de información con el canal ID para el comando de entrada de flujo. Normalmente dicho contenido es cero para el canal de entrada del teclado, pero al producirse una carga automática (BOOT) o al entrar en funcionamiento el comando LOAD, contendrá el canal ID para archivo abierto.

En lugar de seguir el procedimiento de carga automática descrito más arriba, el pirata podría introducir esta orden: LOAD *mdvL_boot*, e inmediatamente listar el programa para descubrir más información. Pero como este archivo es una lista de comandos que no necesitan números de línea, se autoejecutará siempre aunque se emplee LOAD o MERGE.

El listado 1 es una demostración de cómo establecer un archivo de ejecución automática (BOOT). Una vez grabado, intente cargar el programa con LOAD o MERGE y pulse BREAK a continuación (CTRL + SPACE). Seguidamente, introduzca el comando LIST y descubrirá que el programa ha desaparecido, ya que no es un programa propiamente dicho, sino una lista de comandos directos provenientes del cartucho de *microdrive*. Este método puede usarse para cualquier programa SuperBASIC (no necesita llamarse BOOT ni tampoco tiene por qué estar forzosamente almacenado en el *microdrive* 1).

INVISIBILIZANDO ARCHIVOS

Aunque el método anterior puede hacer desistir temporalmente al pirata, todavía puede estar en condiciones de utilizar el comando COPY. En este caso deberemos fabricar un archivo invisible, de



modo que no aparezca en el directorio. Un ejemplo de utilización de este método es SBYTES mdvL_0,49152. En este caso, grabaríamos el contenido de la ROM en el *microdrive* 1 con un nombre de archivo nulo. Al efectuar un DIR mdvL_, el directorio no muestra ningún nombre de archivo, pero el número de sectores libres se verá sensiblemente reducido.

Como alternativa, también puede almacenar archivo con nombres como mdvL_«ESPACIO». Estos no pueden copiarse porque el QL no puede trabajar si tras el subrayado encuentra un carácter de espaciado.

DESACTIVACION DE BREAK

Uno de los métodos más conocidos para evitar la piratería consiste en inhabilitar la acción de BREAK.

El teclado sufre una exploración cada 20 milisegundos debido a las interrupciones generadas por una de las ULAs del QL. Afortunadamente, este mecanismo de interrupción está condicionado por el puntero de una tabla de datos situado en SV_PLIST (en la dirección S2803C). Esta palabra apunta a una lista de 8 bytes. Los primeros 4 bytes son el puntero del siguiente elemento de la tabla de datos. Los otros 4 bytes contienen el puntero del mecanismo de interrupción de 50 Hz que debe ejecutarse en ese momento. Al encenderse, el QL sin expansión sólo tiene una entrada en la tabla, consistente en una rutina (llamada cada vez que se produce una interrupción) que trabaja con todos los *interfaces* externos del QL mediante la comunicación con las ULAs principales y el IPC (*Intelligent Peripheral Controller*).

Para desactivar BREAK desde el SuperBASIC solamente debe introducirse el siguiente comando: POKE_L 163900,0.

Al colocar un 0 en esa dirección de memoria, el QL ignora las tareas externas por estar vacía la tabla. Esto afectará a cualquier *hardware* extra que se esté utilizando y emplee interrupciones de 50 Hz.

LISTADO 1

```
100 OPEN #4,"mdv1.boot"
140 PRINT #4,"Repeat loop:"
150 PRINT #4,"  FAKER MOVING TO 7111:"
160 PRINT #4,"  PRINT Try pressing BREAK!"
170 PRINT #4,"END REPEAT loop"
190 CLOSE #4
```

LISTADO 2

```
MT_LBCHD EQU $1E      INTERRUCCION #1
BASIC_HEADER EQU $6B  TAMANO CABECERA CARGADOR SB
BV_BRK EQU $00F
JB_START EQU $0014
SV_BASIC EQU $2B010   DIRECCION CABECERA SB
SV_SCRST EQU $2B033   DIRECCION FLAG ESTADO PANT.
WAITING EQU -2
```

* COMIENZO DEL CODIGO

```
LEA TASK,A1          PUNTERO PARA LA NUEVA TAREA
                     DE PLANIFICACION
LEA TASK_LINK,A0    PUNTERO PARA RESERVA DE ESP.
MOVE.L A1,A7(A0)    ALMACENA PUNTERO DE TAREA
MOVED *MT_LBCHD,D0 INTERRUCCION
TRAP #1
RTS                 RETORNO AL SB
```

* TAREA DE PLANIFICACION A LLEVAR A CABO

```
TASK MOVE.L SV_BASIC,A0 ESTABLECE DIRECCION DE LA
                     CABECERA DE LA RUTINA SB
LEA BASIC_HEADER(A0),A1
TAB BV_BRK(A1)
TST.W JB_START(A0)   EXAMINA ESTADO TAREA SB
SGE.S NO_ALTER       NO ALTERAR SI SUPERBASIC NO
                     ESTA SUSPENDIDO
MOVE *WAITING,JB_START(A0)
NO ALTER CLR.B SV_SCRST SE ASEGURA QUE LA PANTALLA
                     ESTE ACTIVA
RTS
```

* RESERVA ESPACIO DE TRABAJO

```
DB.L 2
END
```

LISTADO 3

```
130 LET reserved_address=RDCHI (60)
140 GET address=reserved_address
150 RESTORE
160 REPEAT read_and_store_machine_code
170 IF END THEN EXIT read_and_store_machine_code
180 READ machine_code_byte
190 POKE address,machine_code_byte
200 LET address=address+1
210 END REPEAT read_and_store_machine_code
220 CALL reserved_address
230 PRINT #0,"Now try pressing break..."
240 REPEAT forever
250 FOR border_width=97 TO 0 STEP -4
260 BORDER border_width,RND(0 TO 75)
270 END FOR border_width
280 END REPEAT forever
290 :
300 DATA 67,250,0,16,65,250,0,46,33,73
310 DATA 0,4,12,30,78,65,78,117,32,121
320 DATA 0,2,128,16,67,232,0,104,74,230
330 DATA 0,143,74,104,0,20,108,6,49,124
340 DATA 235,254,0,20,66,37,0,2,128,51
350 DATA 78,117,0,0,0,0,0,0,0,0
```

LISTADO 4

```

PC_TCTRL EQU $1B002
PC_MCTRL EQU $1B020
SV_TMODE EQU $2B040
MD_SECTR EQU $012A
MOTORS_OFF_AH EQU $2B5A
MOTORS_OFF_JM EQU $2B64
MOTORS_OFF_JS EQU $2C50
MOTORS_OFF_JSU EQU $2C9A
MOTORS_OFF_MG EQU $2CC0
MOTOR_ON_AH EQU $2B64
MOTOR_ON_JM EQU $2B70
MOTOR_ON_JS EQU $2C56
MOTOR_ON_JSU EQU $2C9A
MOTOR_ON_MG EQU $2CD2
TIME_OUT EQU 1000
PC_NOTMD EQU %11100111
PC_SERB EQU %00010000
INTS_OFF EQU %0000011100000000
INTS_ON_USER EQU %1101100011111111
ERR_OK EQU 0
ERR_NF EQU -7
ERR_EF EQU -16

```

* COMIENZO DE LA RUTINA DE LECTURA DE CABECERA DE UN SECTOR DE CARTUCHO MICRODRIVE

```

TRAP A0
CMP.L #9,D1
BGE.S NOT_FOUND
TST.L D1
REQ.S NOT_FOUND
DRI.W ^INTS_OFF,SR
LEA PC_MCTRL,A3
MOVE.B ^PC_NOTMD,D0
AND.B SV_TMODE,D0
DRI.B ^PC_SERB,D0
MOVE.B D0,PC_TCTRL
MOVE.B D0,SV_TMODE
MOVEA.L ^MOTOR_ON_JS,A0
JSR (A0)
MOVE.W ^TIME_OUT,D3
BAD_HEADER SUBC.W #1,D3
REQ.S BAD_MEDIUM
LEA HEADER_BUFFER,A1
MOVEA.W MD_SECTR,A0
JSR $4000(A0)
BRA.S BAD_MEDIUM
BRA.S BAD_HEADER
MOVED ^ERR_OK,D7
BRA.S FINISH
NOT_FOUND MOVED ^ERR_NF,D7
BRA.S QUIT
BAD_MEDIUM MOVED ^ERR_EF,D7
FINISH MOVEA.L ^MOTORS_OFF_JS,A0
JSR (A0)
MOVE.B ^PC_NOTMD,D0
AND.B SV_TMODE,D0
MOVE.B D0,PC_TCTRL
MOVE.B D0,SV_TMODE
QUIT ANDI.W ^INTS_ON_USER,SR
MOVEA.L D7,D0
RTB

```

* ALMACENAMIENTO PARA CABECERA DE SECTOR

```

HEADER_BUFFER
FLAG DS.B 1
SECTOR_NUMBER DS.B 1
NAME DS.B 10
RAND_NUMBER DS.W 1
END

```

El método sólo es efectivo con programas SuperBASIC sencillos, aunque el teclado puede seguir leyéndose empleando KEYROW (por ejemplo, para leer el cursor o las teclas de función).

UN METODO MAS EVOLUCIONADO PARA LA DESACTIVACION DE BREAK

La desactivación de BREAK puede realizarse de una manera más efectiva con el empleo de una rutina adicional de planificación (*Scheduler Loop Task*). El propio sistema operativo del QL posee una rutina de este tipo, consistente en un programa que controla y planifica la secuencia de tareas y la distribución de los medios de la máquina. En nuestro caso, será solicitada cada vez que se produzca un llamamiento a la rutina de mantenimiento del QDOS, ejecutándose siempre antes que esta última.

Para que pueda desactivarse BREAK, primero tenemos que asegurarnos que la variable del sistema BV_RBK en la dirección \$008F(A6) nunca contenga 0 y que la tarea Super-BASIC siempre esté activa. También debe desactivarse el efecto de CTRL + F5 para asegurar que siempre esté en funcionamiento la pantalla, estableciendo adecuadamente el contenido de la variable del sistema asociada con ella.

El programa ensamblador del listado 2 establece una tarea de planificación para que BV_BRK nunca contenga 0 (con lo que se evita que el QDOS advierta la pulsación de BREAK) e inspecciona el estado de la tarea SuperBASIC alterándolo en consecuencia. La rutina entra siempre en funcionamiento antes que las tareas de planificación de la ROM, las cuales incluyen la secuencia de activación de BREAK.

La instrucción TAS (*Test and Set*) usada en el programa, analiza y establece el bit 7 del byte de operación en una operación indivisible. Esta es una forma limpia de señalar que BREAK no se ha pulsado, en lugar de cargar \$80 en la



variable del sistema correspondiente.

Esta rutina puede incorporarse en el interior de su archivo BOOT utilizando el método descrito anteriormente. En caso de no disponer un ensamblador, utilice el programa del listado 3, que contiene un cargador de código máquina y una demostración de sus efectos.

COMO IMPEDIR LA COPIA DEL CARTUCHO ORIGINAL

Aparte de la utilización de cartuchos ROM, el mejor medio de protección en el QL consiste en obligar al comprador de *software* a mantener su cartucho original.

Para llevar a cabo este método, debe introducirse el cartucho en el *microdrive* 2 y leer una cabecera de sector, obteniéndose un número aleatorio de 16 bits, creado en el momento del formateo. Si el número registrado en el cartucho original se lee durante su creación y se carga en una tabla en el cargador del programa que necesita protección, este cargador puede compararlo con el número leído en el cartucho del *microdrive* 2. En caso de no ser el mismo, o no existir cartucho, entrará en funcionamiento un sistema de reinicialización. En caso contrario, el cargador seguirá cargando el programa principal.

Por supuesto, el pirata tiene todavía una posibilidad entre 65536 de formatear su propio cartucho con el mismo número aleatorio de 16 bits que el del original. Sin embargo, esto bastará para obligarle a desistir de sus propósitos.

Cargando el programa del listado 5, podrá hacer funcionar el código máquina, que colocará un número aleatorio en el cartucho del *microdrive* 1. A continuación, deberá introducir otro cartucho (o el mismo) en el *microdrive* 2. Entonces, el programa lleva a cabo la comparación y le dirá si es o no es el mismo cartucho.

Las rutinas ROM para poner en marcha y parar los motores de los *microdrives* no son vectoriales como las de lecturas de cabecera de sectores, sin embargo, en futu-

LISTADO 5

```

100 MODE 4
110 LET reserved_address=RESPA (144)
120 LET address=reserved_address
130 RESTORE
140 REPEAT read_and_store_machine_code
150 IF EOF THEN EXIT read_and_store_machine_code
160 READ machine_code_byte
170 POKE address,machine_code_byte
180 LET address=address+1
190 END REPEAT read_and_store_machine_code
200 LET rom_ver:=VER$
210 IF rom_ver$(1 TO 2)='RM' OR rom_ver$='FB' THEN
220 PRINT "Return QL to Sinclair"
230 STOP
240 END IF
250 IF rom_ver$(1 TO 2)='AH' THEN
260 POKE_W reserved_address+54,11108
270 POKE_W reserved_address+96,11114
280 END IF
290 IF rom_ver$(1 TO 2)='JM' THEN
300 POKE_W reserved_address+54,11120
310 POKE_W reserved_address+96,11114
320 END IF
330 IF rom_ver$(1 TO 3)='JBU' THEN
340 POKE_W reserved_address+54,11424
350 POKE_W reserved_address+96,11418
360 END IF
370 IF rom_ver$(1 TO 2)='MG' THEN
380 POKE_W reserved_address+54,11474
390 POKE_W reserved_address+96,11468
400 END IF
410 REPEAT test_microdrive_cartridges
420 PRINT "Place a cartridge in adv1 and press a key"
430 PAUSE
440 CALL reserved_address,1
450 LET adv_one_number=PEEK_W (reserved_address+132)
460 PRINT "place a cartridge in adv2 and press a key"
470 PAUSE
480 CALL reserved_address,2
490 LET adv_two_number=PEEK_W (reserved_address+132)
500 IF adv_one_number=adv_two_number THEN
510 PRINT "Cartridge in adv2 same as that in adv1"
520 ELSE
530 PRINT "cartridge in adv2 not the same as in adv1"
540 END IF
550 END REPEAT test_microdrive_cartridges
560 :
570 DATA 78,64,178,160,0,0,0,9,100,76
580 DATA 74,129,193,72,0,124,7,0,71,249
590 DATA 0,1,128,32,16,80,0,231,192,57
600 DATA 0,2,128,160,0,0,0,16,19,192
610 DATA 0,1,128,2,19,192,0,2,128,160
620 DATA 32,124,0,0,44,86,78,144,54,60
630 DATA 3,232,83,67,103,74,67,250,0,62
640 DATA 48,120,1,42,78,160,64,0,90,10
650 DATA 96,236,128,0,96,6,126,249,94,72
660 DATA 126,240,32,124,0,0,44,80,78,144
670 DATA 16,60,0,231,192,57,0,0,128,160
680 DATA 19,192,0,1,128,3,19,192,0,2
690 DATA 106,160,2,124,216,280,32,7,78,117
700 DATA 0,0,0,0,0,0,0,0,0,0
710 DATA 0,0,0,0

```

ras versiones del QL se podrá remediar esto. De momento, debe conocer cuál es su versión de QL para poder modificar el programa.

Pasa saber cuál es su versión de ROM, deberá utilizar la función VER\$. El programa del listado 4 ha de ensamblarse incluyendo las

direcciones correctas que necesita su versión. En cuanto al programa de demostración SuperBASIC (listado 5) las direcciones necesarias se POKEan de forma automática.

Orlando Araujo Martín



ICE: Iconos, ventanas y ratones

Cuando se conecta un ordenador provisto de estas facilidades, aparece en pantalla la superficie de un escritorio con diversos objetos situados sobre él: son los iconos. Cada uno de ellos posee una función determinada, que se corresponde con comandos del sistema operativo. Para facilitar la selección se hace uso de un ratón, dispositivo que al ser desplazado sobre la mesa provoca el movimiento de un cursor en la pantalla. Colocado dicho cursor sobre el icono seleccionado, se pulsa una tecla incorporada al ratón y el ordenador ejecuta la tarea solicitada. Al igual que en un escritorio verdadero, podemos superponer en la pantalla diversas ventanas a modo de documentos con la información que necesitamos consultar. De esta manera, el uso del teclado queda reducido al mínimo.

El Macintosh, de Apple Computer, incorpora un sistema de iconos plenamente evolucionado, y su facilidad de manejo ha marcado un hito en la historia de los ordenadores personales. Recientemente, Digital Research ha desarrollado GEM, un entorno de iconos, ventanas y ratón del que existen versiones para diversas máquinas, como el IBM PC, el Atari 520 ST, y otras.

La idea de facilitar el trabajo y aprendizaje del usuario es sin duda la última moda. Sin embargo, comenzó a forjarse a comienzos de la década de los 70, en el centro de Xerox Corp en Palo Alto, California, cuando los investigadores de esta empresa decidieron que eran los ordenadores los que habían de

Hasta hace poco, el usuario sin conocimientos de informática debía entablar una lucha desigual con su ordenador. Los sistemas operativos obligaban al aprendizaje de enormes listas de comandos y sintaxis. El Lisa de Apple, mediante el empleo de un interface de usuario basado en ventanas, iconos, ratón y menus pull down, inició un proceso de simplificación en la utilización de ordenadores personales. Al Lisa le sucedió el Macintosh, con características similares y un precio más atractivo. Con el ICE (Icon Control Environment), el QL se incorpora a la misma tendencia.

adaptarse a las personas, y no al contrario, como venía sucediendo.

ICONOS EN EL QL

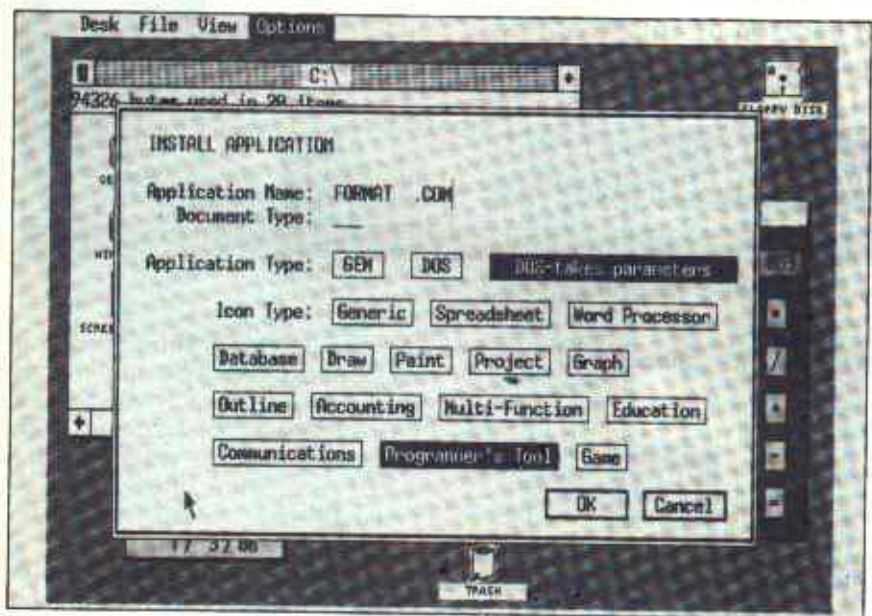
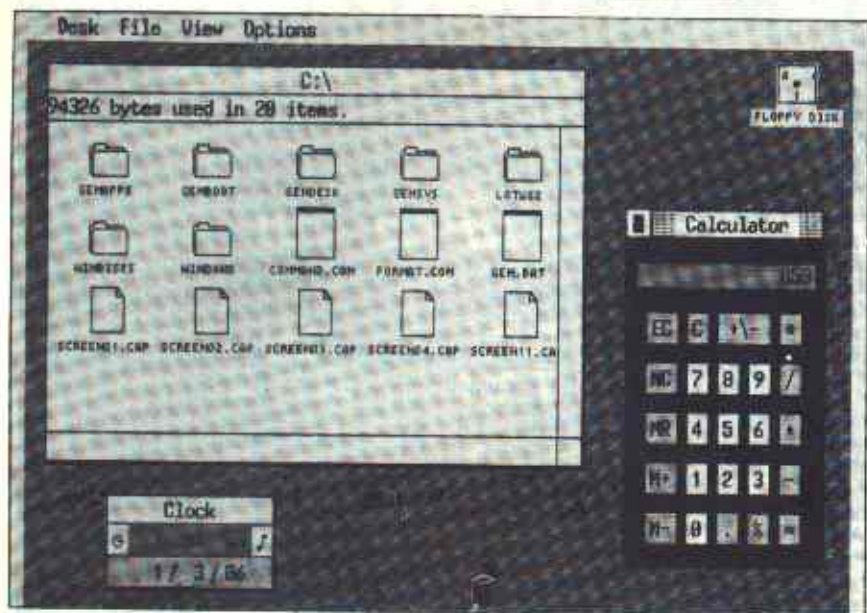
El QL también se ha querido incorporar a estas innovaciones. Basándose en los sistemas ya existentes, la compañía Eidersoft ha creado el Icon Control Environment (ICE), suministrado en un cartucho ROM. Diseñado como «herramien-

ta amable» de cara al usuario para hacer frente al complejo QDOS, el ICE tiene mucho en común con los sistemas de iconos y ventanas de otros ordenadores. Esto permite proteger al usuario de los desvarios del sistema operativo gracias a un extenso surtido protector a base de gráficos. La idea fundamental del ICE consiste en aproximar el manejo del ordenador a la gestión habitual de una oficina.

Después de enchufar el QL con la EPROM ICE de 16 K insertada en el conector de ampliación de memoria ROM, en la pantalla aparece un escritorio sobre el que hay dibujadas diversas figuras o iconos que identifican las diferentes tareas que puede realizar el QL, archivos disponibles y varias facilidades como calendario, calculadora y papelera.

COMODIDAD Y RAPIDEZ DE MANEJO

Para utilizar el ICE, todo lo que el usuario debe hacer es limitarse a desplazar por la pantalla una flecha a modo de cursor o puntero mediante las teclas de desplazamiento, joystick o ratón (este último estará disponible próximamente en nuestro país). Cuando el puntero está junto al icono elegido por el usuario o cerca de una de las opciones de control (situadas en la parte inferior de la pantalla) lo único que hay que hacer es pulsar la barra de espaciado una o dos veces, dependiendo de si desea selec-



GEM en IBM PC. Su parecido con el sistema del Macintosh no puede ser mayor.

cionar la tarea o inicializar cualquier operación implicada. El proceso, que no requiere ninguna introducción de datos desde el teclado, es prácticamente un juego de niños.

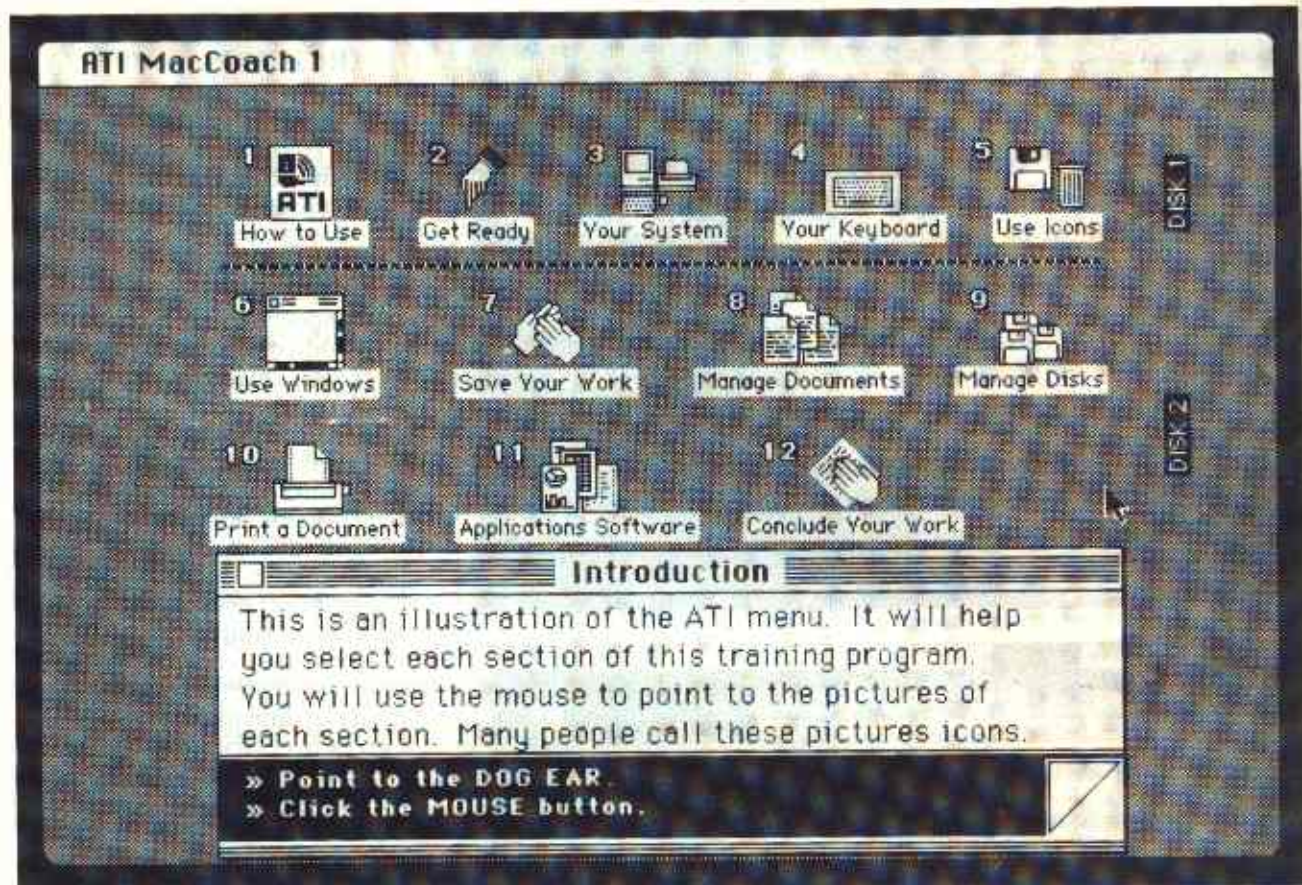
Las operaciones básicas de mantenimiento, como pueden ser la extracción de directorios, copia de archivos y ejecución de programas, se llevan a cabo de forma implícita por el ICE. El usuario sólo se limita a señalar y pulsar, ICE hace el resto. Por ejemplo, si se ve en la

necesidad de copiar o borrar varios archivos, basta señalar el archivo en cuestión y llevarlo después a su dispositivo de destino o a la papelera según tenga que copiar o destruir el archivo señalado. Si las operaciones a realizar son de carácter irreversible, ICE pedirá al usuario la confirmación de sus intenciones. Cuando se producen errores, la pantalla muestra los mensajes correspondientes, escritos, por supuesto en inglés (aunque bastante sencillos de comprender).

GESTION DE ARCHIVOS

Los archivos en disco y *microdrive* se paginan en la pantalla en grupos de 12. ICE es capaz de distinguir automáticamente diferentes clases de archivos, utilizando diferentes iconos para representarlos. Las clases de ficheros que puede reconocer son archivos generales en código máquina, programas en código máquina ejecutables (EXEC), programas en SuperBASIC, documentos QUILL, hojas del ABACUS, gráficos del EA-SEL, y bases de datos del ARCHIVE, así como archivos de ejecución automática (boot). La mayor ventaja que se desprende en este caso, es que el usuario puede saber mediante un simple vistazo el contenido de un archivo. Otro punto menos evidente consiste en que para hacer funcionar un programa ya no es necesario distinguir entre archivos que se ejecutan con EXEC o RUN. Los programas cargadores de los paquetes de PSION (versión 2.000) ya no son necesarios.

El mantenimiento general y las operaciones del sistema aparecen de modo explícito como opciones en un panel de control en la parte inferior de la pantalla. Estos permiten cambiar el nombre de archivo (RENAME), inspeccionar sus contenidos (VIEW) y solicitar información acerca de uno de ellos (INFO), con lo cual conoceremos el nombre completo del archivo, longitud y tipo y últimos datos rectificados o añadidos. Con las opciones de este panel también se pueden formatear diskettes o cartuchos de *microdrive* (FORMAT) y solicitar información referente a la memoria disponible, rutinas que están funcionando y la versión del QDOS o ICE que se está utilizando. La opción CUSTOM permite variar según las necesidades del usuario, los datos (que normalmente funcionan por omisión con unos valores predeterminados) referentes a la fecha, hora, nombres de dispositivos, instalación de impresora, velocidad del cursor y sensibilidad de las teclas. Finalmente, existe una opción para



MacCoach, un programa para aprender a manejar el Macintosh.

abandonar el ICE y volver al SuperBASIC.

FUNCIONES SUPLEMENTARIAS

ICE incorpora dos funciones especiales: calendario y calculadora. Utilizando la primera, la pantalla muestra un calendario completo con todos los meses, marcando la fecha actual iluminándola en rojo. Con el empleo de la segunda, aparece en la pantalla la parte frontal de una típica calculadora de bolsillo, con sus teclas y pantalla de visualización de datos. Incluye una tecla de memorización y porcentaje y funciona desplazando el cursor sobre las teclas.

Al adquirir la EPROM ICE, Eidersoft también proporciona al comprador de un cartucho de *microdrive* que contiene una serie de programas (para utilizarlos conjuntamente con el ICE) que le per-



Pantalla de QL con ICE. Se observa la calculadora y varios iconos.

miten hacer funcionar en multitarea cuatro programas al mismo tiempo. Este paquete de programas (llamado CHOICE) puede ser de gran utilidad con una ampliación de 256 Kbytes de memoria, ya que si reservamos 128 Kbytes para ARCHIVE, ABACUS y QUILL (por ejemplo) podremos hacerlos funcionar simultáneamente y acceder individualmente a cada uno de ellos pulsando simplemente CRTR + F3. No solamente es impresionante la velocidad con la

que nos movemos entre las distintas aplicaciones, sino también el hecho de que cuando volvemos a una de ellas, nos encontramos exactamente en el mismo punto donde las dejamos antes, ya sea documento, base de datos u hoja de cálculo.

Al no tener un elaborado sistema de ventanas, el ICE es mucho menos sofisticado que otros programas semejantes. Sin embargo, hay que tener en cuenta que el programa solamente ocupa 16 K, en lugar de los 90 K de su hermano mayor GEM. De todos modos, es fácil de utilizar y está diseñado a prueba de errores. En combinación con el paquete CHOICE, abre una nueva dimensión para el usuario que no había logrado sacar el máximo fruto de su QL hasta este momento.

Nota: ICE se comercializa en el Reino Unido a un precio aproximado de 60 libras.

Orlando Araujo Martín