REUBICADOR DE PROGRAMAS EN C/M

La REUBICACION de programas en código máquina se divide en dos partes. Primero uno tiene que escribir el programa en forma REUBICABLE. Entonces uno lo presenta al sistema con un cargador de REUBICAR, el cual se ocupa de colocar el nuevo código junto a cualquier memoria ya en uso, y deja el sistema tan poco cambiado como sea posible.

na rutina que se reubica por sí mismo tiene la marca de ser profesional. Una que no lo hace es semi-profesional. Para nuestros propios usos en casa todos los programadores hacemos la reubicación en el momento de ENSAMBLAR: variando ORG simplemente. Pero es como usar una manivela para arrancar el coche en lugar de la batería. **Reubicación es la batería de arranque que quita los problemas en el momento de poner el sistema en marcha.** No obstante, la reubicación es el toque final a un programa, y es lo suficientemente difícil (hablando del Z80) que no suele aplicarse hasta que el programa esté en su forma definitiva.

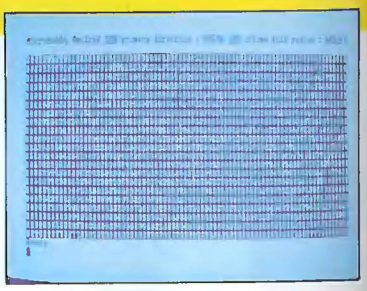
Rutinas en máquinas profesionales

La rutina que transforma un programa en reubicable se ve en el listado de ensamblador. Es el mismo que hemos usado en el programa HOBBYCOP del número 27 de la revista, pero entonces figuraban sólo los bytes del código máquina. El uso de ORG 0 hace que todas las etiquetas generadas en la compilación son relativas al principio de la rutina. La llamada falsa al firmware fija el valor del PC (Program Counter) que se recupere del stack y se añade a las direcciones relativas necesarias en la manera mostrada. El uso de una llamada falsa para fijar PC, como a menudo pasa con programación del Z80, es un truco que una vez visto, no se olvida.

Generación de la tabla

¿Cómo se genera la tabla de reubicación? El programa HOJEAMEN.TXT se presenta con su listado en el momento de ensamblar. Se ha usado GENA de Hisoft —un ensamblador de dos pasos— que obliga a colocar la tabla de reubicación al final. Se ha usado mayúsculas para las etiquetas que representan nudos lógicos del programa. Minúsculas se usan para puntos de entrada desde el programa principal *(p. ej. Basic)* y para las entradas en la tabla de reubicación. Estas últimas se ha dado un número «qq» si corresponde a instrucciones de tres bytes, y «**rr**» si son de cuatro. Sus entradas correspondientes en la tabla son entonces qq+1 y rr+2. No se reubica las instrucciones son saltos relativos: JR DJNZ, etc. Tampoco las llamadas al Jump Block *(saltos)* del firmware: entonces éstas se han dejado en forma numérica JP #BB33 en lugar de dar el nombre JP km.set.control.

Después de preparar la tabla de reubicación al final del programa, hay que incluirlo en la rutina de reubicar con algo como el comando *F HOJEAMEM.TXT de GENA o incluirlo directamente con un merge. Para los que usamos cassette, la segunda manera es la más fácil, siempre que la memoria lo permite.



Ahora se ensambla todo. Con ORG 0 el código objeto hay que guardarlo en alguna otra parte de la memoria. GENA lo hace con opción 16. Toma nota de la longitud de la TABLE al final del programa. Esta longitud se pone ahora en la pseudo-instrucción **«tabla: DEFS 76»** — 76 era la longitud para el ejemplo final, pasando los bytes de la **«TABLE»** a su nueva posición **«tabla»** para que no ocupen memoria indebidamente después de ser usados. Serán tirados con la basura. Una manera de hacerlo se da al final de la rutina de reubicar.

Programa HOJEAMEN

Un ejemplo de un CARGADOR de reubicar se ve en el programa Basic en el cual el fichero binario del ensamblador ha sido pasado a código máquina en líneas DATA..Claro que si tiene el fichero en binario, no hay que teclear líneas 100-350 ni la subrutina 2000 (2000-2999). En su lugar, linea 1007 se cambia a:

1007 LOAD''hojeamen'', h+1

y el binario se carga en memoria más rápidamente. **Si tecleas los DATA, usa MODE 2:** el formato es de 64 caracteres por línea para ajustarse a la norma de tu revista, y si saltas un dato será claro a simple vista. Hay un checksum al final de cada línea: un error se señala con el mensaje «Error en xxx» y las líneas de DATA listarán. No hace falta contar espacios: las identaciones (p. eje. líneas 2010-2110) se usan para ayudar comprender la estrucutra de los bucles, y no necesitas seguirlas. Las observaciones (REMs) que siguen d apóstrofe (') pueden reemplazarse con un par de asteriscos ('**').

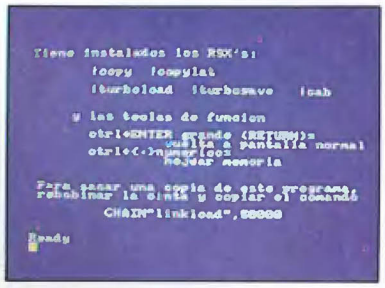
Si no tienes interés ahora mismo en usar la reubicación, teclea el programa en Basic porque el resultado es una herramienta potente en abrir programas en binario, en seguir la disponibilidad de la memoria, en depurar programas largos en Basic, etc.

```
*H Rutina de reubicar : Microhobby AMSTRAD Semanal

        ORG 0000        ;usar opción 16 con GENA
GETLOC:EQU #BB21        ;llamada falsa al firmware
REUBIC:CALL GETLOC      ;pone PC+3 en el stack
        DEC SP
        DEC SP
        EX (SP),HL      ;HL=PC+3 original
        DEC HL
        DEC HL
        DEC HL          ;offset (constante de reubicar)
        INC SP
        INC SP          ;SP restaurado
        PUSH HL
        LD DE,tabla-2   ;tabla de reubicar estará aquí
        ADD HL,DE
        POP BC          ;BC=offset
REUB1 :LD E,(HL)        ;empezar bucle
        INC HL
        LD D,(HL)
        INC HL          ;saca una dirección de la tabla
        LD A,D
        OR E            ;comprobar señal para fin de tabla
        JR Z,REUB2      ;hecho
        PUSH HL         ;puntero en la tabla
        EX DE,HL        ;HL=dirección
        ADD HL,BC
        PUSH HL         ;dirección+offset
        LD E,(HL)
        INC HL
        LD D,(HL)
        EX DE,HL        ;HL=(dirección+offset)
        ADD HL,BC       ;offset añadido
        EX DE,HL
        POP HL          ;saca dirección+offset
        LD (HL),E       ;pokear el nuevo valor
        INC HL
        LD (HL),D
        POP HL          ;saca puntero en la tabla
        JR REUB1        ;al bucle
REUB2 :JP keydef        ;JP LOGRSX si es programa RSX ó RET
        DEFW REUB2+1    ;comenzar ya la reubicación
tabla :DEFS 76          ;ó lo que sea, 76 en este ejemplo
endtbl:DEFW 0           ;señal de fin de tabla
        DEFS 2          ;zona de seguridad
*F HOJEAMEM.TXT

;ASSEMBLE opción-16 (estilo GENA)
;SAVE fichero binario en cinta o disco
;LOAD este mismo fichero a, digamos, #5000
;INTELLIGENT COPY bajo MONA (opción I)
;  Start: 5000+TABLE
;   Last: 5000+zzzzzz-1
;     To: 5000+tabla
;SAVE binario bajo MONA con opción WRITE (opción W)
;   Name: HOJEAMEM  por ejemplo
;  First: 5000
;   Last: 5000+TABLE (no +TABLE-1 un error en MONA)
;  Start: -
;Usar ajustandolo a h = himemory con LOAD a (h-TABLE)
;Tirar la basura de reubicación moviendo el puntero
;            de memoria arriba hasta (h-TABLE+entry).
;Versión: 7.11.86
```

La rutina capta-error 5000 es un intento de dejar SYMBOL AFTER de BAsic en su estado original. Se nota que la memoria tira hacia arriba otra vez en 3000 después de la reubicación.

Listing HOJEAMEN.BAS

El programa que ahora tienes, HOJEAMEN.BAS, es diseñado a ser cargado cuando se enciende la máquina. Deja dos nuevas teclas de función: CTRL/ [.]-numérico te permite hojear por la RAM, y CTRL/ENTER-grande (o CTRL/RETURN estilo 128) te devuelve cuando lo quieres tu pantalla que estabas usando antes (INKs PAPER PEN BORDER MODE). Están asignados los números 158/159, los más altos posibles, para no molestar a otros programas que se puedan usar después. **Las rutinas están disponibles como teclas y no como RSX porque son de uso en modo comando y no en tiempo de ejecución.**

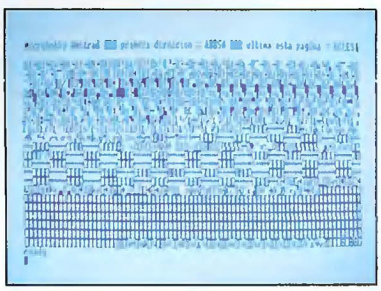


Ahora pulsa CTRL [.]-numérico. Dirección de comienzo: ENTER, y empieza en el 0000 por defecto. Estas son las celdas de la memoria. Es tan rápida la pantalla, que hay una pausa de dos segundos al final de cada página. Cuando se aburre de ver RAM vacía, pulsa una tecla y se parará. Ahora dirección de comienzo: &b446 y pulsa una tecla para pararlo ya. Lo que se ve es el buffer de las teclas de función con RUN", nuestros CALL &XXXX, etc. (En modelos más modernos que el nuestro, puede ser que la dirección no es exactamente &B446, pero estará cercana.)

Cuando quieres, pulsa ESC una vez para volver a Basic. ¡Qué bonito! ¿Verdad? Si tienes impresora, puedes sacar lo que ves a papel usando nuestro RSX COPY de hace unas pocas semanas. (Era también reubicable.) Ahora pulsa CTRL/ENTER-grande y ¡voila!

No hay que quitar estas cosas de la máquina. Están allí para usarse con otros programas. Por ejemplo, estás tecleando un programa largo en Basic y algo no funciona. Hojea la memoria baja hasta que veas el final del Basic. Entonces se ve como Basic apila los variables al final de sí mismo durante la ejecución. Y se ve **"COUNT. SWAP..."** y muy cercana **"COUNT. SUAP..."** ¡Ajá! Un pequeño error de ortografía inglesa.

Vas a encontrar la función CTRL/ENTER-grande muy útil. Muchas veces nos encontramos con una pantalla ilegible. Tocamos esta tecla, y otra vez tenemos la pantalla del momento en que hicimos la última llamada a HOJEAMEN. Diviértete.



```
                        1 ;LC HOJEAMEM Amstrad Semanal 1Feb86
                        2 ;hojear memoria - ctrl/num.pad (.)
                        3 ;reponer inks,mode,paper,border,etc.
                        4 ;                 ctrl/ENTER (RETURN)
007F  3ADC00            5 keydef: LD    A,(DIRMES-6)
0082  EE24              6         XOR   36
0084  FE4C              7         CP    "L"
0086  C0                8         RET   NZ
0087  069F              9         LD    B,159
0089  3E12              10        LD    A,18
008B  CD33BB            11        CALL  #BB33
008E  05                12        DEC   B
008F  3E07              13        LD    A,7
0091  C333BB            14        JP    #BB33
0094  CFCFAA            15 entry: DEFB  207,207,164
0097  31393836          16        DEFM  "1986 Hobby Press S.A."
00AC  0D12FF            17 ASKDR1: DEFB 13,18,#FF
00AF  07                18 ASKDIR: DEFB 7
00B0  496E7075          19        DEFM  "Input direccion "
00C0  64652063          20        DEFM  "de comienzo &...."
00D1  08080808          21        DEFB  8,8,8,8,#FF
00D6  0C                22 DIRMES: DEFB 12
00D7  4D696372          23        DEFM  "Microhobby Amstrad "
00EA  CFCFCF            24        DEFB  207,207,207
00ED  20707269          25        DEFM  " primera direccion = &"
0103  FF                26        DEFB  #FF
0104  20CFCFCF          27 DIRMS2: DEFB  32,207,207,207
0108  20756C74          28        DEFM  " ultima esta pagina = &"
011F  FF                29        DEFB  #FF
0120  D90AFF            30 DIRMS3: DEFB  217,10,#FF
0123                    31 DIRECC: DEFS  2
0125  00                32 USFLAG: DEFB  0
0126  0D1C20            33 USDATA: DEFB  13,28,32
0129  0B0B1C01          34 USINK0: DEFB  11,11,28,1
012D  202004            35 USINK1: DEFB  32,32,4
0130  021D              36 USMOD:  DEFB  2,29
0132  0B0B0E            37 USBOR:  DEFB  11,11,14
0135  200F              38 USPAP:  DEFB  32,15
0137  010DFF            39 USPEN:  DEFB  1,13,#FF
013A  0D1C20            40 BRDATA: DEFB  13,28,32
013D  0B0B1C01          41 brink0: DEFB  11,11,28,1
0141  20200402          42 brink1: DEFB  32,32,4,2,29
0146  0B0B0E20          43 brbor:  DEFB  11,11,14,32,15,1,13,7,#FF
014F                    44 relnic:
014F  AF                45        XOR   A
0150  322501            46 qq1:   LD    (USFLAG),A
0153  212601            47 qq2:   LD    HL,USDATA
0156  C3F501            48 qq3:   JP    PRTTXT
0159                    49 browse:
0159  3A2501            50        LD    A,(USFLAG)
015C  B7                51        OR    A
015D  2036              52        JR    NZ,BROWS1
015F  CD93BB            53        CALL  #BB93
0162  323701            54 qq4:   LD    (USPEN),A
0165  CD99BB            55        CALL  #BB99
0168  323501            56 qq5:   LD    (USPAP),A
016B  CD11BC            57        CALL  #BC11
016E  323001            58 qq6:   LD    (USMOD),A
0171  AF                59        XOR   A
0172  CD35BC            60        CALL  #BC35
0175  ED432901          61 rr1:   LD    (USINK0),BC
0179  3E01              62        LD    A,1
017B  322501            63 qq7:   LD    (USFLAG),A
017E  CD35BC            64        CALL  #BC35
0181  ED432D01          65 rr2:   LD    (USINK1),BC
0185  CD3BBC            66        CALL  #BC3B
0188  ED433201          67 rr3:   LD    (USBOR),BC
018C  213A01            68 qq8:   LD    HL,BRDATA
018F  CD19BD            69        CALL  #BD19
0192  CDF501            70 qq9:   CALL  PRTTXT
0195  21AC00            71 BROWS1: LD   HL,ASKDR1
0198  CDF501            72 qq10:  CALL  PRTTXT
019B  CD81BB            73        CALL  #BB81
019E  CDFF01            74 BROWS6: CALL KFLUSH
01A1  21AF00            75 qq11:  LD    HL,ASKDIR
01A4  CDF501            76 qq12:  CALL  PRTTXT
01A7  0604              77        LD    B,4
01A9  210000            78        LD    HL,0
01AC  222301            79 qq13:  LD    (DIRECC),HL
01AF  212301            80 qq14:  LD    HL,DIRECC
01B2  CD06BB            81 INPHEX: CALL #BB06
01B5  FEFC              82        CP    #FC
01B7  2839              83        JR    Z,BROWS5
01B9  FE0D              84        CP    13
01BB  2805              85        JR    Z,INPHX1
01BD  CD0802            86 qq15:  CALL  ASC2HL
01C0  20F0              87        JR    NZ,INPHEX
01C2  CDFF01            88 INPHX1: CALL KFLUSH
01C5  ED5B2301          89 rr4:   LD    DE,(DIRECC)
01C9  CD84BB            90 BROWS2: CALL #BB84
01CC  CD2902            91 qq16:  CALL  DIROUT
01CF  0E07              92        LD    C,7
01D1  06F0              93 BROWSA: LD   B,240
01D3  1A                94 BROWS3: LD   A,(DE)
01D4  C5                95        PUSH  BC
01D5  D5                96        PUSH  DE
01D6  CD5DBB            97        CALL  #BB5D
01D9  D1                98        POP   DE
01DA  C1                99        POP   BC
01DB  13                100       INC   DE
01DC  10F5              101       DJNZ  BROWS3
01DE  0D                102       DEC   C
01DF  20F0              103       JR    NZ,BROWSA
01E1  CD81BB            104       CALL  #BB81
01E4  3EC8              105       LD    A,200
01E6  CD6702            106 qq17:  CALL CDELAY
01E9  CD1BBB            107       CALL  #BB1B
01EC  30DB              108       JR    NC,BROWS2
01EE  FEFC              109       CP    #FC
01F0  20AC              110       JR    NZ,BROWS6
01F2  21AC00            111 BROWS5: LD  HL,ASKDR1
01F5                    112 PRTTXT:
01F5  7E                113       LD    A,(HL)
01F6  23                114       INC   HL
01F7  FEFF              115       CP    #FF
01F9  C8                116       RET   Z
01FA  CD5ABB            117       CALL  #BB5A
01FD  18F6              118       JR    PRTTXT
01FF  0600              119 KFLUSH: LD  B,0
0201  CD09BB            120 KFL1:  CALL #BB09
0204  D0                121       RET   NC
0205  10FA              122       DJNZ  KFL1
0207  C9                123       RET
0208  4F                124 ASC2HL: LD  C,A
0209  FE61              125       CP    "a"
020B  3802              126       JR    C,ASCHL1
020D  D620              127       SUB   #20
020F  D630              128 ASCHL1: SUB "0"
0211  D8                129       RET   C
0212  FE0A              130       CP    10
0214  380B              131       JR    C,ASCHL2
0216  D607              132       SUB   7
0218  FE0A              133       CP    10
021A  D8                134       RET   C
021B  FE10              135       CP    16
021D  D0                136       RET   NC
021E  ED6F              137 ASCHL2: RLD
0220  23                138       INC   HL
0221  ED6F              139       RLD
0223  2B                140       DEC   HL
0224  79                141       LD    A,C
0225  05                142       DEC   B
0226  C35ABB            143       JP    #BB5A
0229  21D600            144 DIROUT: LD  HL,DIRMES
022C  CDF501            145 qq18:  CALL PRTTXT
022F  4A                146       LD    C,D
0230  CDAF02            147 qq19:  CALL CHEXPT
0233  4B                148       LD    C,E
0234  CDAF02            149 qq20:  CALL CHEXPT
0237  210A01            150 qq21:  LD   HL,DIRMS2
023A  CDF501            151 qq30:  CALL PRTTXT
023D  218F06            152       LD    HL,240*7-1
0240  19                153       ADD   HL,DE
0241  4C                154       LD    C,H
0242  CDAF02            155 qq31:  CALL CHEXPT
0245  4D                156       LD    C,L
0246  CDAF02            157 qq32:  CALL CHEXPT
0249  212001            158 qq33:  LD   HL,DIRMS3
024C  C3F501            159 qq22:  JP   PRTTXT
024F  79                160 CHEXPT: LD  A,C
0250  E6F0              161       AND   #F0
0252  0F                162       RRCA
0253  0F                163       RRCA
0254  0F                164       RRCA
0255  0F                165       RRCA
0256  CD5C02            166 qq23:  CALL NASC11
0259  79                167       LD    A,C
025A  E60F              168       AND   #0F
025C  FE0A              169 NASC11: CP  10
025E  3802              170       JR    C,NAS1
0260  C607              171       ADD   A,7
0262  C630              172 NAS1:  ADD  A,"0"
0264  C35ABB            173       JP    #BB5A
0267  01D809            174 CDELAY: LD  BC,2520
026A  0D                175 CDLAY1: DEC C
026B  20FD              176       JR    NZ,CDLAY1
026D  05                177       DEC   B
026E  20FA              178       JR    NZ,CDLAY1
0270  3D                179       DEC   A
0271  20F4              180       JR    NZ,CDELAY
0273  C9                181       RET
0274  80005101          182 TABLE: DEFW keydef*1,qq1*1,qq2*1,qq3*1
027C  5A016301          183       DEFW browse*1,qq4*1,qq5*1,qq6*1
0284  77017C01          184       DEFW rr1*2,qq7*1,rr2*2,rr3*2
028C  8D019301          185       DEFW qq8*1,qq9*1,BROWS1*1,qq10*1
0294  9F01A201          186       DEFW BROWS6*1,qq11*1,qq12*1,qq13*1
029C  B001BE01          187       DEFW qq14*1,qq15*1,INPHX1*1,rr4*2
02A4  CD01E701          188       DEFW qq16*1,qq17*1,BROWS5*1
02AA  2A022D02          189       DEFW DIROUT*1,qq18*1,qq19*1,qq20*1
02B2  38024D02          190       DEFW qq21*1,qq22*1,qq23*1
02B8  3B024302          191       DEFW qq30*1,qq31*1,qq32*1,qq33*1
02C0                    192 zzzzzz:
                        193

Pass 2 errors: 00
```

```
100 DATA CD21BB3B3BE32B2B2B3333E5112D0019C15E2356237AB328110846
110 DATA E5EB09E55E2356EB09EBE1732372E118E7C37F002B008000510B7B
120 DATA 01540157015A01630169016F0177017C0183018A018D019301056D
130 DATA 96019901 9F01A201A501AD01B001BE01C301C701CD01E701F3096D
140 DATA 012A022D02310235023802 4D0257023B0243024702 4A02000002BF
150 DATA 00003ADC00EE24FE4CC0069F3E12CD33BB053E07C333BBCFCF0A7B
160 DATA A4313938362048 6F6262792050726573732053 2E412E0D12FF07EB
170 DATA 07496E7075742064697265636 96F6E2064652063 6F6D696508FD
180 DATA 6E7A6F20262E2E2E2E08080808FF0C4D6963726F686F626279078E
190 DATA 20416D7374726164 20CFCFCF207072696D6572612064972650A4D
200 DATA 6363696F6E203D2026FF20CFCFCF20756C74696D6120657374 0A53
210 DATA 61207061676 96E61203D2026FFD90AFF00000000D1C200B0B1C06F0
220 DATA 012020040 21D0B0B0E200F010DFF0D1C200B0B1C012020040202 86
230 DATA 1D0B0B0E200F010D07FFAF3225012126 01C3F5013A2501B72005C3
240 DATA 36CD93BB323701CD99BB323501CD11BC323001AFCD35BCED430ADE
250 DATA 29013E01322501CD35BCED432D01CD3BBCED43320121 3A01CD082D
260 DATA 19BDCDF50121AC00CDF501CD81BBCDFF0121AF00CDF50106040B9C
270 DATA 21000022230121230 1CD06BBFEFC2839FE0D2805CD080220F007B4
280 DATA CDFF01ED5B2301CD84BBCD29020E0706F01AC5D5CD5DBBD1C10C73
290 DATA 1310F50D20F0CD81BB3EC8CD6702CD1BBB30DBFEFC20AC21AC0CBB
300 DATA 007E23FEFFC8CD5ABB18F60600CD09BBD010FAC94FFE6138020C78
310 DATA D620D630D8FE0A3808D607FE0AD8FE10D0ED6F23ED6F2B79050C3B
320 DATA C35ABB21D600CDF5014ACD4F024BCD4F02210401CDF501218F09FC
330 DATA 06194CCD4F024DCD4F02212001C3F50179E6F00F0F0F0FCD5C08A3
340 DATA 0279E60FFE0A3802C607C630C35ABB01D8090D20FD0520FA3D09B5
350 DATA 20F4C901DD
1000 endit=&274:entry=&94'     longitud de codigo/long. de basura
1001 hojea=&159:vuelta=&14F
1002 ON ERROR GOTO 4000
1003 sym=256:GOSUB 5000'            tratamiento del symbol buffer
1005 h=HIMEM-endit
1006 MEMORY HIMEM-endit
1007    direc=h:GOSUB 2000'                                  cargador
1008   'si ya hay el binario, sustituir: 1007 LOAD"hojeamem",h+1
1009    CALL h+1
1010 GOTO 3000'                                                  fin
2000 '                              rutina carga datos con checksum
2010 FOR linea=100 TO 350 STEP 10
2020   READ postizo$
2030   check.sum=VAL("&"+RIGHT$(postizo$,4))
2040   FOR puntero=1 TO LEN(postizo$)-5 STEP 2
2050     dummy=VAL("&"+MID$(postizo$,puntero,2))
2060     check.sum=check.sum-dummy
2070     direc=direc+1
2080     POKE direc,dummy
2090     NEXT puntero
2100   IF check.sum<>0 GOTO 4020
2110   NEXT linea
2120 '                                  sitio para acomodacion propia
2130 '                                                        etc
2999 RETURN
3000 MEMORY HIMEM+entry:sym=240:GOSUB 5000' reponer user-symbols
3040 a$="CALL &":b$=HEX$(h+1+hojea,4):c$=CHR$(13)
3045 KEY 158,c$+a$+b$+c$
3050 PRINT"Hojear la memoria RAM con"TAB(3)a$+b$
3060 b$=HEX$(h+1+vuelta,4)
3065 KEY 159,c$+a$+b$+c$
3070 PRINT"Vuelta a colores de antes con"TAB(3)a$+b$
3090    PRINT
3100 PRINT"CTRL + (.) del bloque numerico = HOJEAR
3110 PRINT"CTRL + (ENTER grande o RETURN) = VUELTA
3120    PRINT
3130 NEW'                  prog HOJEAMEM Microhobby Amstrad Semanal
4000 IF ERR<>permit.error THEN PRINT"**** Error ****":STOP
4010 RESUME NEXT
4020 MODE 2:PRINT CHR$(7)"Error en"linea:LIST-999'Dr.RAC badajoz
5000 permit.error=5:SYMBOL AFTER sym'          rutina para manejar
5001 permit.error=0:RETURN'                el buffer de los symbols
```