

# MONITOR DE DISCO

*A ver, entre nosotros, ¿quién no está harto de que le sucedan cosas raras al disco, que normalmente acaban en la misteriosa desaparición de ficheros cruciales?*

*Además, el disco es una especie de territorio vedado al sufrido usuario, algo así como «El Castillo Negro del Hechicero Negro». Ya es hora de remediar este estado de cosas, y «DiscEdit» es el programa de hacer malabarismos con el disco, el programa es corto, bonito y usa un sofisticado estilo de programar.*



DiscEdit es un monitor de disco que le permitirá editar y manipular cualquier sector o pista. Siendo cuidadosos, incluso es factible recuperar ficheros borrados accidentalmente.

Para poder usarlo a pleno gas, hemos de tomar en consideración cómo trabaja el sistema de disco de **Amstrad**. Existen tres formatos estándar:

- Formato Sistema (CP/M).
- Formato IBM (IBM PC y CP/M compatible).
- Formato datos.

Todos ellos tienen algo en común: cada uno posee 40 pistas por disco, y los sectores son de una longitud de 512 bytes. Las principales diferencias son el número de sectores por pista y los números de sector.

## Los distintos formatos de disco

Aquí hay una lista para cada formato:

- Formato sistema:  
9 Sectores por pista  
números de sector, 65 a 73
- Formato IBM:  
8 sectores por pista  
números de sector, 1 a 8
- Formato datos:  
9 sectores por pista  
números de sector, 193 a 201

No es necesario preocuparse del formato del disco que queramos editar, porque DiscEdit hará el trabajo por nosotros; lo anterior es sólo a título informativo, y para el que desee

modificar el programa por su cuenta y riesgo, adaptándolo a sus necesidades concretas.

## Funcionamiento de DiscEdit

Pasemos a estudiar cómo funciona DiscEdit.

Telee el programa y sávelo. Al arrancarlo, la pantalla cambiará a modo 2, se dibujará un espacio cerrado y se nos preguntará por una unidad de disco. Hay que responder 0 (para el disco A) o 1 (para el B).

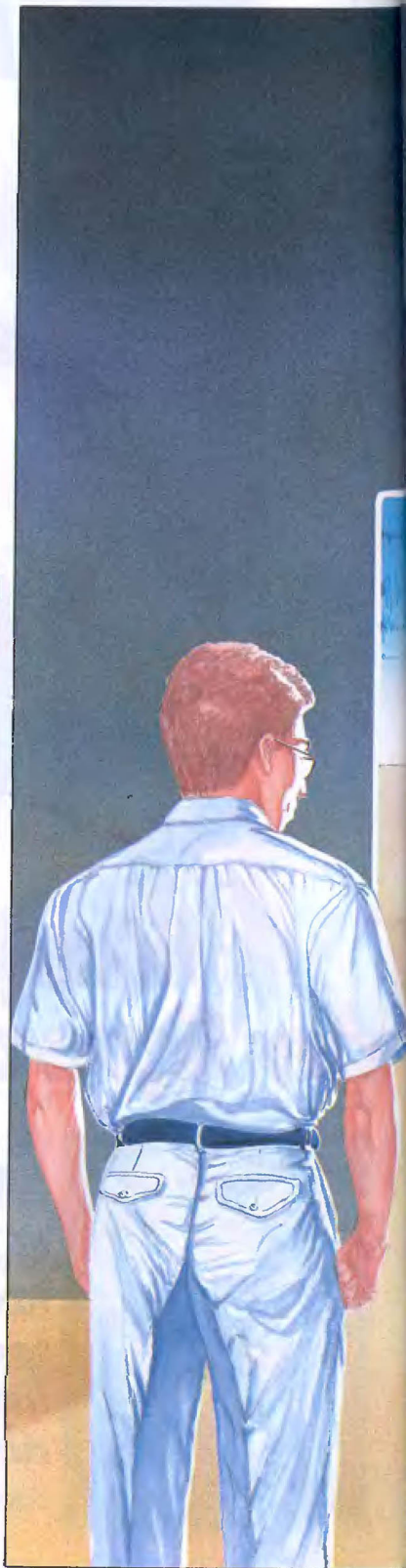
El disco seleccionado entrará en funciones durante un momento. DiscEdit hace esto para averiguar el formato y otras cosas; el resultado de la operación aparecerá en la esquina superior derecha de la pantalla.

Acto seguido, hay que indicarle al programa la pista inicial donde comenzará la edición (el «**track**»), un número comprendido entre 0 y 39. Lógicamente, después DiscEdit inquirirá acerca del sector que deseamos editar.

Como hemos indicado más arriba, el número de sector puede variar con el formato del disco; no se preocupe: el rango de valores aceptables aparecerá siempre en pantalla.

Al principio de cada línea hay un número hexadecimal que hace referencia al «**offset**» de los datos en el sector, esto es, el número del byte del sector a partir del cual vamos a editar. El resto de la línea muestra el contenido de ese byte, junto con el de los 15 bytes siguientes y su equivalencia en formato ASCII.

La pantalla no es lo suficientemente grande para mostrar los 512 bytes de un golpe, así que sólo aparecen



256. La otra mitad se obtiene pulsando la tecla «2» del teclado numérico (f2 en el 664 y 6128). Pulsándola una vez más, se retorna a la primera mitad. El número siguiente a «página» en la parte superior de la pantalla indica la porción que estamos viendo: 0 si es la primera ó 1 si es la segunda. A la izquierda del número de página están los de pista y sector en uso en ese momento.

Hay a su disposición una serie de opciones de edición como examinar, leer y escribir pistas y sectores. La tabla número 1 lista todo esto, junto con los controles de teclado correspondientes.

**\*\*TABLA I\*\***

No obstante, la página de ayuda listará todos los comandos pulsando 0 del teclado numérico (fo en 664 y 6128).

Si ocurre algún error de disco en el proceso de edición, el mensaje aparecerá en la parte inferior de la pantalla junto con el consabido mensaje de «RETRI, IGNORE O CANCEL?».

En este caso, es conveniente reintentar, porque a veces funciona; no tenemos ni idea de a qué se debe este comportamiento errático del **Amstrad**.

**Cómo usar discEdit**

Vamos a echar un vistazo a un típico ejemplo de cómo debiera usarse DiscEdit.

Primero de todo, el disco debe seleccionarse. En la mayoría de los casos, respondremos 0, señalando el disco A. Lo siguiente es escoger pis-

ta y sector: supongamos que introducimos 2 y 65, respectivamente.

El sector se encargará del disco y aparecerá en pantalla. Si, a pesar de todo, nos hemos equivocado, podemos desplazarnos a través del disco con las flechas del cursor, manteniendo apretada la tecla «SHIFT» («MAYS» en el 6128).

Por ejemplo, pulsando MAYS+ cursor derecha, se cargaría del disco el sector 66 de la pista 2. Una vez que estamos en el sector deseado, podemos comenzar a editar. Hay dos formas de hacerlo: en hexadecimal o en ASCII. Se selecciona uno y otro pulsando el 3 del teclado numérico (f3 para 664 y 6128).

En modo hexadecimal, los contenidos se alteran entrando dígitos del 0 al 9 y letras de la «a» a la «f». Observe que el teclado numérico no debe usarse para los números; use las teclas encima de «QUERTY».

El byte que está siendo editado puede cambiarse moviendo el cursor de edición al byte que queramos, pulsando las teclas del cursor apropiadas. La otra posibilidad de edición, como decíamos, permite entrar caracteres ASCII; por ejemplo, si se tecldea ABCD, aparecerán los bytes &41, &42, &43 y &44.

Se puede saber en qué modo de edición estamos observando dónde se encuentra el cursor; si está encima de los números hexa, estamos en modo numérico, de lo contrario en ASCII. Una vez que el sector se ha dejado a nuestro gusto, podemos escribirlo en el disco. Para ello, basta pulsar la tecla 4 del teclado numérico (f4), y responder afirmativamente a la pregunta de si estamos seguros. Es el momento de descubrir cómo guarda el **Amstrad** la información en el disco.

Opciones de pista/sector	Siguiente sector
SHIFT           Cursor derecha	Sector anterior
SHIFT           Cursor izquierda	Avanza una pista
SHIFT           Cursor arriba	Retrocede una pista
SHIFT           Cursor abajo	
<b>Comandos de edición</b>	un byte hacia la izquierda
Cursor izquierda	un byte hacia la derecha
Cursor derecha	una línea arriba
Cursor arriba	una línea abajo
Cursor abajo	
<b>Otras teclas de control</b>	Página de ayuda
<b>(Teclado numérico en el 464)</b>	Nuevos parámetros de disco
0                   (F0 CPC664-6128)	Cambio de página
1                   (F1 CPC664-6128)	Modo de edición HEX/ASCII
2                   (F2 CPC664-6128)	Escribir sector al disco
3                   (F3 CPC644-6128)	Fin del programa
4                   (F4 CPC664-6128)	
5                   (F5 CPC644-6128)	

```

10 REM DISCEDIT
20 REM (C) AMSTRAD SEMANAL

30 REM
40 MODE 2:INK 0,1:INK 1,26:PAPER 0
50 MEMORY &7FFF:GOSUB 1270:' POKE MAC
HINE CODE
60 GOSUB 630:' INITIALISE
70 WINDOW SWAP 0,1
80 GOSUB 780:' DRAW SCREEN
90 CLS
100 T=0:CC=0:CR=0:WD=4:WI=24:OFF=64:V
4=D$
110 PRINT CHR$(23);CHR$(1);TAG #WD:G
OSUB 540:POKE &908C,&80:GOTO 320
120 S$="":F=0:WHILE S$="":C=0:F=F XOR
1:MOVE OFF+CC*WI+T*8,334-CR*16:PRINT
#WD,CHR$(143):WHILE S$="" AND C<20:
C=C+1:GOSUB 750:WEND:WEND:IF F=1 THEN
MOVE OFF+CC*WI+T*8,334-CR*16:PRINT #
WD,CHR$(143);
130 ADDRESS=PEEK(&908B)+PEEK(&908C)*2
56+CC+CR*16
140 IF ASC(S$)<237 THEN 350
150 IF ASC(S$)<239 THEN GOSUB 1230:IF
M$="N" THEN 120
160 IF ASC(S$)=237 THEN K$="write":GO
SUB 480:GOTO 120
170 IF ASC(S$)=238 THEN 1250
180 IF ASC(S$)=244 THEN 1140
190 IF ASC(S$)=246 THEN 1200
200 IF ASC(S$)=247 THEN GOSUB 920:GOS
UB 780:GOSUB 820:GOTO 330
210 CC=0:CR=0:T=0:CLS

```

```

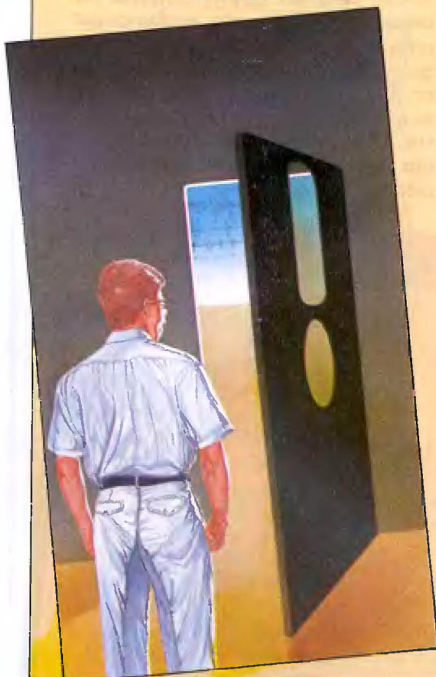
220 IF ASC(S$)=245 THEN POKE &908C,PE
EK(&908C)XOR 1:GOTO 330
230 IF ASC(S$)=251 THEN sector=sector
+1
240 IF ASC(S$)=250 THEN sector=sector
-1
250 IF ASC(S$)=248 THEN track=track+1
260 IF ASC(S$)=249 THEN track=track-1
270 IF ASC(S$)=244 THEN GOSUB 540:POK
E &908C,&80
280 IF sector>FSN+R THEN sector=FSN:t
rack=track+1
290 IF sector<FSN THEN sector=FSN+R:t
rack=track-1
300 IF track<0 THEN track=39
310 IF track>39 THEN track=0
320 K$="read":GOSUB 480:IF PEEK(&90E3
)<>0 THEN 120
330 LOCATE #1,8,3:PRINT#1,drive:LOCAT
E #1,19,3:PRINT#1,track:LOCATE #1,31,
3:PRINT#1,sector:LOCATE #1,42,3:PRINT
#1,PEEK(&908C)AND &7F:LOCATE #1,58,3:
PRINT#1,A$
340 LOCATE #2,1,1:WINDOW SWAP 0,2:TAG
OFF #WD:CLS#3:CLS#4:CLS#5:CALL &9000:
TAG #WD:WINDOW SWAP 0,2:GOTO 120
350 IF T=1 AND WD=4 THEN POKE ADDRESS
,(PEEK(ADDRESS) AND &F0)+VAL("&"+S$)
360 IF T=0 AND WD=4 THEN POKE ADDRESS
,(PEEK(ADDRESS) AND &F)+VAL("&"+S$)*1
6
370 IF WD=5 THEN POKE ADDRESS,ASC(S$)
380 LOCATE #1,9+CC*3,5+CR:K$=HEX$(PEE
K(ADDRESS)):IF LEN(K$)=1 THEN PRINT#1
,"0";
390 PRINT#1,K$:LOCATE #1,59+CC,5+CR:K
$="":IF PEEK(ADDRESS)>31 THEN K$=CHR
$(PEEK(ADDRESS))
400 PRINT#1,K$;
410 IF WD=4 THEN T=T XOR 1:IF T=0 THE
N CC=CC+1
420 IF WD=5 THEN CC=CC+1
430 IF CC=16 THEN CC=0:CR=CR+1
440 IF CC=-1 THEN CC=15:CR=CR-1
450 IF CR=16 THEN CR=0
460 IF CR=-1 THEN CR=15
470 GOTO 120
480 J=PEEK(&90E5):U=PEEK(&90E6)
490 POKE &90E5,track:POKE &90E4,drive
:POKE &90E6,sector:POKE &90E3,0
500 IF ASC(K$)=114 THEN CALL &90B1 EL
SE CALL &90C3
510 IF PEEK(&90E3)<>0 THEN CLS:PRINT"
Disk ";K$;" error, Track: ";track;"Sec
tor: ";sector:CHR$(7):track=J:sector=U
:T$="":WHILE T$="":T$=INKEY$:WEND:CLS
:RETURN
520 IF ASC(K$)=119 THEN PRINT"Sector
written successfully";CHR$(7)

```

```

530 RETURN
540 GOSUB 820
550 CLS:LOCATE 1,1:PRINT"Input Drive:
";nc=1:mi=0:ma=1:GOSUB 840:drive=VA
L(W$):LOCATE #1,8,3:PRINT#1,drive
560 IF drive=0 THEN !A ELSE !B
570 FSN=PEEK(PEEK(DPH+16*DRIVE))+PEEK(
DPH+1+16*DRIVE)*256+15)
580 R=8:IF FSN=&41 THEN a$="SYSTEM" E
LSE IF FSN=&C1 THEN a$="DATA " ELSE
a$="IBM " :R=7
590 LOCATE #1,58,3:PRINT#1,a$
600 CLS:LOCATE 1,1:PRINT"Input Track:
";nc=2:mi=0:ma=39:GOSUB 840:track=V
AL(W$):LOCATE #1,19,3:PRINT#1,track
610 CLS:LOCATE 1,1:PRINT CHR$(22);CHR
$(1);"Input Sector (";CHR$(8);FSN;CHR
$(8);"-";CHR$(8);FSN+R;CHR$(8);)": ";
CHR$(22);CHR$(8);nc=3:mi=FSN:ma=FSN+
R:GOSUB 840:sector=VAL(W$):LOCATE #1,
31,3:PRINT#1,sector;" "
620 CLS:RETURN
630 CALL &908E:'init machine code
640 DPH=PEEK(&BE40)+PEEK(&BE41)*256+1
8:D$="0123456789ABCDEF"
650 KEY DEF 14,0,245:KEY DEF 5,0,246
660 KEY DEF 15,0,247:KEY DEF 20,0,237
670 KEY DEF 12,0,238:KEY DEF 13,0,244
680 KEY DEF 2,1,241,249,249
690 KEY DEF 0,1,240,248,248
700 KEY DEF 8,1,242,250,250
710 KEY DEF 1,1,243,251,251
720 POKE &908B,0:POKE &908C,&80
730 WINDOW #1,1,80,22,25:WINDOW #2,2,
80,5,25:WINDOW #3,2,6,5,20:WINDOW #4,
9,5,5,20:WINDOW #5,59,75,5,20
740 RETURN
750 S$=INKEY$:IF (S$<CHR$(237) OR S$)
CHR$(252)) AND INSTR(" "+V$,UPPER$(S$)
)<2 AND LEN(V$)>0 THEN S$=""
760 IF V$<>" " THEN S$=UPPER$(S$)
770 RETURN
780 CLS #1:PEN #1,1:LOCATE #1,1,1:PRI
NT#1,"Amstrad Disc Sector Editor."
790 MOVE 0,380:DRAWR 216,0
800 MOVE 0,72:DRAWR 0,272:DRAWR 600,0
:DRAWR 0,-272:DRAWR -600,0:MOVER 40,0
:DRAWR 0,272:MOVER 405,0:DRAWR 0,-272
810 RETURN
820 LOCATE #1,1,3:PRINT#1,"Drive:
Track: Sector: page:
Format: ":RETURN
830 nc=2:mi=2:ma=10
840 W$=""
850 S$="":WHILE S$="" OR LEN(W$)=nc A
ND S$<>CHR$(127) AND S$<>CHR$(13) OR
S$=CHR$(127) AND W$="" :S$=INKEY$:WEND
860 IF S$=CHR$(13) THEN 900
870 IF S$=CHR$(127) THEN W$=LEFT$(W$,

```



```

LEN(W$)-1):PRINT CHR$(8);CHR$(16);
880 IF S$(CHR$(127)) THEN PRINT S$;W
$=W$+S$
890 GOTO 850
900 IF VAL(W$)<MI OR VAL(W$)>MA THEN
FOR A=1 TO LEN(W$):PRINT CHR$(8);CHR$(
16);NEXT:PRINT CHR$(7);GOTO 840
910 RETURN
920 WINDOW SWAP 0,1:CLS
930 LOCATE 31,1:PRINT CHR$(24);" *COM
MAND MENU* ";CHR$(24)
940 LOCATE 1,3:PRINT CHR$(24);" Movin
g to new track or sector:";CHR$(24)
950 PRINT" <SHIFT>"+CHR$(243)+" ...
move forward one sector."
960 PRINT" <SHIFT>"+CHR$(242)+" ...
move back one sector."
970 PRINT" <SHIFT>"+CHR$(240)+" ...
move forward one track."
980 PRINT" <SHIFT>"+CHR$(241)+" ...
move back one track."
990 LOCATE 1,9:PRINT CHR$(24);" Editi
ng commands:";CHR$(24)
1000 PRINT" "+CHR$(243)+" ... move r
ight one byte."
1010 PRINT" "+CHR$(242)+" ... move l
eft one byte."
1020 PRINT" "+CHR$(241)+" ... move d
own one line."
1030 PRINT" "+CHR$(240)+" ... move u
p one line."
1040 LOCATE 1,15:PRINT CHR$(24);" Oth
er command keys (numeric KEY pad):";C
HR$(24)

```

```

1050 PRINT" f0 ... Help."
1060 PRINT" f1 ... Select new disk p
arameters."
1070 PRINT" f2 ... Page toggle, (tog
gles display between bytes"
1080 PRINT" 0-2
55 & 256-511 of a sector)."
1090 PRINT" f3 ... Toggles between H
EX & ASCII for editing."
1100 PRINT" f4 ... Writes sector to
disk."
1110 PRINT" f5 ... Exit program."
1120 LOCATE 30,24:PRINT CHR$(24);" PR
ESS SPACE TO EXIT ";CHR$(24);S$="":WH
ILE S$("<" ):S$=INKEY$:WEND
1130 CLS:WINDOW SWAP 0,1:RETURN
1140 IF ASC(S$)=243 THEN 410
1150 IF ASC(S$)=242 AND WD=4 THEN T=T
XOR 1:IF T=1 THEN CC=CC-1
1160 IF ASC(S$)=242 AND WD=4 THEN 430
1170 IF ASC(S$)=242 AND WD=5 THEN CC=
CC-1:GOTO 430
1180 IF ASC(S$)=240 THEN CR=CR-1:GOTO
450
1190 CR=CR+1:GOTO 450
1200 TAGOFF #WD:WD=WD XOR 1:TAG #WD:T
=0
1210 IF WD=4 THEN WI=24:OFF=64:V$=D$
ELSE WI=8:OFF=464:V$=""
1220 GOTO 120
1230 IF ASC(S$)=237 THEN K$="Save sec
tor" ELSE K$="Quit program"
1240 LOCATE 1,1:PRINT K$;" ARE YOU SU
RE (Y/N)?";CHR$(7);H$="":WHILE INSTR

```

```

("Y/N",H$)(2:H$=UPPER$(INKEY$)):WEND:C
LS:RETURN
1250 TAGOFF #WD:PRINT CHR$(23);CHR$(0
):MODE 1:CALL &BB00
1260 END
1270 IF PEEK(&9000)=&DD THEN RETURN
1280 RESTORE 1360
1290 CHECKSUM=0
1300 FOR G=&9000 TO &90DC
1310 READ B$:POKE G,VAL("&"+B$)
1320 CHECKSUM=CHECKSUM+VAL("&"+B$)
1330 NEXT
1340 IF CHECKSUM=30232 THEN RETURN
1350 PRINT" M/C DATA ERROR";CHR$(7)
1360 DATA DD,2A,8B,90,DD,E5,C1,78,E6,
7F
1370 DATA 32,8D,98,CD,62,90,79,32,8D,
90
1380 DATA CD,62,90,DD,E5,06,03,CD,81,
90
1390 DATA 06,10,DD,7E,00,32,8D,90,CD,
62
1400 DATA 90,3E,28,CD,5A,B8,DD,23,10,
EE
1410 DATA 06,02,CD,81,90,DD,E1,06,10,
DD
1420 DATA 7E,00,FE,20,02,45,90,3E,2E,
C5
1430 DATA CD,5D,B8,C1,DD,23,10,ED,3E,
0D
1440 DATA CD,5A,B8,3E,0A,CD,5A,B8,DD,
E5
1450 DATA C1,3E,00,B9,C2,04,90,C9,21,
8D
1460 DATA 90,ED,6F,CD,70,90,ED,6F,CD,
78
1470 DATA 90,C9,E6,0F,C6,30,FE,3A,FA,
7B
1480 DATA 90,C6,07,C5,CD,5A,B8,C1,C9,
C5
1490 DATA 3E,09,CD,5A,B8,C1,10,F7,C9,
00
1500 DATA 00,00,DD,21,DD,90,21,0B,90,
CD
1510 DATA D4,BC,DD,75,00,DD,74,01,DD,
71
1520 DATA 02,21,DC,90,CD,04,BC,DD,75,
03
1530 DATA DD,74,04,DD,71,05,C9,21,00,
00
1540 DATA ED,5B,E4,90,3A,E6,90,4F,DF,
DD
1550 DATA 90,D2,D5,90,C9,21,00,00,ED,
5B
1560 DATA E4,90,3A,E6,90,4F,DF,E0,90,
D2
1570 DATA D5,90,C9,3E,FF,32,E3,90,C9,
84
1580 DATA 85

```

