

Az Országos Commodore Egyesület lapja

1992/4



*A jövő
beviteli
készülékei*

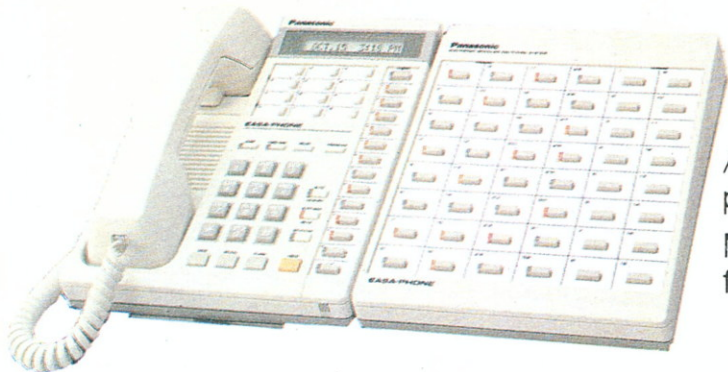
**Adataink
élettere**

Neto TRADING

Telefon Szalon

Budapest, XIII.,
Fürst S. u. 14/b.
Tel.: 132-8739 131-5573

PANASONIC telefonközpontok
szereléssel is,
3–12 fővonal,
8–32 mellék.



Amerikai és magyar szabványú
postai csatlakozók, átalakítók
Memóriás és kulcsos telefonok,
főnök-titkári rendszer

Kisméretű asztali fax 44 500 Ft+ÁFA

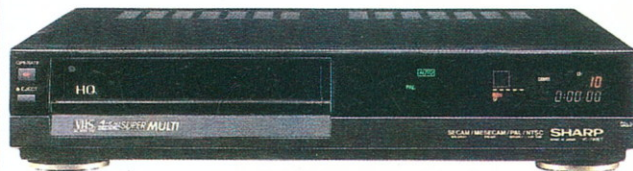
SHARP és SUPRA televíziók, videók, kamkorderek

VIDEÓK:

VC-90 ET titánbevonatú
6 fejes,
Hifi sztereó 69 400 Ft

VC-7900 ET titánbevonatú
4 fejes
sztereó,
multi super 48 800 Ft

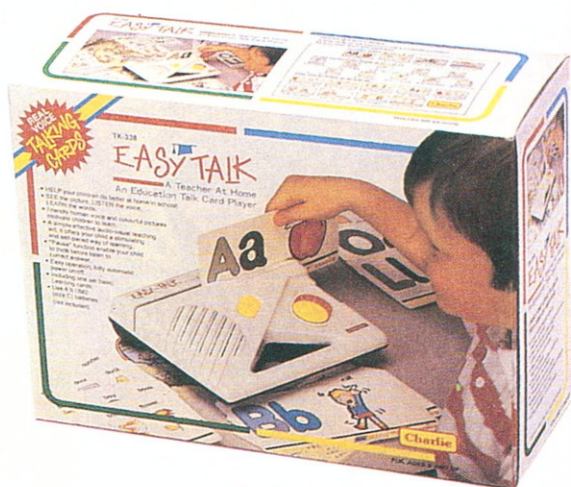
VC-6V3 DP lejátszó
felvételi
lehetőség 22 400 Ft



KAMKORDEREK:

VL-C 6400 beépített monitor
8-szoros zoom 87 400 Ft

VL-C 7500 beépített
színes monitor
12-szeres zoom 110 000 Ft



EASY-TALK az IDEÁLIS HÁZITANÁRI!

Új audió-vizuális
angol nyelvoktatókészülék
gyermekek részére

MIT, HOGYAN, HOL, MIKOR?

EGYESÜLETI ÜGYEK: Egyesületünknek tagja lehet mindenki, aki a tagsági díjat befizeti. A tagdíjat személyesen az egyesület irodájában (1025 Budapest, Vöröstorony utca 29. Telefon: 1-76-22-57), vagy átutalással az MNB 217-98 292, OTP 565-3610-8 számlára lehet befizetni. Megrendelés esetén szám-lát küldünk.

Pötyögőszolgálatunk valamint a szervizkedvezmény és az apróhirdetés lehetősége tagjaink rendelkezésére áll.

A **DEÁKPÁHOLY** tagjai minden hónapban megkapják a C-újságot, a tagsági díj egy egy évre 777 forint.

A **PLUSZPÁHOLY** tagjai minden hónapban megkapják a C-újságot, és kapnak havonta 3 db vásárlási utalványt. A tagsági díj egy évre 1888 Ft.

A **SZUPERPÁHOLY** tagjai havonta 15 példányt kapnak a C-újságból, és ezzel havonta 15x3 db vásárlási utalványt is. Az éves tagsági díj 20 900 Ft.

ÜGYFÉLFOGADÁS: Minden kedden és csütörtökön 12–16 óra között várjuk tagjainkat és az érdeklődőket.

PÖTYÖGŐSZOLGÁLAT: Az újságban megjelenő programokat másolja a megrendelők részére. Megrendelhető személyesen az egyesület irodájában vagy postai utánvétellel. Postacím: 1388 Budapest 62., Postafiók: 86.

APRÓHIRDETÉS: Az egyesületi tagoknak ingyen áll rendelkezésre. Nem tagoknak a hirdetés ára 100 forint. A hirdetés módja: az újságban megjelenő nyomtatvány kitöltésével.

A **C-ÚJSÁG RÉGEBBI SZÁMAI** megvásárolhatók az egyesület irodájában, vagy megrendelhetők utánvétellel.

Kedvezményes ár! Tagoknak olcsóbb!

Az újságban eddig megjelent programok gépenként össze-
gyűjtve megrendelhetők. VC 20, C16, PLUS/4, C128, C64. To-
vábbi felvilágosítást is adunk a 1-76-22-57-es telefonszámon vagy
levélben!

Vidéki pluszpáholy-tagjaink háromhavi tikkett összegyűjté-
sekor igénybe vehetik a NOVOTRADE 2C Áruház csomagküldő
szolgálatát.

VIDÉKEN TOVÁBBI INFORMÁCIÓK KAPHATÓK:

Baja, AXIS Kft.,
Győri Bartók Béla Művelődési Ház,
Jászberényi Városi Könyvtár,
Kecskemét, SZIGMA—BIT,
Pécsi Apáczai Csere János Gimnázium,
Zalaegerszegi Ságvári Endre Gimnázium.

Az Országos Commodore Egyesület módszertani kiadványa

Egyesületi iroda és szerkesztőség:
1025 Budapest, Vöröstorony utca 29. Telefon: 1-76-22-57
Felelős kiadó: Horváth Judit, az egyesület elnöke
Főszerkesztő: Rados Péter, az OCE főtítkára
Felelős szerkesztő: dr. Horváth András
Művészeti szerkesztő: Bausz Sándor
Levélcím: Commodore Újság, 1388 Budapest, 62. Pf.: 86.
Index: ISSN 0237-756 X
Terjeszti a Magyar Posta
Megvásárolható a hírlapárusoknál
92.0027 MSZH Nyomda és Kiadó Kft., Budapest
Felelős vezető: Nagy László

Kedves Tagtársaink, Olvasóink!

Egyesületi és szerkesztőségi CÍMÜNK MEGVÁLTOZOTT!

1992. február 1-jétől új címünk:

1025 Vöröstorony utca 29.

Telefon: 1-76-22-57.

Megközelíthető a Batthyány térről a 11-es autó-
busszal. A Kapy utcai megállónál kell leszállni.
Elképzelhető, hogy a költözés miatt kicsit
akadozni fog a kapcsolattartás Önökkel. Ezért
megértésüket és türelmüket kérjük. Lapunk
megjelenésében fennakadás nem lesz.

Postacímünk változatlan:

1388 Budapest 62., Postafiók: 86.

OCE

Pötyögőszolgálat

A kiválasztott programok megrendelhetők
levélben, utánvétellel.

Postacím:

1388 Budapest 62., Postafiók: 86.

Felvilágosítás telefonon:

1-76-22-57.

Tisztelt Szerkesztőség!

Még csak néhány hónapja vagyok tagja az Egyesületnek,
de a C-újság már most nagyon tetszik. Úgy érzem, hogy a ha-
zai hobby-gépes felhasználók többsége valamelyik Commodore
gépet használja, ezért szükség van egy lapra, amely ezzel a
gépcsaláddal foglalkozik. Szerintem a Commodore géptulajdo-
nosok többségének a véleményét mondom, amikor azt mon-
dom: szükség van az Egyesületre. Szükség van rá azért, hogy
ne kelljen egyedül, magányosan, esetleg kis csoportokban ta-
nulni ezeknek a gépeknek a használatát, hanem legyen egy fó-
rum, ahol tippeket, ötleteket adhatnánk át egymásnak.

Külön dicsérem „A gépi kódú programozás Commodore
gépeken” című sorozatot, mert ha már az olvasók találnak az
újságban Assembly listákat, azokat ne csak legépelni tanulják
meg, hanem saját programban is hasznát vegyék. Ehhez nagy-
szerű segítség, ha az újság indít egy ilyen irányú sorozatot.
Ezért tartom kiváló ötletnek a sorozatot. Én ugyan már olvas-
tam a Mikrovilág Magazin hasonló jellegű sorozatát, de az nem
tért ki bővebben a hardware részére és csak a C-64-es gépre vo-
natkozó adatokat közölt. Nekem ugyan 64-esem van, de jobban
örülök, ha egy cikket több géptípusra is írnak, mert így nem
csak az adott gép tulajdonosai olvassák el, hanem mindenki
tanulhat belőle valamit.

Seftsik Péter

Tisztelt Szerkesztőség!

A mellékelt lemezen elküldöm Önöknek két programomat.
Az első: TOLOGATÓ.

Már a nevéből is kitűnik, hogy ez a program Kovács János hasonló című — 1991/7—8. számban megjelent — programjának átdolgozása; ezúttal C-64-re (Standard Basic).

Megírására az készítetett, hogy nagyon tetszett a játékötlet.

Kezelése: a téglalapokra itt is a bal felső sarkukkal kell hivatkozni, a koordináták felcserélhetőek — tehát A4 helyett 4A-t is elfogad. Miután kijelöltük, nyomjuk meg a RETURN-t.

Amenynyiben a kijelölés tényleg egy lapocska bal felső sarkát adta meg, a lapocska ki fog világosodni. Ellenkező esetben hibajelzést kapunk. A téglácska új helyét hasonlóképpen kell megadni. A játékot fel lehet adni a ” ” gombbal.

A második egy régen készült programom:

ABC RENDEZŐ

Képes egy max. 255 bejegyzésből álló szöveget sorba rendezésére, nyomtatására, lemezre írására, illetve olvasására.

Remélem, hogy a kezelése egyértelmű lesz.

Tisztelettel:

Abinéri Gábor

```
0 REM *****
1 REM * PROGRAM: TOLOGATO SORSZ: *
2 REM * C=64 *
3 REM * IRTA: ABINERI GABOR *
4 REM *****
5 GOTO51
6 POKE211,X:POKE214,Y:SYS58640:RETURN
7 GETA#:IFA#=""THEN7
8 IFA#>="A"ANDA#<="E"THEN13
9 IFA#>="1"ANDA#<="4"THEN15
10 IFA#=CHR#(13)THEN17
11 IFA#=""THEN46
12 GOTO7
13 PRINT"#####":Q=ASC(A#)-64:W=1024+Q*80:POKEW,PEEK(W)+128:LY=Q
14 GOTO7
15 X=3:Y=14:GOSUB6:PRINT"1 2 3 4":Q=ASC(A#)-48:W=1585+2*Q:POKEW,PEEK(W)+128
16 LX=Q:GOTO7
17 IFLX=0ORLY=0THEN7
18 PRINT"#####1 2 3 4":IFE=1THEN21
19 IFP<LX,LY>=5ORP<LX,LY>=0THENGOSUB35:GOTO7
20 S=13:L=P<LX,LY>:GOSUB50:OX=LX:OY=LY:LX=0:LY=0:E=1:GOTO7
21 L=P<OX,OY>:V=(L-1)AND2)/2:F=(L-1)AND1:G=OX:H=OY
22 IFLX-OX<0ANDLY-OY<0THENGOSUB35:GOTO7
23 R=ABS((-LX-0X)>0)+(LY-OY)>0)*-2):ONRGO24,27
24 U=ABS(LX-0X):FORT=1TOU:P(G,H)=0:P(G+V,H)=0:P(G,H+F)=0:P(G+V,H+F)=0
25 G=G+SGN(LX-0X):IFP(G,H)<0ORP(G+V,H)<0ORP(G,H+F)<0ORP(G+V,H+F)<0THEN32
26 NEXT:GOTO30
27 U=ABS(LY-OY):FORT=1TOU:P(G,H)=0:P(G+V,H)=0:P(G,H+F)=0:P(G+V,H+F)=0
28 H=H+SGN(LY-OY):IFP(G,H)<0ORP(G+V,H)<0ORP(G,H+F)<0ORP(G+V,H+F)<0THEN32
29 NEXT:GOTO30
30 P<LX+V,LY>=5:P<LX,LY+F>=5:P<LX+V,LY+F>=5:P<LX,LY>=L:X=2*OX+1:Y=2*OY:GOSUB6
31 PRINTT$(L):S=6:GOSUB50:E=0:GOTO37
32 T=U:NEXTT:GOSUB35
33 P<OX+V,OY>=5:P<OX,OY+F>=5:P<OX+V,OY+F>=5:P<OX,OY>=L
34 LX=OX:LY=OY:S=6:GOSUB50:LX=0:LY=0:E=0:GOTO7
35 X=3:Y=19:GOSUB6:PRINT"### HIBA ###":POKE198,0:WAIT200,63:POKE781,19
36 SYS59903:LX=0:LY=0:RETURN
37 IFLX=1ANDLY=4ANDL=4THEN40
38 EP=EP+1:X=10:Y=16:GOSUB6:E#=STR$(EP)
39 PRINT"###RIGHT$( "000"+RIGHT$(E#,LEN(E#)-1),3):LX=0:LY=0:GOTO7
40 X=10:Y=23:GOSUB6:PRINT"###MEGNYERTED A JATEKOT!"
41 X=15:Y=21:GOSUB6:PRINT"###GRATULALOK":I=1
42 FORI=1TO12:GOSUB6:PRINTMID$(S#,I,1)"GRATULALOK"
43 GETA#:IFA#<>""THENI=12:NEXT:GOTO45
44 NEXT:GOTO42
45 GOSUB6:PRINT" ### POKE781,23:SYS59903
46 X=0:Y=21:GOSUB6:PRINT"###JJ JATEK (I/N)":POKE198,0
47 GETA#:IFA#="I"THENRUN
48 IFA#<>"N"THEN47
49 END
50 X=2*LX+1:Y=2*LY:GOSUB6:POKE199,1:POKE646,S:PRINTL$(L):RETURN
51 :POKE53281,0:POKE53280,0:POKE53272,20:PRINT"###":DIMP(5,6)
52 L$(4)="#####"
```



```

53 L$(3)="          " : L$(2)="          " : L$(1)="          "
54 T$(4)="          " : T$(3)="          "
55 T$(2)="          " : T$(1)="          "
56 FORI=1T05:FORL=1T04:READP(L,I):NEXTL,I
57 FORI=0T05:P(I,0)=5:P(I,6)=5:NEXT:FORI=1T05:P(0,I)=5:P(5,I)=5:NEXT
58 PRINT"          " : FORI=1T010:PRINT"          " : NEXT
59 PRINT"          " : PRINT"          "
60 PRINT"LEPESEID: 000" : S$="          "
61 FORLX=1T04:FORLY=1T05:L=P(LX,LY):S=6:GOSUB50:NEXTLY,LX
62 X=21:Y=0:GOSUB6:PRINT"          -TOLOGATO-          " : X=15:Y=2:GOSUB6
63 PRINT"KATONA JANOS NYOMAN IRTA:" : X=19:Y=4:GOSUB6
64 PRINT"          " : PRINTTAB(19)"          "
65 PRINTTAB(19)"          " : PRINTTAB(19)"          "
66 PRINTTAB(19)"          " : PRINTTAB(19)"          "
67 PRINTTAB(19)"          " : PRINTTAB(17)"          "
68 PRINTTAB(17)"ELJUTTATHI A BAL ALSO" : PRINTTAB(17)"SAROKBA.A LAPOCSKAKRA"
69 PRINTTAB(17)"A BAL FELSO SARKUKKAL" : PRINTTAB(17)"KELL HIVATKOZNI."
70 GOT07
71 DATA4,5,3,5,5,5,3,5,1,1,0,0,2,2,3,5,5,5,3,5
    
```

READY.

```

0 REM *****
1 REM * PROGRAM:ABC RENDEZO SORSZ: *
2 REM *          C=64 *
3 REM *          IRTA:ABINERI GABOR *
4 REM *****
5 DIMP$(261):I=1:POKE53281,11:POKE53280,11
6 GOT076
7 FORY=SHT02STEP-1:POKE781,Y:SYS59903:NEXT:RETURN
8 POKE781,SL:SYS59903:RETURN
9 POKE198,0:POKE214,24:POKE211,15:SYS58640:PRINT"          SZOKOZ          " : WAIT203,60:RETURN
10 POKE214,23:SYS58640:PRINT"          " : RETURN
11 INPUT#15,H1$,H2$,H3$,H4$:CLOSE15:H$="          "+H1$+", "+H2$+", "+H3$+", "+H4$+"
12 PRINT"          " : SPC((40-LEN(H$))/2):H$:CLOSE9:RETURN
13 PRINT"          " : SPC(12)"          ADATBEVITEL          " : GOSUB10:POKE214,24:SYS58640
14 PRINT"          " : F1          ADATBEVITEL VEGE          " : F7          HIBA          " : INT((I-1)/10)+1"          LAP          "
15 IF(I-10*INT(I/10))-1<0THEN33
16 PRINT"          " :
17 GETA$:IFA$=""THEN23
18 IFA$=CHR$(13)THEN25
19 IFA$=CHR$(20)THEN31
20 IFA$="          "THEN76
21 IFA$="          "THEN24
22 IFA$<"          OR          "          "THEN17
23 Q$=Q$+A$:PRINT"          " : A$:          " : GOT017
24 Q$="" : SL=PEEK(214):GOSUB8:POKE211,0:SYS58640:PRINT"          " : GOT017
25 IFQ$=""THEN17
26 PRINT"          " : IFI>255THEN41
27 IFLEN(Q$)>40THENPRINT"          " : IFLEN(Q$)>80THENQ$=LEFT$(Q$,80):PRINT"          "
28 P$(I)=Q$:I=I+1:Q$=P$(I):IFPEEK(214)<21THENPRINT:PRINTQ$          " : GOT017
29 SN=21:GOSUB7:PRINT"          " : (I-1)/10+1"          LAP          "
30 PRINT"          " : GOT017
31 IFLEN(Q$)>1THENQ$=LEFT$(Q$,LEN(Q$)-1):PRINT"          " : GOT017
32 Q$="" : SL=PEEK(214):GOSUB8:PRINT:PRINT"          " : GOT017
33 C1=10*INT((I-1)/10):IFC1=0THENC1=1
34 FORC=C1TOI:PRINTP$(C):PRINT:NEXT:PRINT"          " : GOT017
35 PRINT"          " : SPC(12)"          ADATRENDEZES          "
36 POKE204,0:PRINT"          " : POKE56325,255
37 FORU=1TOI:FORC=1TOI
38 IFP$(U)<P$(C)THENW$=P$(C):P$(C)=P$(U):P$(U)=W$
39 NEXTC,U:FORC=1TOI:P$(C)=P$(C+1):NEXTC:POKE207,0:POKE204,1
40 POKE56325,60:GOT076
41 PRINT:PRINT"          " : BETELT A MEMORIA          " : SL=24:GOSUB8:GOSUB9:GOT076
42 PRINT"          " :          ADATKIIRAS          " : GOSUB10:POKE198,0
    
```



```

43 PRINT"Képernyő 1 ■ KEPERNYŐRE":PRINT"Képernyő 2 ■ NYOMTATORA"
44 PRINT"Képernyő 3 ■ VISSZA A FOMENURE
45 GETA#:ONVAL(A#)GOTO46,51,76:GOTO45
46 SN=8:GOSUB7:PRINT"Képernyő"
47 FORC=0TO1STEP20:FORO=0TO19
48 E#=P#(C+O+1)+ "
49 PRINTLEFT$(E#,39):NEXTO
50 GOSUB9:SL=24:GOSUB8:SN=22:GOSUB7:NEXTC:GOTO76
51 PRINT"J"SPC(10)" ADATOK NYOMTATASA ":PRINT"MI":PRINT"!";POKE204,0
52 OPEN4,4,0:FORC=1TO1:PRINT#4,P#(C):NEXT:CLOSE4:POKE207,0:POKE204,1:GOTO76
53 POKE198,0:PRINT"J"SPC(13)" SAVE/LOAD "":PRINT"Képernyő 1 ■ ADATOK KIMENTESE"
54 PRINT"Képernyő 2 ■ ADATOK BETÖLTÉSE":PRINT"Képernyő 3 ■ DIRECTORY"
55 PRINT"Képernyő 4 ■ VISSZA A FOMENURE
56 GETA#:ONVAL(A#)GOTO57,62,67,76:GOTO56
57 PRINT"J"SPC(10)" ADATOK KIMENTESE ":GOSUB10
58 PRINT"Képernyő"SPC(19)"":PRINT"MI LEGYEN A CIME ":INPUTE#
59 E#=" "E#+" " "E#=LEFT$(E#,16)
60 OPEN15,8,15:OPENS,8,8,"@:"E#+"S,W":PRINT#8,I
61 FORC=1TO1:PRINT#8,P#(C):NEXTC:PRINT#8," ":GOSUB11:GOSUB9:GOTO76
62 PRINT"J"SPC(10)" ADATOK BETÖLTÉSE ":GOSUB10
63 PRINT"Képernyő"SPC(19)"":PRINT"MI A FILE CIME ":INPUTE#
64 E#=" "E#+" " "E#=LEFT$(E#,16)
65 OPENS,8,8,E#+"S,R":OPEN15,8,15:INPUT#8,E
66 FORC=1TOE:INPUT#8,P#(C):NEXTC:I=E:GOSUB11:GOSUB9:GOTO76
67 PRINT"J"TAB(13)" DIRECTORY ":GOSUB10:PRINT:POKE56325,255
68 OPEN1,8,0,"#":POKE781,1:SYS65479:GETA#,A#
69 GETA#,A#:IFST=64THENSYS65484:CLOSE1:GOTO74
70 GETA#,B#:PRINT"MI"ASC(A#+CHR$(0))+256*ASC(B#+CHR$(0));
71 GETA#:PRINTA#:IF A#<>" "THEN71
72 PRINT:IFPEEK(214)<22THEN69
73 GOSUB9:POKE198,0:SL=24:GOSUB8:SN=22:GOSUB7:PRINT"Képernyő":GOTO69
74 GOSUB9
75 POKE56325,60:GOTO76
76 PRINT"J"SPC(16)" FOMENU "
77 PRINT"Képernyő 1 ■ ADATBEVITEL":PRINT"Képernyő 2 ■ ADATRENDEZES"
78 PRINT"Képernyő 3 ■ ADATOK KIIRASA":PRINT"Képernyő 4 ■ SAVE/LOAD"
79 GETA#:ONVAL(A#)GOTO13,35,42,53:GOTO79

READY.
    
```

C nyelv Commodore-osoknak

Miért programoznánk mindig csak BASIC-ben? Még a Commodore gépek is kínálnak más lehetőséget. Az egyik, a PC-ken szinte standardként használatos C nyelv. A Profi C segítségével mi is megtehetjük az első lépéseket.

A Profi C-nek két változata létezik — egy C64-es és egy C128-as verzió. Mindkét változat azonos utasításkészlettel rendelkezik, különben is szinte teljesen azonosak. Szerencsére azonban a 128-as verzióban a gép jobb felépítéséből és nagyobb tárolókapacitásából adódó előnyöket teljesen kihasználták. Így a C128-as tulajdonos a Profi C-t természetesen a 80 karakteres üzemmódban használhatja, a C-lemezeket autostarttal láthatja el, vagy például FAST-módban dolgozhat. Egy komplett tárolóbank a C128 esetében RAM floppyként is használható, ami a compiler és linker funkció alkalmazása-

kor jelentős sebességnövekedést eredményez. Természetesen a C-rendszer 128-as változata maga is öntöltő képességű lemezen van, azaz csak a lemezt kell behelyezni és a gépet bekapcsolnunk. Rögtön ezután a részletes használati utasítás segítségével már tovább is léptünk, és megírhatjuk első programunkat.

Parancsok — mint a CP/M-nél

Először a beépített C-parancsprocesszor töltődik be a C64/ C128 tárába.

Ez nem más, mint egy saját kis üzemmód. A parancsprocesszorral történő első találkozás rögtön nagy hasonlóságot sejtet a CP/M-mel: A C rendszer „A”-val jelentkezik be. Az „A” betű azt jelenti, hogy az „A” floppy meghajtó működésre kész (megfelel a 8-as egységnek). A lemezkezelés teljesen saját parancs alapján történik, melyeket a parancsprocesszor tartalmaz. Mint a CP/M-ben megszokott, a tartalomjegyzék (DIR)-re listázható, és egy adatblokk tartalma a (TYPE) segítségével íratható a képernyőre.

A forrásprogram előállításához a Profi C lemezen található egy külső editor, mely a parancsprocesszorra történő megfelelő utalással hívható. A felhasználó rendelkezésére áll egy szövegszerkesztő, amely nemcsak kifejezett szövegszerkesztési funkciókat kínál, de egy extra oldalt is, amely pl. a compilálás után előállított hibaadatoknak helyet biztosít. Az editor szövegszerkesztő funkciói a követke-

zők: Eltolás, másolás, átkapcsolás az extralapra, törlés, sőt még egy keresés és csere funkció is. Az éppen szerkesztett program nyomtatási listája még az öreg programozó rókákat is elkényezteti. Minden oldal elején megtalálható a program neve, az oldalszám és a dátum. Ezenkívül minden sor számozott. Ez az editorban nincs így, az ugyanis, mint a legtöbb szövegfeldolgozó program, a parancssorban található sor/oszlop kijelzéssel dolgozik. Az editor, úgy mint a programcsomag többi része is, képes a RAM floppyval (csak C128) együttműködni. Ha a szerkesztő alapjelkészlete nem tetszik, ezt meg is változtathatjuk. Három különböző karakterkészlet áll rendelkezésünkre.

A kész program compilálásához az editorból ki kell lépni, ekkor automatikusan ismét a parancsprocesszor jelentkezik. Ezután hívhatjuk a compilert, amely a forrásadatár megnevezését kéri. A compiler egy ún. link-modult állít elő, nem kész gépi kódot. A C-ben írt program elindítása előtt szükséges még a program linkelése. Ez folyamat során véglegesen futtatható kódok előállítását történik, ami azt jelenti, hogy a program már csak járulékos modulokat tartalmaz, mint pl. a ki/bevitel vezérlését.

A RAM floppy

A C128-asba integrált RAM floppy a parancsprocesszor betöltésekor automatikusan az első tárbankba kerül és a compilálásához szükséges legfontosabb adatokkal feltöltődik. Ez azt jelenti, hogy a programozáshoz már csak a nullás tárlap 64 kbyte memóriája áll rendelkezésünkre. Ennek ellenére rövid használat után a RAM floppyt nélkülözhetetlennek fogjuk tartani. Rögtön a RAM floppyba töltődik a (Stdio.h) nevű C-alapkönyvtár, melyben fontos végrehajtási utasítások és a link funkció futtatásához szükséges adatok vannak definiálva. Némely program is tárolható a RAM floppyban. Ez elsősorban a teszteléskor jelentős, mert így a tesztelésre kerülő program nagyon gyorsan az editorba tölthető. A compilálás is gyorsabban történik a RAM floppy alkalmazása esetén. A RAM floppy egész tartalma egy speciális parancsprocesszor utasításal lemezen tárolható vagy onnan visszatölthető. Természetesen minden egy állományba kerül. Visszatöltéskor az erre szolgáló utasítás a megfelelő adatrészeket ismét az eredetinek megfelelően választja szét. Ha egy forráskódot kívánunk a RAM floppyból a lemezre vinni, ez az editoron keresztül minden gond nélkül megoldható.

A nyelv

A Profi C csaknem a teljes C nyelvet magában foglalja, beleértve a felhasználó által definiálható adattípusokat is. Emellett tartalmaz néhány, a gépre jellemző bővítést. Így pl. a C128 esetében mindkét, a 80 karakteres kijelzésért felelős VDC-chip kontrollregiszter saját utasításokon keresztül elérhető. Ami az adatbázis programozóknak nagyon hiányzik, az az adatok tetszés szerinti elérhetősége. A szekvenciális file-ok kezelése ezzel szemben teljesen megoldott. Még a hibacsatorna kiolvasása sem okoz nehézséget a kézikönyvben mellékelt leírás segítségével. Ezenkívül a felhasználó rendelkezésére áll egy átfogó könyvtár, amely segítségével nagyobb programozói feladatok is megoldhatók.

Ami mindenképpen nagyon pozitív, az a grafikus utasítások átfogó könyvtára. A lemezen mellékelt demoprogram a lefordítás után a számos grafikus lehetőség egy részét mutatja be (oszlopdiaagram, analóg óra). Aki szívesen programoz strukturáltan, és emellett még a számítógépes grafikát is szereti, a Profi C-ben mindkettőt megtalálja. A rendszerprogramozóknak lehetőséget biztosít a mutatókkal történő munkához. Ezek segítségével konkrét tárhelyekhez nyúlhatunk és velük dolgozhatunk. Ez a C nyelvet egészen a gépi kódú programozás közelébe helyezi. Aki szívesen alkalmazza a strukturált programozást és ennek ellenére gépi kódhoz közel kíván dolgozni, egy hatékony módszerhez jut ennek megvalósítására. Kétségtelenül utalnunk kell itt arra, hogy a mutatókkal való munkánál a legnagyobb elővigyázatosságra van szükség. A tárhelyek manipulálásakor figyelmetlenség esetén meglehet, magát a programot rongáljuk meg.

Saját programok

Akit a Profi C parancsai nem elégítenek ki, minden további nélkül saját utasításokat írhat. A C nyelv úgy lett megírva, hogy bármikor saját parancsokat állíthatunk elő. A járulékos parancsok a CP/M-hez hasonlóan lefordított formában lemezen kerülnek tárolásra, és mint a normál, beépített parancsok bármikor felhasználhatók.

Hamar rájöhettünk, hogy a profi C programok csak a parancsprocesszoron belül futnak. Tehát nem írhatunk olyan C-rutinokat, amelyek önállóan futhatnak és pl. SYS-paranccsal hívhatók. Természetesen egy C-programból készíthetünk önállóan futó programot, amely először a

parancsprocesszort, majd a programot tölti és futtatja. Ezen túlmenően az így compilált programok mindig egy bonyolult címképpel jelentkeznek be. Mindez pedig bizonyos játékoktól eltekintve a compiler felhasználhatóságát igen szűk határok közé szorítja. Ez különösen kár, mert a Profi C különben egy igen jólsikerült, profi felhasználóknak is megfelelő szoftver lenne.

Részletes kézikönyv

A Profi C német kézikönyve (350 oldal) igen átfogó és informatív. A kezdőnek egyben tankönyvül szolgál a C nyelv megtanulásához, mivel egy részletes gyakorlati részt is tartalmaz, amely nem fukarkodik a példákkal. Számos beépített funkció nagyon szemléletesen kerül leírásra. A RAM floppy és a parancsprocesszor kezelésének pontos leírása könnyíti meg a programmal történő első munkát.

Elvárható lenne a standard C-könyvtár nyomtatásban történő megjelentetése. Aki még sosem hallott a C nyelvről, jó ha gondoskodik egy saját C könyvről, ugyanis a Profi C kézikönyve számos esetben bizonyos alapismeretek meglétét feltételezi.

Összefoglalva

Összegzésképpen elmondhatjuk, hogy a Profi C-vel egy igen hatékony C64/C128-as fejlesztőprogram áll rendelkezésünkre, amely még sok kényelmet és gyorsaságot is biztosít. Annak idején a program 99 márkába került, s magam is találok ennek Ft-ra „felszorozott” árú példányaival a bizományi boltjaiban. Ma ezt Becker C-t legföljebb barátainktól szerezhettük be, hiszen már a Data Becker cég sem foglalkozik C64/128-as programokkal.

Szolnoki Béla

Pluszok:

- teljes C nyelv
- részletes grafikakönyvtár
- a C128-ason a RAM floppy használható
- komfortos szövegeditor
- a rendszer saját utasításokkal bővíthető
- részletes és sokoldalú kézikönyv
- igen jó ár/szolgáltatási viszony
- biztonsági másolat kapható

Mínuszok

- az eredeti lemezek másolás ellen védve
- a programoknak az utasításprocesszort tartalmazniuk kell

Régi mozdulat, új lehetőségek

Cserélhető merevlemez

A számítógépek világában lassan mindennapossá vált a merevlemez (winchester lemez) használata háttértárként.

Az IBM PC kompatibilis gépek rohamos terjedésével a merevlemez teret hódított a floppy lemezzel szemben, elsősorban tároló kapacitása miatt, ami több tízszerese vagy akár százszoros is lehet egy floppy lemez kapacitásának. Ugyanakkor a merevlemez sebessége írás/olvasásnál egy nagyságrenddel nagyobb mint a floppy lemezé, ezzel is jobban kihasználva az egyre bonyolultabb és gyorsabb mikroprocesszorok műveleti készségét.

Egy jó tulajdonság azonban elveszni látszott a winchester lemez használatával: a könnyed lemezcsere, egy jól ismert mozdulattal lemezt cserélni és pillanatok alatt teljesen új lappal indulni a következő lemezzel, még akkor is ha ez a lemez már sokkal többre képes mint a floppy lemez. A winchester lemez ugyanis beépített a számítógépbe, nem biztosítva cserelhetőséget.

Ezt a hiányosságot hivatott pótolni a SyQuest Technology amerikai cég cserélhető merevlemeze, amely képes ötvözni a hagyományos merevlemez paramétereit a cserélhetőség minden előnyével.

A kétféle tároló kapacitással kínált cserélhető merevlemez (44 MegaByte és 88 MegaByte) könnyen beépíthető bármely IBM PC kompatibilis gépbe. A merevlemez meghajtó mérete megegyezik az 5.25 inch-es 1.2 MegaByte-os floppy meghajtó méretével.

Novotrade
SZERVÍZ Kft.

A vezérlő kártya egy ma már szabványnak számító SCSI (Small Computer Standard Interface) kártya. Ilyen szabványú illesztési lehetőség fellelhető már az újabb házi számítógépeknél is (Amiga 2000, Amiga 3000).

A 44 MegaByte-os SQ 555 típusú és a 88 MegaByte-os SQ 5110 típusú cserélhető merevlemez meghajtó egyaránt 20 ms átlagos elérési idejű, két író/olvasó fejjel, beépített szervo felülettel. A lemez porvédett műanyag kazettában helyezkedik el. A meghajtó üzemel első merevlemezként vagy fix merevlemez mellett bármilyen sorrendben.

A SyQuest Technology cserélhető merevlemeze kitűnően alkalmazható adatmentésre és biztonságos tárolásra, adatszervezésre (nagy adatbázisok, CAD rendszerek, DTP), ugyanakkor hagyományos merevlemezként is használható.

A magyarországi forgalmazást felvállaló Novotrade Szervíz Kft. több mint egyéves tapasztalatot szerzett a cserélhető merevlemez beszerelésében és üzemeltetésében. A tesztek kiváló minőségről tanúskodnak.

Mindent összevetve nyugodt szívvel ajánlható a SyQuest Technology cserélhető merevlemeze.

Bogocs György

TUNGSRAM-MAX[®]

A Tungsrám Magnetic Media RT.-t 1990 áprilisában alapították vegyes vállalként. Fő részvényesünk a General Electric (USA) — Tungsrám RT.

A cég hagyományos és nagy sűrűségű mágneslemezek nemzetközi szabvány szerinti (ANSI, ISO, ECMA) gyártásával és forgalmazásával foglalkozik.

A Tungsrám-Max lemezeket több mint 20 országba exportáljuk. Magyarországon a piaci részesedésünk 20% körül van. Belföldi értékesítési munkánkban alapvető szempontnak tekintjük azt, hogy ha van egy világszínvonalú magyar termék, mint a Tungsrám-Max lemez, akkor a magyar nagyfelhasználó cégek, költségvetési intézmények ezt használják. Így is támogatva az egyébként rendkívül nehéz helyzetben lévő magyar ipart és gazdaságot.

Különösen nagy hangsúlyt fektetünk arra, hogy az oktatási intézményeket és tanulóikat viszonylag alacsony áron kiváló minőségű floppy lemezekhez juttassuk, hiszen ők lesznek azok, akik ha átveszik ezt a Széchenyi korát idéző szemléletet, sokat tehetnek Magyarországotért.

Ennek érdekében felvettük a kapcsolatot az ország egyetemével, főiskolaival, számítástechnikai képzést nyújtó középiskolaival. Az oktatási intézményekben kihelyezett vásárokat szervezünk. Különböző alapítványokon keresztül támogatjuk is ezeket az iskolákat.



Lexikon

III. rész

File

Ejtése „fájl”. Magyarul ez annyit tesz: állomány, tárolt adathalmaz. A felhasználó szempontjából egy állomány összetartozó információk gyűjteménye, amelyet egybefüggően valamilyen adathordozón (diszk, kazetta) lehet megtalálni. Egy ilyen állomány lehet például egy tárolt program. De a relatív és a szekvenciális file is ide tartozik. Nagy általánosságban azt lehet mondani, hogy az állomány egy bizonyos név alatt egy adathordozóra mentett adatok halmaza. Az teljesen mellékes, hogy ez a file szöveg, egy személyi adatgyűjtemény, egy HiRes kép vagy egy program-e. Speciálisan a lemez-meghajtót tekintve a file egyes adatblokkok egymás után történő rendbesorolása.

Ha a Commodore lemez-meghajtónkról beolvassuk a tartalomjegyzéket (directory), akkor a listázáskor megnézhetjük, hány file is található a lemezen.

Lemez-meghajtó

Sok házi számítógép esetében a lemez-meghajtó alatt olyan bővítő készüléket kell érteni, amely információ tárolására szolgál, és nem tartozik a gép alapkiépítéséhez. Ezek használatához szükség van egy különleges operációs rendszerre, amely ellátja az egység vezérlését. Ez az a bizonyos DOS (Disk Operating System). A Commodore lemez-meghajtók (1541, 1551, 1570, 1571 és 1581) esetében már be van építve a saját operációs rendszer, így a számítógépnek ezzel már nem kell foglalkozni. Ezek a meghajtók saját mikroprocesszorral rendelkeznek, ezért „intelligens” egységeknek is lehet tekinteni őket. Intelligensnek azért, mivel a számítógépnek csupán egy parancsot kell feléjük küldeni, a továbbiakban a meghajtó önállóan dolgozik tovább és nem tartja

föl a computert. Az információk tárolására kis, mágnesezhető lemezek szolgálnak, amelyeket diszknek (floppy diszknek) is neveznek.

Floppy-speeder

A floppy-speeder magyarul lemez-meghajtó gyorsító rendszer. A Commodore lemez-meghajtók, a 1541, a 1570 és a 1571 (ez utóbbiak a C64-es módban) a leglassúbb ilyen egységeknek minősülnek, ami csak létezik. Ez többek között az egyes állományok igen lassú betöltésében és tárolásában mutatkozik meg. Éppen ezért az idők során különböző floppy-gyorsítók fejlesztettek ki, amelyek ezt a sebességet már elviselhető mértékre növelik. Vannak olyan rendszerek, amelyek olyan gyorsak, hogy azok az állományok, amelyeket az eredeti rendszer hosszú percekig tölt be, ezután másodpercek alatt a számítógépbe kerülnek. Ugyanakkor ezek részben igen drágák is. Ez például abból adódik, hogy a hardveres ráfordítás a párhuzamos speederek, RAM floppyk esetében igen nagy, és ez alaposan mege-meli a fejlesztés és rendszer építőelemeire eső költségeket.

Bővítő modul

A szokásos módszer egy működőképes programnak a számítógép tárolójába való betöltésére az, hogy azt beolvastatjuk a mágneslemezzel vagy a kazettáról. Ilyenkor azonban a program hosszúságától függően meglehetősen nagy várakozási idővel lehet számolni (ha mondjuk nincs floppy vagy kazettás gyorsítónk). Azonban lehetőség van arra is, hogy a programokat egy vagy több EPROM-ba vigyük, majd ezeket egy alkalmas kapcsolás segítségével a számítógép bővíthe-

tő (expansions) kapujába dugjuk. Ez a kapcsolat, amelyik az említett EPROM-okat tartalmazza a modul. Ezek lehetnek játékok, felhasználói szoftverek vagy BASIC bővítések is.

Ezeknek a moduloknak az a legnagyobb előnyük, hogy a bennük tárolt programok azonnal a számítógép bekapcsolása után a felhasználó rendelkezésére állnak, és legtöbbször nem vesznek el helyet az egyébként szűkös meglévő tárolóból.

Illesztőhely (interface, Schnittstelle)

Ez a fogalom a számítógépes körben általánosságban két, egymással kapcsolatban álló rendszer találkozási helye. A szakirodalom kétféle illesztőhelyről szokott beszélni, egyrészt az ember-gép, másrészt a gép-gép illesztőhelyről. Az előbbi esetében (amelyet felhasználói interfésznek is mondanak) a számítógép kezelői elemekre kell gondolni (billentyűzet és képernyő), de ugyaneide tartozik a programoknak az a része, amelyik a felhasználóval kommunikál. A szoftverek esetében ehhez még a felhasználói felület kifejezést is használják. A másik viszonylatban, a gép-gép interfésznel arról a helyről van szó, amelyen keresztül az információ forgalom bonyolódik a számítógép és a periféria között.

Például igen sok nyomtató (amelyek a Centronics szabvány szerint működnek, mint pl. az Epson, a Fujitsu vagy a Star) használatához egy nyomtató interfészre (illesztő-adapterre) van szükség. Ez az adapter végzi el azután a különböző jelek „illesztését”.

Expansions port

Ez a fogalom magyarul a bővítő portot (kaput) takarja. Erre a csatlakozásra a C64/C128 buszrendszerének szinte minden vezetéke ki van véve. Ez azt jelenti, hogy ott az összes olyan fontos jel a rendelkezésre áll, amelyek a számítógép vezérléséhez nélkülözhetetlenek. Ezzel pedig igen sok mindent lehet kezdeni. A szabadon hozzáférhető cím- és adatbusz segítségével a számítógéphez bővítő modulokat és EPROM-okat könnyen csatlakoztatni lehet. Éppígy akár egy másik kernalt is csatlakoztathatunk anélkül, hogy a gépet szét kellene szednünk.

A C64/C128-as bővítő portját manapság főleg a párhuzamosan dolgozó floppy-gyorsítók (Prologic DOS, TurboAccess, Dolphin DOS, Jiffy DOS stb.) a játé- és

bővítő modulok használják. Más hasonló, ismert felhasználások például a MIDI vagy IEEE illesztések.

Userport

Ez a fogalom magyarul a felhasználó számára meghatározott illesztőhelyet (kaput) takarja. A C64/128 eme csatlakozása egy változatos célokra használható port. Azért változatos, mivel az itt található 8 darab adatvezeték tetszőlegesen és egymástól függetlenül lehet be- vagy kivezető vonallá kapcsolni. Így például ezen keresztül mondjuk mérési eredményeket lehet fölvenni, amelyet egy program földolgoz és kiértékel. Vagy mondjuk ezt a nyolc vezeték kimenetként programoz-

zuk, és ezzel vezéreljük pl. a hűtőszekrényünket, a rádiós vekkert vagy akár egy fényorgonát. Elméletileg itt mindenféle vezérlésre mód van. A userport (ejtsd „júzerport”) felhasználásában azonban főleg a Centronics nyomtatók, modemek és az akusztikus csatolók (RS232) illesztése dominál. De ezt a kaput használják a floppygyorsítók (speederek) is.

Directory

Azaz a lemez tartalomjegyzéke. A lemezes adattárolás egyik nagy előnye a kazettával szemben, hogy ott a rendelkezésünkre áll a tartalomjegyzék, amelyet meg lehet tekinteni. A 1541-es floppynál ezt a LOAD "\$",8 parancs kiadásával ér-

hetjük el, és azt a LIST paranccsal nézhetjük meg. Ily módon lehetőségünk van arra, hogy megállapítsuk, hány file (adatállomány) van a lemezen, azok mennyi helyet foglalnak el, hogy van-e még szabad hely, de még az állományok típusát is (REL, PRG,USR, SEQ és DEL) meghatározhatjuk. Ezen kívül a directory megengedi azt, hogy minden egyes állományt közvetlenül kiválasszunk, mivel az is föl van jegyezve, hogy a file hol is található a lemezen. Ez pedig igen nagy könnyítés a kazettás tárolással szemben, mivel ott a szalagot mindig oda kell hajtani, ahol az adott állomány ténylegesen található, ez pedig időrabló és nehézkes dolog.

(Folytatjuk)

NOVOTRADE—2C Kft. ÁRLISTA

Hardverek

C64 alapgép	14 600 Ft
VC 1541 drive	16 600 Ft
Datasette	3 000 Ft
Amiga 500 alapgép	49 990 Ft
Amiga 2000 alapgép	125 600 Ft
Amiga mouse	4 000 Ft
C 1084S sztereómonitor	32 000 Ft
C 1802 monitor	25 000 Ft
C64 mouse	3 500 Ft
Amiga RF modulátor	3 900 Ft
Amiga tárbővítő (512 Kb)	8 000 Ft
Amiga AT kártya	47 675 Ft
Amiga digitalizáló	29 700 Ft

C64 játékok neve

	Kazetta	Lemez
Chamonix Challenge	499 Ft	549 Ft
Eddie Edwards Super Ski	499 Ft	549 Ft
Eszkimó	345 Ft	—
Hostages	549 Ft	599 Ft
Impossible Mission II	581 Ft	668 Ft
Diamond/I Want More	—	549 Ft
Nautilus	399 Ft	—
Ninja Testvérek	399 Ft	—
Operation Neptune	—	599 Ft
Prohibition	499 Ft	—
Rettenthetetlen	390 Ft	—
Rolling Twins	399 Ft	549 Ft
Sim City	—	599 Ft
Smaragdvár	345 Ft	—
Space Knight	340 Ft	—
Space Racer	—	549 Ft
Tin Tin On The Moon	—	599 Ft
Warlock Quest	499 Ft	549 Ft
Waterpolo	450 Ft	—
Xonox	399 Ft	499 Ft

Hardverkiegészítők

2 RCA kábel	410 Ft
3 RCA kábel	575 Ft
Hálózati kábel	480 Ft
Adatkábel (soros bus)	360 Ft
Antennakábel	340 Ft
Antenna váltókapcsoló	490 Ft
C64 Euro-kábel	685 Ft
Amiga Euro-kábel	1250 Ft
Ékezetes Eprom SP-180	2545 Ft
Ékezetes Eprom MPS 1230	3500 Ft
Mikrokapcsolós joystick	1000 Ft
Műszerész porszívó	490 Ft
C64 tápegység	3500 Ft
1541 tápegység	2700 Ft
14" monofilter	990 Ft
14" colourfilter	1100 Ft
Lemeztartó 3 db-os 5,25"	99 Ft
Lemeztartó 50 db-os 5,25"	700 Ft
Lemeztartó 100 db-os 5,25"	820 Ft
Lemeztartó 10 db-os 3,5"	160 Ft
Lemeztartó 40 db-os 3,5"	700 Ft
Lemeztartó F80 db-os 3,5"	850 Ft
Lemeztartó 140 db-os 3,5"	1300 Ft
Mouse pad	245 Ft
Festékszalag FX-1050	675 Ft
Festékszalag MPS-1230	700 Ft
Display monitorszemüveg	750 Ft

Viszonteladók! Felhasználók!

*Dyras festékszalagok
nagy választékban és olcsón
kaphatók!*

Gépi kódú programozás Commodore gépeken

(C+4, C16,
VIC-20, C64,
C116 és C128)

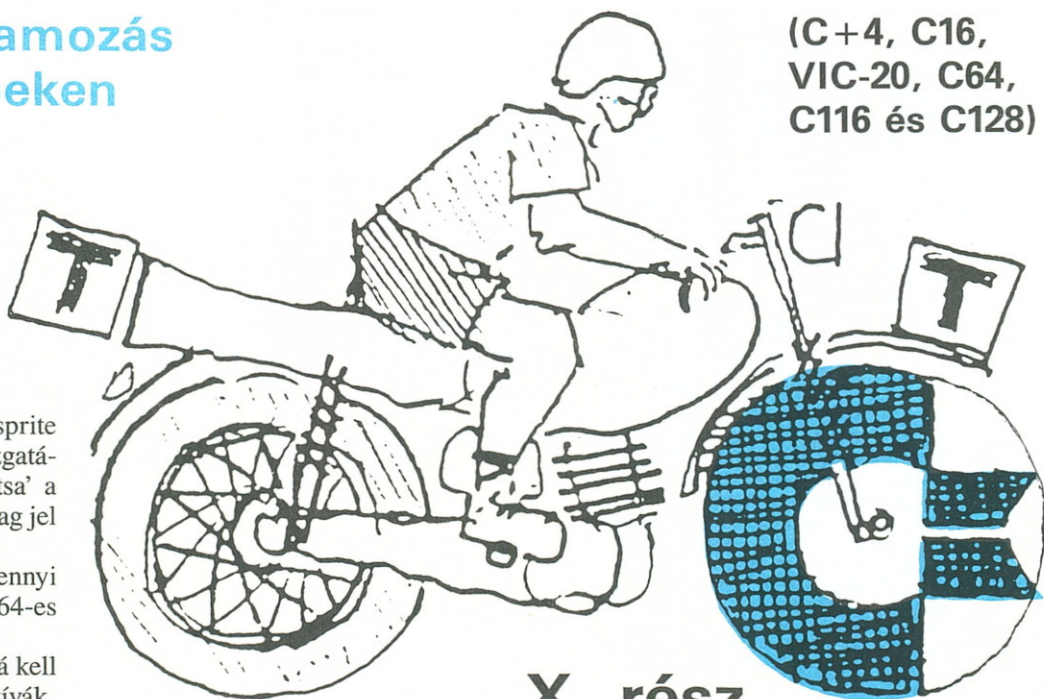
CSILLAG- VÁGTA

Mostani példánkat a software-sprite kezelés ihlette. Cél egy figura mozgatása a képernyőn úgy, hogy 'ne bántsa' a háttérrel. Legyen ez a figura egy csillag jel (*).

Ez a program működik valamennyi C-gépen, kivétel a C-128, ahol csak 64-es üzemmódban fut.

Az assembler kezeléséhez hozzá kell fűzni pár szót. Léteznek ún. direktívák, melyek a fordítónak adott parancsok. Ezek a fordítás menetét vezérlik. Leggyakoribbak a <.BYT> és <.WOR> (a <.> a direktívák része, tulajdonképpen ismertetőjelük). E két parancs helyet foglal a memóriában, és azt feltölti az általunk megadott értékkel. A <.BYT> után byte-okat kell megadni, a <.WOR> után 16 bites ún. szavakat. Persze mindhárom számrendszerrel használhatjuk. Ha címkét teszünk a direktívák elé, akkor az ott elhelyezett adatokra azzal a címkével tudunk hivatkozni. Az (<'>) (aposztróf)-ok közé zárt jelenek az ASCII kódját fordítja le az assembler. Ezt a <.BYT> direktívában is lehet alkalmazni, ahol szövegek beírására kiválóan alkalmas.

A C-gépek eltérő képernyő mérete miatt gépkódban meg kell szenvednünk. Ahhoz, hogy könnyen és gyorsan megtudjuk a (*) X és Y kordinátáiból a képernyő memóriának azt a címét, ahova ki kell írunk, szükségünk lesz egy táblázatra, mely tartalmazza a képernyősorok elejének címeit. Így elkerülhető a szorzás a (*) kiírásakor. Egy pozíció kiszámításához elég a megfelelő sorcím kiolvasása a táblázatból (indexelt címzés mód), majd egy további összeadás (még egy indexelés), melyben az oszlop számot adjuk a címhez. A táblázat kiszámításakor sem végzünk szorzást, mindig az előző táblázat elemhez adjuk hozzá a sorhosszúságot. Ez cseles, mert a 16 bites cím és 8 bites sorhossz összeadásnál a cím felső byte-ját az XR-ben tartjuk és akkor adunk hozzá egyet (INX), ha a C flag 1 (BCC → C flag = 0 → átugorjuk az INX-et). Kissé zavaró lehet még, hogy a cím alsó byte-ját az AC-ban tároljuk és azt ki kell írni és



X. rész

vissza kell tölteni a TEMP változóból, mert a táblázat feltöltésére használt indexelt címzéssel csak az AC-t lehet a memóriába írni, az XR-t nem. Sajnos ezen bűvészkedések, mármint a regiszterek egymásba másolgatása, el- és visszamenetése gyakori, mert sok hasznos művelet csak bizonyos regiszterekkel végezhető el.

A főprogram (a 47-es sortól) az elmondottak szerint feltölti az YR-t az X

kordinátával (XKOR), az XR-t az Y kordinátával (YKOR). Ez utóbbi alapján a táblázatból kiolvassa és az AC-n keresztül a ZPL-ZPH byte-okra (ami egy 16 bites nulladiklapos mutató) írja a sor kezdőcímét. A ZPL-t YR-rel indirekt indexelve (az ott lévő címhez YR-t hozzáadva) betölti az AC-ba a (*) leendő helyén álló betűt, majd ugyanide kiírja a (*)-ot.

Ezzel a (*) megjelent. Most várjunk egy billentyűre. A leütött billentyűnek

○	100000	DATA	4C,3C,60,00,00,28,19,0C,0135	○
○	100001	DATA	00,00,00,00,00,00,00,00,0000	○
○	100002	DATA	00,00,00,00,00,00,00,00,0000	○
○	100003	DATA	00,00,00,00,00,00,00,00,0000	○
○	100004	DATA	00,00,00,00,00,00,00,00,0000	○
○	100005	DATA	00,00,00,00,00,00,00,00,0000	○
○	100006	DATA	00,00,00,00,00,00,00,00,0000	○
○	100007	DATA	00,00,00,00,20,ED,FF,0E,029A	○
○	100008	DATA	05,60,8C,06,60,20,F3,FF,0369	○
○	100009	DATA	00,0C,F0,0C,00,FD,F0,04,0549	○
○	100010	DATA	A9,1E,00,06,A9,0C,00,02,0324	○
○	100011	DATA	A9,04,80,07,60,AA,00,00,02EB	○
○	100012	DATA	A9,00,99,0A,60,8D,09,60,02A2	○
○	100013	DATA	8A,99,23,60,AD,09,60,18,02D4	○
○	100014	DATA	6D,05,60,90,01,E8,C8,CC,03DF	○
○	100015	DATA	06,60,00,E6,AC,03,60,AE,03D9	○
○	100016	DATA	04,60,8D,0A,60,85,14,8D,02E1	○
○	100017	DATA	23,60,85,15,B1,14,8D,08,0277	○
○	100018	DATA	60,A9,2A,91,14,20,E4,FF,03DB	○
○	100019	DATA	AC,03,60,C9,90,F0,12,C9,0440	○
○	100020	DATA	1D,F0,24,C9,91,F0,C9,00,0477	○
○	100021	DATA	11,F0,40,C9,20,00,E6,F0,0400	○
○	100022	DATA	4D,AD,03,60,00,06,AD,05,02E5	○
○	100023	DATA	60,8D,03,60,CE,03,60,AD,032E	○
○	100024	DATA	08,60,91,14,18,90,85,EE,0358	○
○	100025	DATA	03,60,AD,03,60,CD,05,60,02A5	○
○	100026	DATA	00,05,A9,00,8D,03,60,18,0286	○
○	100027	DATA	90,E5,AD,04,60,00,06,AD,0409	○
○	100028	DATA	06,60,8D,04,60,CE,04,60,0289	○
○	100029	DATA	18,90,04,EE,04,60,AD,04,037F	○
○	100030	DATA	60,CD,06,60,00,C9,A9,00,03D5	○


```

10031 DATA 80,04,60,18,90,C1,AD,08,030F
10032 DATA 60,91,14,60,00,00,00,00,0165
20000 AS$="0123456789ABCDEF"
20001 DIMH(ASC("F"))
20002 FORI=0TO15:H(ASC(MID$(AS$,I+1)))=I
20003 NEXT
20004 BA=6*4096
20005 PRINT"KONTROLL INDUK !"
20006 FORI=0TO32:C=0
20007 FORB=0TO7
20008 READA$:GOSUB20029
20009 C=C+A
20010 NEXTB:PRINT". ";
20011 READB$:A$=LEFT$(B$,2):GOSUB20029
20012 B=A:A$=RIGHT$(B$,2):GOSUB20029
20013 IFB*256+A<>CTHENPRINT"ADATHIBA A"10000+I
      -EDIK SORBAN !!!":STOP
20014 NEXTI
20015 PRINT:PRINT"KONTROLL KESZ !"
20017 RESTORE
20018 FORI=0TO32
20019 FORB=0TO7
20020 READA$:GOSUB20029
20021 POKEBA+I*8+B,A
20022 NEXTB:PRINT". ";
20023 READA$
20024 NEXTI
20025 PRINT:PRINT"OK."
20026 PRINT"INDITASA ."
20027 PRINT"SYS"BA"
20028 END
20029 H=H(ASC(A$)):L=H(ASC(RIGHT$(A$,1)))
20030 A=H*16+L:RETURN
    
```

megfelelően meg kell változtatni az XKOR vagy YKOR-t, az előző helyre vissza kell írni a betűt, amit a PUFF változóban tároltunk és újra előállítva a táblázatból a címeket beolvasni az új helyen álló betűt, kiírni a (*)-ot ... és így tovább...

Amennyiben a szóköt nyomtuk meg, a program visszaírja az eltárolt betűt és visszatér BASIC-be.

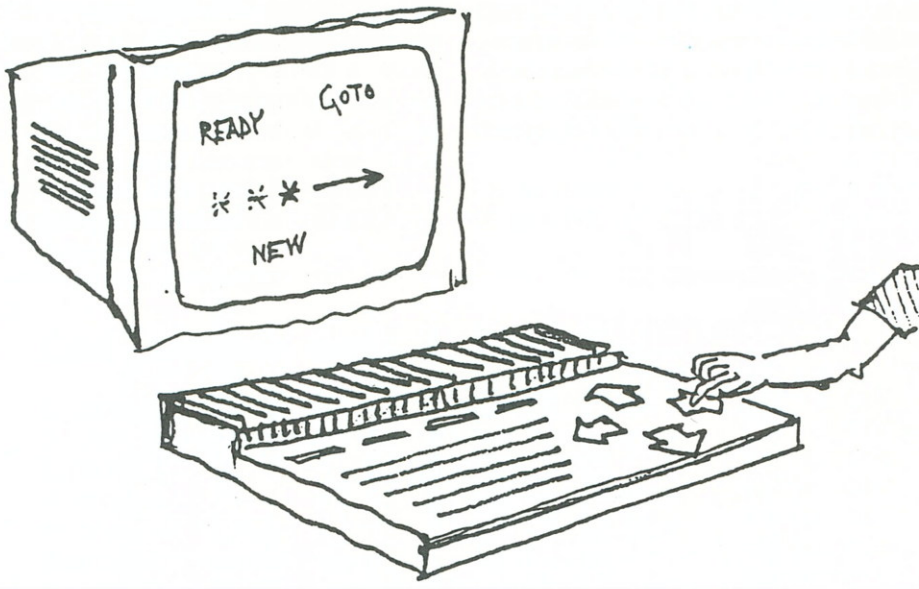
Érdekes azon eltűnődni, hogy az YR és az XKOR, bár hasonló szerepet tölt be, mégsem azonos. Az YR-ben az eltárolt betű X kordinátája van, míg az XKOR a csillag helyét jelenti. Pontosabban azt, ahol lennie kell. Miután a képernyőt helyreállítottuk (a PUFF-ot visszaírtuk) ez a kettő azonos lesz, de csak addig, amíg egy billentyű hatására odébb nem kell mozdulni.

Egyszerű trükk a CLC és BCC utasítások egymás után írása, ténylegesen ez egy feltétel nélküli relatív ugrást valósít meg, ami egyébként nincsen az utasításkészletben. Értelme, hogy azt a programot, amely csak relatívugrásokat tartalmaz, egy monitorbeli T (transzfer, mozgató) paranccsal más címre lehet másolni, úgy hogy ott hibátlanul le is fut. Sajnos az adatterület marad a helyén az abszolút címek miatt. Itt ezek az ugrások inkább stílus és szokás értékűek, egyszerű JMP is megfelelne.

A kordináták csökkentése egyszerűen annyi, megvizsgáljuk 0-e, ha igen akkor a képszelességet (ill. magasságot) állítjuk be, és mindentől függetlenül levonunk egyet (DEC). Emiatt körbe fog szaladni a (*) az ernyőn. A növelésnél hasonló a helyzet, de ott a növelés után nézzük meg kicsúszunk-e a képből, ha igen, nullázzuk a kordinátát, vagyis bejövünk a túloldalra.

Nem szoltunk még az operációs rendszerből (KERNAL) itt felhasznált rutinokról. Az IOBASE rutin az I/O terület kezdőcímét adja meg, ezzel kényelmes beazonosítani, melyik gépen is fut a program. Ezt a képernyő kezdőcímének beállítására használjuk. A rutin az XR (alsó byte) és YR-ben (felső byte) adja meg a címet. A felső byte C+4-en \$FD C64-en \$DC és VC20-on \$90. A második rutin, a SCREEN, a képernyő méretét adja meg. Az YR-ben a sorok, az XR-ben pedig az oszlopok számát kapjuk vissza. A GETIN rutin a billentyűzet puffereiből olvas, magyarárn visszaadja az AC-ban a leütött billentyű ASCII kódját. Ha nincs ilyen a Z flas értéke 1 a rutin lefutása után, vigyázat a rutin mindegyik regisztert használja (vagyis elrontja ami előtte bennük volt).

Nagy Tamás



```

00001 0000      ;Put"demo1.src
00003 0000      *= $5000
00004 6000 4c 3c 60      jmp init
00005 6003
00006 6004
00007 6005
00008 6006
00009 6007
00010 6008
00011 6009
    
```




0	00012	600a		zpl	= #14	mutató a képre	0
	00013	600a		zph	= zpl+1		
	00014	600a		;			
0	00015	600a		scrlo	* = *+25	alsó byte-ok	0
	00016	6023		scrhi	* = *+25	felső byte-ok	
	00017	603c		;			
0	00018	603c	20 ed ff	init	jsr \$ffed	screen KERNAL	0
	00019	603f	8e 05 60	stx xmax		szélesség	
	00020	6042	8c 06 60	sty ymax		magasság	
0	00021	6045	20 f3 ff	jsr \$fff3		iobase KERNAL	
	00022	6048	c0 dc	cpy #\$dc			0
	00023	604a	f0 0c	beq c64			
0	00024	604c	c0 fd	cpy #\$fd			0
	00025	604e	f0 04	beq Plus4			
	00026	6050	a9 1e	wc20	lda #\$1e	wc20 bázis	0
	00027	6052	d0 06	bne bazis			
0	00028	6054	a9 0c	Plus4	lda #\$0c	c-16, plus/4	0
	00029	6056	d0 02	bne bazis			
	00030	6058	a9 04	c64	lda #\$04	c-64	
0	00031	605a	8d 07 60	bazis	sta scrbas	megvan a Kezdet	0
	00032	605d	a3	tax			
	00033	605e	a0 00	ldy #0			
0	00034	6060	a9 00	lda #0			0
	00035	6062	99 0a 60	szamol	sta scrlo,y	LO(Y)=	
	00036	6065	8d 09 60	sta temp		(XMAX*Y)AND255	
	00037	6068	8a	tax		HI(Y)=	
0	00038	6069	99 23 60	sta scrhi,y		(XMAX*Y)/256	0
	00039	606c	ad 09 60	lda temp			
	00040	606f	18	clc			
0	00041	6070	6d 05 60	adc xmax			0
	00042	6073	90 01	bcc szamel			
	00043	6075	e8	inx			
0	00044	6076	c8	szamel	iny		0
	00045	6077	cc 06 60	cpy ymax			
	00046	607a	d0 e6	bne szamol			
0	00047	607c	ac 03 60	maszik	ldy xkor	itt kezd mászni	0
	00048	607f	ae 04 60	ldx ykor			
	00049	6082	bd 0a 60	lda scrlo,x			
	00050	6085	85 14	sta zpl			
0	00051	6087	bd 23 60	lda scrhi,x			0
	00052	608a	85 15	sta zph			
	00053	608c	b1 14	lda (zpl),y		ami ott volt..	
0	00054	608e	8d 08 60	sta puff		..azt eltesszük	0
	00055	6091	a9 2a	lda #'*		..csillagot meg	
	00056	6093	91 14	sta (zpl),y		..kitesszük	
0	00057	6095	20 e4 ff	bill	jsr \$ffe4	getin KERNAL	0
	00058	6098	ac 03 60	ldy xkor			
	00059	609b	c9 9d	cmp #157		csr balra	
0	00060	609d	f0 12	beq bal			0
	00061	609f	c9 1d	cmp #29		csr jobbra	
	00062	60a1	f0 24	beq jobb			
0	00063	60a3	c9 91	cmp #145		csr fel	0
	00064	60a5	f0 33	beq fel			
	00065	60a7	c9 11	cmp #17		csr le	
0	00066	60a9	f0 40	beq le			0
	00067	60ab	c9 20	cmp #32		székész	
	00068	60ad	d0 e6	bne bill		mást nyomtunk	
0	00069	60af	f0 4d	beq vege			0
	00070	60b1		;			
	00071	60b1	ad 03 60	bal	lda xkor	bal szél?	
0	00072	60b4	d0 06	bne bb			0
	00073	60b6	ad 05 60	lda xmax		akkor bejön	
	00074	60b9	8d 03 60	sta xkor		a jobb szélén	
0	00075	60bc	ca 03 60	bb	dec xkor		0
	00076	60bf	ad 08 60	tovabb	lda puff	eltarolt betű	
	00077	60c2	91 14	sta (zpl),y		wisszaírása	
	00078	60c4	18	clc			
0	00079	60c5	90 b5	bcc maszik		mindig ugrik!	0
	00080	60c7		;			
	00081	60c7	ee 03 60	jobb	inc xkor	egyet jobbra	
0	00082	60ca	ad 03 60	lda xkor			0
	00083	60cd	cd 05 60	cmp xmax		jobb szél?	
	00084	60d0	d0 05	bne jj			
0	00085	60d2	a9 00	lda #0		akkor bejön	0
	00086	60d4	8d 03 60	sta xkor		a bal szélén	
	00087	60d7	18	clc			
0	00088	60d8	90 e5	bcc tovább		wissza. stb.	0
	00089	60da		;			
	00090	60da	ad 04 60	fel	lda ykor	lefelül van?	
0	00091	60dd	d0 06	bne ff			0
	00092	60df	ad 06 60	lda ymax		akkor bejön	
	00093	60e2	8d 04 60	sta ykor		alul	
0	00094	60e5	ca 04 60	ff	dec ykor		0
	00095	60e8	18	clc			
	00096	60e9	90 d4	bcc tovább		wissza. stb.	0
0	00097	60eb		;			0




```

○ 00098 60eb ea 04 60 le inc ykor ;egyet le ○
00099 60ea ad 04 60 lda ykor ○
00100 60f1 cd 06 60 cmp ymax ;legalul van? ○
00101 60f4 d0 c9 bne tovabb ○
00102 60f6 a9 00 lda #0 ;akkor bejan ○
00103 60f8 8d 04 60 sta ykor ; felul ○
00104 60fb 18 clc ○
00105 60fc 90 c1 bcc tovabb ; vissza. stb. ○
00106 60fe ; ○
00107 60fe ad 08 60 vege lda puff ; kilépéskor is ○
00108 6101 91 14 sta (zpl),y ; visszahelyez ○
00109 6103 60 rts ○

○ end of assembly, error count = 00000 ○

○ bal 60b1 bazis 605a bb 60bc bill 6095 ○
c64 6058 fel 60da ff 60e5 init 603c ○
jj 60d7 jobb 60c7 le 60eb maszik 607c ○
Plus4 6054 puff 6008 scrbas 6007 scrhi 6023 ○
scrlo 600a szamel 6076 szamol 6062 temp 6009 ○
tovabb 60bf uc20 6050 vege 60fe xkor 6003 ○
xmax 6005 ykor 6004 ymax 6006 zph 6015 ○
zpl 6014 ○

```



Optikai szövegkiemelés

A mellékelt Flashtext program a számítógépes szövegek színes aláfestésére szolgál. Ez egy DATA sorokba rejtett gépi kódú, illetve egy demo célokat szolgáló BASIC programból áll. Az Assembler rész a kazettapuffert foglalja le a 828—952 között. Kazettára így nem tudjuk kimenteni.

A programot vigyük be, majd mentsük ki egy lemezre. A betöltés a LOAD "FLASHTEXT",8 utasítással történhet, az indításhoz a RUN parancs kell. Ekkor történik a fő rész generálása.

A normál írásmódban (a Flashtext nélkül) minden karakter azonos háttérszínt kap. A segítséggel négy különböző színt választhatunk, s ezek közül három villogót. Erre a célra az úgynevezett kibővített színmódus (Extended Colour Mode) szolgál. A villogást pedig az interrupt segítségével valósítjuk meg.

Az első villogó színt a szöveg (SHIFT)-tel történő bevitelekör (jó a (SHIFT LOCK) is) érhetjük el. A másodikat az inverz mód bekapcsolásával (CTRL 9). A harmadikat a (SHIFT) és az inverz egyidejű használatával, a negyediket pedig a normál írással. A színeket természetesen mi választhatjuk. Az első módé a 905—913-as címeken áll, a másodiké a 914—922 között, a harmadiké a 923—931-nél. SYS 932-vel a rutin lekapcsolható. Az újraindítás a SYS 828-cal lehetséges.

```

○ 1 data 173,17,208,9,64,141,17,208,169,0,133,254,133,253,120,169,87 ○
2 data 162,3,141,20,3,142,21,3,88,96,165,253,201,2,208,39,164 ○
3 data 254,185,137,3,141,34,208,185,146,3,141,35,208,185,155,3,141 ○
4 data 36,208,230,254,165,254,201,9,208,4,169,0,133,254,169,0,133 ○
5 data 253,76,49,234,230,253,76,49,234,9,0,7,1,7,8,9,0 ○
6 data 0,0,2,10,7,1,7,10,2,0,0,0,11,12,15,1,15 ○
7 data 12,11,173,17,208,41,191,141,17,208,120,169,234,162,49,141,21 ○
8 data 3,142,20,3,88,96 ○
9 for i=828 to 952:read a:Poke i,a:next:Print "952!":Poke 53280,0:Poke 53281,0:sys ○
828 ○
10 Print "A 'FLASH TEXT' 4 kulonbozo" ○
11 Print "21. irasmodot tud:" ○
12 Print "22. normal szoveg" ○
13 Print "23. SZOVEG BEIRAS SHIFT -TEL" ○
14 Print "24. Iras inverz karakterekkel" ○
15 Print "25. INVERZ IRAS SHIFT -TEL" ○
16 Print "26. sys 9328 leallija a villogast, sys 828" ○
17 Print "27. ujrainditja a 'FLASH TEXT'-et":Poke 198,0:wait 198,1:sys 932:end ○

○ ready. ○

```


Áttérés COMMODORE-ról IBM gépre

VÁLTÓ

Mottó:
Minden gépnél van jobb

Programozási nyelvek

Az IBM gépek talán legelőnyösebb oldala a rendelkezésre álló programozási nyelvek sokasága. Alapesetben használhatjuk a BASIC nyelv egy viszonylag egyszerű változatát, amelyet a rendszerrel együtt szoktak adni. Eredeti IBM gépeken ez a BASICA névre hallgat, de ez elég ritka, elterjedtebb a GWBASIC nevű program. Mindkét említett program interpreter jellegű, azaz úgy működik, ahogy a COMMODORE gépek BASIC-je: beírjuk a programot, majd RUN paranccsal futtatjuk. A fejlettebb nyelvek általában COMPILER jellegűek, azaz egy (tetszőleges, vagy a nyelv sajátja) szerkesztővel

bevisszük a programot, majd ebből egy fordító-szerkesztő rendszer (általában, EXE, ritkábban COM kiterjesztésű) futtatható állományt állít elő. Az elterjedtebb nyelvek IBM gépeken:

BASIC	az alap...
PASCAL	tanuláshoz ideális
ASSEMBLER	gépi kódú programozáshoz, haladóknak
DBASE	adatbázis-kezelés
C	gépközei, hatékony

Természetesen ez nem a teljes választék, és ezen belül is számtalan program lehetséges. Jellemző példa, hogy egy PC-vel egy éve foglalkozó ismerősöm kérdé-

semre, mely szerint hányféle BASIC-et ismer, a következőt válaszolta:

— Ha a különböző verziószámokat nem nézem, akkor hármat, ezek a BWBASIC, TURBOBASIC, QUICKBASIC.

Az IBM gépeknél az előző mondatban a 'különböző verziószám' kitételnek nagyon fontos szerepe van. Itt ugyanis a nyugati üzletpolitikának köszönhetően, egy programrendszer soha nem készül el igazán, maximum leállnak a fejlesztésével. Így előfordul, hogy az, aki egy programnyelv 2.1-es verziójával foglalkozik 2 éve, nem tud olyan jó, gyors, egyszerű programot írni, ami annak, aki a 3.3-as verziót ismeri egy hónapja — ugyanabból a programból —, gyerekjáték, mert annyit fejlődött a nyelv. Előfordul, hogy mire elkészül egy programtermék új verziója, már nem is kezdik el árusítani, mert a lényegesen jobb is kész lesz egy hónap múlva, és többet vesztené a cég a cserével, mint amennyit az egy hónap várakozással veszít. A csere nem elírás, hiszen a dologhoz az is hozzátartozik, amit 'UPGRADE'-nek neveznek, vagyis a régi verzió tulajdonosai kedvezményes — olykor jelképes — összegért jutnak az újhoz.

Magyarországon a kezdő IBM felhasználók birtokában általában a BASIC programok közül a GWBASIC van meg, így ezt ismertetem kissé részletesebben. Ez a program a közismert Microsoft cég terméke, első verziója 1983-ban jelent meg. Az általam ismert utolsó a 3.22 verziószámot viselte.

Az Országos Commodore Egyesület szolgáltatásai:

C64-be átkapcsolható új operációs rendszer (Speed) + reset beépítése:	2000 Ft	8—16 Kbyte-os epromkártya (cartridge, eprom nélkül)	600 Ft
1541 kompatibilis lemezegységbe Speeddos beépítése (átkapcsolhatóan) 40 TRACK (+ 85 blokk/lemezoldal), valamint párhuzamos 15 pólusú Canon csatlakozó beépítése:	2000 Ft	C64-hez tároló oszcilloszkóp (párhuzamos kábel nélkül)	7500 Ft
C64 USER-port 1541-es lemezegység összekötő párhuzamos kábel:	1300 Ft	A háttértárakhoz epromok programozása (kész programok, vagy saját, hozott programok beégetésével) egységesen:	500 Ft
1541 kompatibilis lemezegységbe elektronikus lemezlyukasztó beépítése:	800 Ft	C64 bővítő-port elosztó (egyszerre 4 db cartridge lehet a gépben, melyeket gombnyomásra lehet kapcsolni)	7500 Ft
PAGEFOX magyar ékezetes szövegszerkesztővel rendelkező cartridge:		C64 USER — CENTRONICS nyomtatókábel (GEOS kábel)	1500 Ft
(Epson típusú nyomtató min. 640 képpontos szükséges a nyomtatáshoz)	6000 Ft	256K RAM-diszk	13000 Ft
FASTLOAD (lemezes gyorstöltő, másoló, monitor)	1400 Ft	64/256K RAM-diszk	9000 Ft
TTL IC-TESZTER cartridge + program	4300 Ft	256K RAM-diszk (RAM-ok nélkül)	7500 Ft
288/256 Kbyte-os eprombank (vezérlő eprommal)	5340 Ft	A fenti bővítések megrendelhetők levélben, vagy az OCE irodájában személyesen. Ha személyesen kívánja megrendelni, kérjük, előtte telefonáljon.	
Epromégető (2716-tól 27256-ig)	4300 Ft	Árainkat az alkatrészárak változásai befolyásolhatják.	

A program indítása: GWBASIC(enter) (előfordul BASIC és BASICA neven is).

A program bejelentkezése után a COMMODORE gépeknél megszokott módon vihetünk be BASIC programsorokat. Az utasítások hasonlóak az ott megismertekre, leginkább talán a BASIC 3.5-re és 7.0-ra (C16-os ill. C128-as). Az alsó sorban egy tájékoztató panel közli, melyik funkciógombra mit programoztak. Ezek a funkciók a KEY SORSZAM, „szöveg” utasítással megváltoztathatók. Nem soroljuk fel az összes lehetséges utasítást, parancsot, belső változót, függvényt (ezekből összesen mintegy 150 van), de ami nem nyilvánvaló, és első próbálkozáskor jó tudni:

— Az aktuális katalógusban szereplő bejegyzéseket a FILES paranccsal nézhetjük meg. A parancsban a keresési utat idézőjelek között megadhatjuk, de akkor a végén is tegyük ki a \-t! A záró idézőjel elhagyható.

— A programból közvetlenül elérhető az MKDIR, RMDIR, CHDIR parancs, alkatalógusok létrehozására, törlésére, illetve az aktuális katalógus megváltoztatására, de ezek rövid formája (MD, RD, CD) NEM használható!

— A FILES és a CHDIR nem mindig ugyanarra vonatkozik! Előfordulhat, hogy a CHDIR-rel beállítottuk a keresési utat, ahova-ahonnan a SALE-LOAD

történik, de a FILES nem innen ad listát. Ez általában akkor szokott megtörténni, ha a CHDIR paranccsal meghajtott is váltunk.

— A mentés szintaktikája a következő:

SAVE "nev",a

A név után a (vessző-a) elhagyható. Szerepe az, hogy ezt megadva a mentés 'normál' ASCII alakban történik, így tetszőleges szövegszerkesztővel módosíthatjuk majd a programot, azaz nem kell majd újra a BASIC-et behívni, illetve ha később ki akarjuk nyomtatni programunkat, akkor is ezt a kapcsolót kell használnunk. Visszatöltés a LOAD "név" formában történik, a formátum felismerése (ASCII vagy nem) automatikus. A kimentett programok kiterjesztése .BAS lesz, ezt NEM kell megadni, sem mentésnél, sem töltésnél, szintén automatikusan kezeli a gép. Ezt a lehetőséget az is jól kihasználhatja, aki hozzászokott már valamelyik szövegszerkesztőhöz, amivel a legtöbb ilyen SCII formátumú file-t is lehet készíteni. Ekkor a program bevetele történhet a szövegszerkesztővel is.

— A GWBASIC-et elhagyni a SYSTEM paranccsal lehet. A visszatérés a rendszerbe mindig oda történik, amit a FILES mutat!

— Az aktuális időt és aktuális dátumot a TIME\$ és DATE\$ tartalmazza,

ezek megváltoztatása a rendszer DATE és TIME parancsaival egyenértékű, tehát a rendszeridőt állítja!

— Ha kész programállományunk van, akkor a program azonnal indítható DOS-ból a GWBASIC PROGRAMNÉV parancs kiadásával. Ilyenkor a GWBASIC betöltése után betölti a programot is, majd azt el is indítja egy képernyőtörléssel kezdve. Ha a program lefutott, a GWBASIC-ben maradunk, kilépni a SYSTEM-mel lehet. Ha a program utolsó utasítása a SYSTEM, a kilépést is automatikusan végrehajtja.

— Ha van printer, közvetlenül írhatunk rá a kiíró parancsok L betűs változatával: LLIST, LPRINT.

— Zenét egyszerűen a PLAY utasítással csálhatunk ki a gépből. Az utasítás a BASIC 7.0-ban előforduló PLAY-hez kísértetiesen hasonló.

— Néhány utasítás az [ALT]+kezdőbetűje hatására előáll.

— A PRINT utasítás ugyanúgy kérdőjellel rövidíthető, mint a Commodore gépeknél.

— A GWBASIC igen kényes a szintaktikára, ezért a megfelelő helyeken a [SPACE] használhata kötelező!

— A GWBASIC programozására 60K terület, azaz pontosan 60300 Byte áll rendelkezésünkre.

Lengyel István

Képernyő swapping

A mellékelt kis utility profiknak és kezdőknek egyaránt érdekes. Alkalmazásával mód nyílik szövegek, sőt grafikák ki- és bekapcsolására. Így megvan az a lehetőségünk, hogy képernyőket a „láthatatlan” tartományban manipuláljunk, majd azt egyszerre elővarázsoljuk. A POKE 176, érték (vagy az LDA #ér-

ték:STA \$B0) rögzíti a képernyő magasságát, azaz hogy meddig toljuk lefelé a tartalmat. Például a 14 (\$0E) a képernyőt teljesen be-, a 214 (\$D6) teljesen kikapcsolja. Hurkok és táblázatok ügyes szervezésével érdekes hatásokat lehet elérni a képernyő lágy görgetésétől a szinuszforma vibrálásig. A mellékelt BASIC betöltő a programot a 49152—49226 területre viszi, ahol az a SYS 49152-vel indítható. Itt a 176-os tárolócímmel lehet kísérletezni.

```

10 REM *****
20 REM *      INDITAS: SYS 49152 *
30 REM *      MAGASSAGALLITAS A *
40 REM *      POKE 176, MAGASSAG *
50 REM *      ERTEKADASSAL. *
60 REM * MAGASSAG= 14 EGESZEN FENT *
70 REM * MAGASSAG=214 EGESZEN LENT *
80 REM *****
30000 FOR I=49152 TO 49226:READ J:POKE I,J:NEXT
32000 DATA 120,169,0,141,14,220,169,241,141,26,208,169,0,141,25,208,169,27,141
32001 DATA 17,208,169,30,141,18,208,169,0,141,255,63,162,47,160,192,142,20,3,140
32002 DATA 21,3,169,14,133,176,88,96,166,176,173,18,208,205,18,208,240,251,41
32003 DATA 7,9,24,141,17,208,202,208,238,169,1,141,25,208,76,49,234

READY.
    
```


Adataink élettere

Bizonyára ismeretes a szituáció: Lét-rehaztunk egy csomó adatot, amelyek a tárolóban vannak. De hova tegyük azokat? Azt fogjuk most megmutatni, hogyan lehet ezt a problémát optimálisan megoldani az állományok, idegen szóval file-ok (ejtsd: fájlok) segítségével.

Tulajdonképpen az adatállományok nagy szekrényeknek tekinthetők, amelyben az adatokat be tudjuk rakni. Na de mi is az az állomány pontosan? Nos semmi más, mint egy kupac adat gyűjteménye, amelyek önmagukban egységet képezve egy kartotékrendszerben vannak elhelyezve. Egyszerűen kifejezve az állományok esetében olyan adatokról van szó, amelyekre egy meghatározott gyűjtőfogalom alatt — ez az állomány neve — hivatkozhatunk. Állományról tulajdonképpen akkor beszélhetünk a szó igazi értelmében, ha az adatainkat, a programot stb. egy külső készülék segítségével adathordozóra rögzítjük. Hogy mit rögzítettünk, az adja az állomány fajtáját. Pl. program esetében programállományról beszélünk. Láthatjuk tehát, hogy az állományokat majdnem mindig külső készülékek segítségével (amelyeket perifériáknak is nevezünk) tároltunk. Ez lehet a Datasette, vagy a mágneslemez meghajtó is. Az állományok tehát egyfajta emlékezet raktárak a számítógép számára, amelyek annak kapcsolása után is a rendelkezésünkre állnak.

Sajnos a programállományokkal nem lehetséges valamely számításokhoz, vagy statisztikai kiértékelésekhez segítséget nyújtani adatok formájában. Ahhoz ugyanis másféle speciális állományokra van szükség, amelyek a mi problémáink megoldásához szükségesek és érvényesek. Vegyünk példának egy címjegyzéket. Minden komplett cím egy információs egységet jelent. Ezeket a címeket ugyanúgy le lehet rakni egy file-ban is. Egy állomány mindig egyes rekordokból (tételekből) áll. Minden egyes rekord egy információs egységet hordoz, ebben az esetben egy teljes címet.

Állományok rögzítéséhez természetesen szükség van egy megfelelő hordozóeszközre. A C64-es tulajdonosok kétféle lehetőség közül választhatnak. Vagy a lassú Datasette-tel dolgoznak, vagy a nemcsak a sebességet tekintve gyorsabb le-

mezmeghajtóval. Most nézzünk meg előbb egy olyan adatállomány mintát, amelyet mind a Datasette-tel, mind a 1541-es lemezmeghajtó egységgel kezelni lehet.

Kazetta vagy lemez?

A szekvenciális állományról akarunk tehát szólni. Ennél a fajtánál az adatokat szép libasorban visszük ki mindaddig, amíg a gép egy RETURN jelet nem küld. Ezt a RETURN jelet a rekord elkészítésekor annak végére rakja a számítógép. A RETURN nemcsak egy jel a számítógépen, hanem tulajdonképpen egy jel is. Ennek a belső ASCII kódja 13, és a felhasználó által történő minden sztring bevitelnél amögé lesz helyezve. A RETURN tehát egy speciális végállapot jelző a számítógép számára. Így tudja a gép később is megtalálni egy rekord végét. A soros állományú, tehát a szekvenciális halmazoknál teljesen mindegy, hogy egy rekord milyen hosszú, annak átvitele elviekben mindig a RETURN jelig történik. A következő rekordot azután egyszerűen ez után a jel után küldjük ki. Ahhoz, hogy egy ilyen állományba valamit beírassunk, az alábbiak szerint kell eljárni:

```
10 OPEN 1,8,2,"TESZT,S,W"
20 PRINT#1,"EZ EGY REKORD"
30 CLOSE
```

A 10. sorban közöljük a számítógéppel, hogy a következőkben szeretnénk a "TESZT" nevű állománnyal dolgozni. Erre egyes egyedül az OPEN parancs szolgál: Hozzárendelni egy állományhoz egy számot, amely alatt arra mindig hivatkozni lehet. Az "S" a név mögött az első vessző után azt jelzi, hogy az állomány szekvenciális, azaz soros. A "W" jelenti azt hogy "WRITE" azaz írás. Ha olvasni akarunk, akkor a "W" helyett "R"-et kell megadni ("READ"), illetve ha újabb adatokat akarunk hozzáfűzni az állományhoz, akkor "A"-t ("APPEND"). Az OPEN parancs után egy állomány mindaddig a rendelkezésünkre áll, amíg azt a CLOSE paranccsal le nem zárjuk, ahogy a 30. sor mutatja.

Amint azt a továbbiakban látni fogjuk, ez nemcsak a szekvenciális állományok esetében van így. De előbb hadd ma-

gyarazzuk el az OPEN parancs mögött álló, talán kissé misztikusnak tűnő számokat.

Az egyes a mi logikai adatállomány számunk. Ez alatt a szám alatt lehet a CLOSE parancs kiadásáig a TESZT nevű állományra hivatkozni, azt feldolgozni. A második szám az úgynevezett készülékszám. Ebben az esetben a 8-as számú mágneslemez meghajtóra hivatkoztunk. Végül a harmadik szám a másodlagos címet testesíteni meg, amely jelentőségét még a későbbiekben meg fogjuk magyarázni. Azt, hogy hogyan kell egy állományra hivatkozni, a 20. sorban láthatjuk. A már ismert PRINT parancs ezúttal kissé másképpen néz ki. Itt most nem a képernyőre vonatkozik, hanem a 10. sorban megnyitott "TESZT" nevű állományra. A számítógép ezt az utasítás mögé helyezett "#" (duplakereszt) jellel ismeri föl. A jel mögött áll a már ismert logikai adatállomány szám, amelyet a megnyitott állományhoz rendeltünk az OPEN paranccsal. A PRINT#1 utasítást követő sztringet a gép egy rekordként beírja a "TESZT" nevű állományba. Így mi máris végrehajtottunk egy komplett adathalmaz műveletet. Foglalkozzunk most a szekvenciális adathalmazok értelmes alkalmazási lehetőségeivel. Ehhez előbb meg kell magyaráznunk azt is, hogyan hivatkozik a floppy meghajtó egy rekordra.

Ha egy állományt olvasásra megnyitunk, akkor az egység az író/olvasó fejlet a file első rekordjára állítja. Minden egyes olvasási művelettel egyet továbblépünk. A tároláskor egy rekord végére a RETURN jelet írtuk, így a lemezmeghajtó könnyen felismeri azok végét. Ezzel a módszerrel szépen sorban egészen az állomány végéig tudunk olvasni. Ahhoz, hogy egy szekvenciális file-t összességében feldolgozzunk, és az egyes rekordokra tetszés szerint hivatkozhassunk tetszés szerint hivatkozhassunk, csak úgy van lehetőség, ha az állományt teljes egészében beolvassuk a számítógépben egy tömbbe. A feldolgozás után az állományunkat azután ismét vissza kell írni az adathordozóra. Egy dolgot azonban mindig szem előtt kell tartanunk, ha megnyitunk egy már létező szekvenciális állományt. Ha azt másodszor írásra nyitjuk meg, akkor azzal a már meglévő rekordokat fölülírjuk. Nin-

csen ugyanis lehetőségünk arra, hogy az állományba beszúrjunk újabb rekordokat. Csupán arra van mód, hogy az állomány mögé, az utolsó rekord után helyezzük el az újabb adatokat. A már meglévő adatokat mondjuk az alábbi módon lehet beolvasni:

```
10 DIM A$(100)
20 OPEN 1,8,2,"TESZT,S,R"
30 FOR I=1 TO 100
40 INPUT#1,A$(I)
50 IF ST=64 THEN I=100:CLOSE 1
60 NEXT I
70 ...
```

A 10. sorban dimenzionálunk egy sztring tömböt, amelybe azután beolvasuk az adatokat. Ezt követően a 20. sorban megnyitjuk a "TESZT" nevű állományt olvasásra. A FOR...NEXT hurokban olvassuk be a rekordokat az INPUT#1 parancs segítségével, amely a PRINT#1 parancssal ellentétben adatokat olvas ki bármely állományból. Természetesen előbb vagy utóbb mindig eljutunk az adathalmaz végére. Ezt okvetlenül le kell kérdezni a programban, mivel a számítógép önállóan nem reagál arra. A C64-es ehhez egy előre definiált változót használ, az ST (STátusz). Mihelyt ez a 64 értéket veszi föl, tudhatjuk, hogy elértük a szekvenciális állományunk végét, tehát el kell hagynunk a beolvasási FOR...NEXT hurkot. Abból persze nem ugrunk ki (pl. egy GOTO 70-nel), hanem a szabályos módon (I-nek a végső értéket adva) kilépünk. Hurkokat csakis ilyen módon szabad „idő előtt” elhagyni! Ha netán szükségünk van a hurokváltozó (itt az I) tényleges értékére, a programunkat így módosítuk:

```
30 FOR I=1 TO 100
40 INPUT#1,A$(I):A=I
50 IF ST=64 THEN I=100:CLOSE 1
60 NEXT I
70 I=A:...
```

Mindezek a rekordok most az A\$(I) tömbben vannak, és azokat tetszésünk szerint fel tudjuk dolgozni. Ha most a táblázat ismét rögzíteni szeretnénk a feldolgozott új állapotában, úgy ismét meg kell nyitni a beolvasás vége után lezárt állományt, de ezúttal írásra. Ha viszont csupán hozzá akarunk fűzni valamit a már meglévő adatokhoz, akkor ilyenkor a megnyitásnál az "R" illetve a "W" helyett az "A" betűt kell használnunk.

Ezekkel az ismeretekkel felszerelve, már bánni is tudunk a szekvenciális állományokkal. Talán ezek hátrányát is föl tudjuk már mérni: Mivel a hatékony feldolgozáshoz állandóan tömböket kell definiálnunk, nagyobb állományok használatakor igen gyorsan beleütközünk a gép tárolóterületének korlátaiba.

Mennyivel jobb lenne, ha lenne egy olyan állományfajta, amelyiknél tetszőlegesen ugrálhatunk ide-oda. Egy olyan

adathalmaz, amelyikből bármely rekordot ki tudjuk olvasni ha éppen kell. Nos a 1541-es floppy rendelkezik ezzel a lehetőséggel. A kézikönyvben azonban erről meglehetősen kevés információt lehet olvasni. Az említett állományfajta neve relatív adathalmaz. Egy ilyen struktúra szervezésénél azonban a felhasználónak több dologra kell ügyelnie mint a soros állományoknál.

Gyors hozzáférés

A relatív állományokat minden esetben egy előre definiált rekordhosszúsággal kell létrehozni. Mindjárt ezzel kapcsolatban alaposan át kell gondolnunk a dolgot, a halmaz megszervezése előtt. Tudnunk kell, hogy milyen információkat fogunk egy rekordba fölvenni, hiszen ettől függ végül is hogy mekkora lesz a rekordhossz. Itt viszont megvan arra is a lehetőségünk, hogy a rekordot különböző részmezőkre osszuk föl. Hogy ez mire is jó? Gondoljuk csak át a dolgot. Mondjuk szeretnénk egy címjegyzéket szervezni. Ebben benne kell akkor lennie egy személy összes adatának. Szükségünk van a névre, a pontos címre, a telefonszámra. Hivatalos esetekben a bankszámlaszám, telex stb. feljegyzése sem lehet káros. De térjünk vissza a relatív állományhoz. Ahhoz, hogy egy rekordra hivatkozhassunk a halmazon belül, kell hogy legyen annak egy sorszáma. Nézzük meg előbb a relatív állomány megnyitásához szükséges parancsot:

```
OPEN 1,8,2,"TESZT2,L"+CHR$(50)
```

Csak semmi pánik! A dolog fele olyan bonyolult sincs, mint amennyinek látszik. Azt már tudjuk, hogy az OPEN parancssal kell egy állományt megnyitni, illetve hogy most a mi esetünkben ismét az egyes logikai számot adtuk a "TESZT2" halmaznak. Egyébként ez a szám 1 és 127 között bármi lehet. A név mögött a vessző után álló "L" közli a meghajtóval, hogy a rekordhosszúságot kívánjuk még a tudomására hozni. Ezt azután sztringként kell átvinni. Erre szolgál a CHR\$ függvény. Közelebbit erről a C64-es kézikönyvében lehet találni. A mi példánkban a rekordhossz pontosan 50 karakter lesz. Ezzel a "TESZT2" halmazt máris mint relatív állományt deklaráltuk és nyitottuk meg. Most még egy olyan utasításra van szükség, amellyel lehetséges az is, hogy tetszőleges helyre lépünk a rekordon belül. Itt viszont már komplikálódik a dolog. Először is meg kell nyitnunk a floppy parancscsatornáját. Ennek a csatornának a száma, ahogy azt a kézikönyvből is kiolvas-

hatjuk a 15. A kiegészítő OPEN parancs tehát ez lesz:

```
OPEN 2,8,15
```

Ezt követően természetesen közölnünk kell a meghajtóval azt is, melyik rekordot akarjuk földolgozni. Ezt a következő sorral lehet elérni:

```
PRINT#2,"P"+CHR$(2)+CHR$(LB)+CHR$(HB)+CHR$(1)
```

Nézzük csak meg közelebbről ezt az elsőre minden bizonnyal zavarosnak tűnő parancssort. A logikai állományszám (2) egyértelműen hivatkozik az előbb megnyitott parancscsatornára. Ezt követően sztringként átadjuk a relatív állományunk OPEN parancsában álló másodlagos címet (amely itt szintén 2). A másodlagos cím (csatornaszámnak is mondják) mindig a harmadik szám az OPEN parancs paraméterei közül. Ennek értéke az állományokat illetően 2 és 14 közé kell hogy essen. Az ezt követő két sztring jelenti a feldolgozni kívánt rekord sorszámat. Talán fölmerül a kérdés, miért van szükség egy szám közlésénél két sztringre? Nos azért, mivel egy CHR\$ utasítással csupán 0 és 255 közötti értékeket tudunk küldeni. Egy relatív halmaz azonban minden további nélkül akár 1000 rekordot is tartalmazhat. Ezért a rekordszámot át kell alakítanunk 256-os számrendszerre, amivel azután két sztring képében az összes lehetséges rekordszámot át tudjuk küldeni. A kívánt értékeket a következő képlettel kaphatjuk meg:

$$HB = \text{INT}(\text{rekordszám} / 256)$$

$$LB = \text{rekordszám} - HB * 256$$

Az így kapott két értéket már problémamentesen át tudjuk vinni. A legutolsó CHR\$ értékkel tudunk a rekordon belül egy tetszőleges helyre lépni. Ha most visszagondolunk a címjegyzékünkre, akkor ez azt jelenti, hogy ki tudjuk olvasni ha kell csak a nevet, csak a címet stb. Ez természetesen csak akkor lehet, ha az egyes adatokat külön külön vittük ki a rekordba. (PRINT#1,név,cím,telefonszám,...). Ugyanis ebben az esetben is el látja a számítógép az információkat a RETURN jellel (rekordhossznál figyelembe venni!). Az INPUT# parancs azután eddig a jelleg olvassa be az adott információt, akárhová is pozícionáltuk a mutatót. Ahogy azt már említettük, a relatív adatkezelés esetében tetszőlegesen lépkedhetünk a rekordok között is.

Talán egy dolgot meg lehet itt említeni. Éppen emiatt, mivel ez az előny megvan, relatív szervezésű állományokat a Datasette segítségével kazettára nem tudunk menteni. Ott ugyanis csak arra van lehetőség, hogy a magnófejet lépésről lépésre vigyük előre, vissza nem nyúlhatunk.

Nos miután már megismerkedtünk a szekvenciális és a relatív állományokkal, rátérhetünk az adatkezelésnek egyik leggyakoribb és igen komfortos módjára. Ehhez először be kell vezetnünk az indexszekvenciális állomány fogalmát. Ez az állományfajta keverékforma, amely a két, már említett struktúrából áll, azaz a szervezéshez szükségünk van úgy relatív, mint szekvenciális állományra.

Profi módszer

Talán már dereng is valami az egész dolog lényegéről. A relatív állományok bár szépek, gyorsak és jók, de ahhoz hogy hivatkozni tudjunk rájuk, ismerni kell az egyes rekordok sorszámait. Ha mondjuk a címjegyzékben a Kovács úr adataira van szükségünk, tudnunk kell azt, melyik rekordba szerepelnek a keresett adatok. Mennyivel jobb volna, ha csak annyit kellene beadnunk hogy „Kovács”, és máris megkapnánk a kívánt információkat a képernyőn! Mondjuk a Kovács lenne a kulcs a megfelelő rekordhoz. A névmezőből kiindulva tehát hivatkozhatnánk az állományunk rekordjaira. Vajon meg lehet oldani ezt a problémát a C64-gyel és a 1541-es floppyval? Nos ehhez kell az, hogy kombináljuk a kétféle állományfajta. Szükség van egy szekvenciális halmazra, amelyben az alábbi adatokat tároljuk: A használt kulcsot, azaz a mi esetünkben a nevet, illetve az ahhoz tartozó rekordszámot a relatív állományban. A szekvenciálisan szervezett adatokat egy, a számítógép tárolójában lévő tömbben tudjuk tárolni. Ha egy meghatározott névre van szükségünk, akkor kikeressük azt a tömbből, fogjuk az ott megtalált rekordszámot is, és máris kiolvashatjuk a relatív állományunkból az ehhez tartozó adatokat. Csupán arra kell mindig figyelni, hogyha valamely rekordot törölünk, akkor a hozzátartozó kulcsot is el kell távolítani. Hasonlóképpen ha bővítjük az adathalmazunkat, akkor a kulcsállományt is ki kell egészíteni. A táblázatunkat azután a program indításakor be kell olvasatni, illetve a kilépéskor le kell tárolni.

A teljesség kedvéért meg kell még említenünk két állománytípust is, amelyek felhasználására a profi, Assembler nyelven programozók vállalkozhatnak. Egyrésztől ott vannak a USER állományok, amelyek nagy vonalakban hasonlóan vannak fölépítve, mint a szekvenciális halmazok. Ezeket a file-okat főleg az ún. „spooling” esetében használják. A spooling azt jelenti, hogy a lemezen található állományt közvetlenül a nyomtatóra küldjük ki adott esetben anélkül, hogy

igénybevennénk a számítógép idejét és segítségét, vagy tárolóterületet onnan. Ilyen megoldásokat azonban csakis gépi kódú programozással lehet megcsinálni. A kezdőnek tehát már meglévő programokat kell igénybe vennie ehhez. Aki a profi lemezkezeléssel akar foglalkozni, annak a szakirodalomhoz kell fordulnia.

Mindenekelőtt a GEOS-szal dolgozó felhasználók számára érdekes a másik megemlíthető állományfajta. Ezt az állományt VLIR-nek nevezik. Ez a Variable Length Indexed Record kifejezésből „gyártott” mozaikszó, ami magyarul annyit jelent, hogy változó hosszúságú indexelt rekordok. Ez az állományfajta

megszabadítja a relatív szervezést attól a mankójától, hogy kötve vagyunk a rögzített rekordhosszhoz. Ezek az állományok azzal a rendkívüli előnnyel rendelkeznek, hogy nemcsak változtatható a rekordhossz, de a szervezés olyan mint amelyet a relatív halmazok esetében már megszoktunk.

Most hogy már megtudtunk néhány dolgot az állományok elméletéről, szervezéséről, talán az lenne a legjobb, ha megpróbálkoznánk egy saját adatbázis kezelő program írásával. Ha valami nem stimelne, gondoljunk arra, egyszer mindenki volt kezdő. A nehézségek pedig mindenkinél ugyanazok, azokon pedig előbb vagy utóbb, de átlépünk.



Gépkereskedelmi és Ügyviteltechnikai Kft.

A Gépker Kft. Canon szerviz az alábbi kedvező árakon értékesíti Canon fénymásoló- és faxkészülékeit:

Canon FC2	49.900 forint
Canon NP 1010	119.000 forint
Canon NP 1520	204.000 forint
Canon NP 2020	269.000 forint
Canon NP 3825	359.000 forint
Canon NP 6650	899.000 forint
Canon NP 8530	1.690.000 forint
Canon CLC 300	1.690.000 forint
Canon fax 80	54.000 forint
Canon fax 120	79.000 forint
Canon fax 270S	109.000 forint
ASI NT 1104 pénztárgép*	40.000 forint

*mely teljes összegben visszaigényelhető az APEH-től

Az árak áfa nélkül értendők és tartalmazzák az 1 év garanciát is. Vásárlásai esetén további 5% kedvezményt biztosítunk.

Cím: Bp., XIII., Frangepán u. 7.

Tel.: 120-9420 * 129-9377 * Fax: 120-9420

A merevlemezek világa

II. rész

Amiga 2000-es SCSI rendszerek

A2090A HD/SCSI kontroller

Bár az A2090A kitűnően viselkedik alapkörülmények között, overscan-képernyő használatokor egyszerűen érthetetlen az a lassulás, amely tapasztalható.

Egy overscan kép bekapcsolásakor nem okoz kárt a lemezen mint elődje az A2090-es tette, azonban különböző időmérési és szinkronizációs problémák merülnek fel nála. Ennek eredményeképpen az A2090A nem elfogadható a legtöbb videó és grafikai alkalmazás esetén.

A drive-ok A2090A-val történő üzembe helyezése egyszerű, nem igényel különleges ismereteket.

Az A2090A speciális jegyei közé tartozik az a képessége, amellyel ellenőrzi az ST506 alapú drive-okat, melyek kevésbé költségesek, azonban jóval lassabbak.

Ez az áramkör a felsorolt kontrollerek között a legköltségesebb, ami pedig nem éppen jó pont.

Még meg kell említeni, hogy már megjelent a piacon az A2091-es ellenőr is, amelynél a gyártó ígérete szerint, megszűntek a fent említett problémák és jelentős sebességnövekedést érhetünk el használatával.

Ezt a berendezést építik be az Amiga 3000-es gépekbe, de mivel még igencsak ritkán tapasztalhatjuk őket (ennek elsődleges oka a készülék magas ára), nem érdemes foglalkozni vele.

Gyártó:	Commodore Business Machines
Ár:	300 DM (csak a kontroller)
Átvitel I/O:	DMA
Megjelenés:	3
Egyszerűség:	4
Különlegesség:	4
Ár:	2
Átlag:	3,25

HardFrame/A2000 kontroller

A HardFrame a legjobban kialakított és legtökéletesebben működő egység a is-

mertettek között. Bár nem sikerült vele elérni azt az adatátviteli sebességet, amelyet a leggyorsabb ellenőrök biztosítani tudnak, mégis ennek a kontrollernek a működési biztonsága veretlen. Megmutatja mindazt, amit egy jól megtervezett DMA interface tenni képes.

A HardFrame egyéb jellemvonásai is kiemelkedőek. A felhasználó nem csak az autobooting nyújtotta előnyöket élvezheti, hanem egyszerűen handblock technikával állíthatja be a rendszer működésének kívánt feltételeit, ezenfelül ez az ellenőr a saját kártyáján biztosít helyet a 3,5"-os drive-nak (handcard), így szabadon marad az egyébként elfoglalt normál floppy-meghajtó helye.

A drive-ok bevezetése és inicializálása kicsit körülményesebb, mint egyes társaié. Ellentétben a többi vizsgált kontrollerrel, a HardFrame installációs programja az ellenőrön keresztül információkat szerez a drive-ról, így megkímél minket sok procedúrától. Sajnos azonban ez a software egy csomó felesleges kérdést is feltesz az átvitel sebességével, a felosztással és a file systemmel kapcsolatban, amelyeket látva egy kezdő felhasználó csak „pislogni” tud.

A HardFrame igen jól megfelelt a tesztek során, azonban magas ára visszariasztólhat tünhet.

Gyártó:	MicroBotics Inc.
Ár:	598 DM (csak a kontroller)
Átvitel I/O:	DMA
Megjelenés:	5
Egyszerűség:	3
Különlegesség:	5
Ár:	3
Átlag:	4

FastCard kontroller

A FastCard kontroller megkülönböztető jellegét az adja, hogy processzor I/O-n alapuló rendszere ellenére ő érte el a legnagyobb adatátviteli sebességet. A 256 Kbyte-os Dperf2 teszt elvégzésekor átlagosan 3/4 millió byte-ot olvasott át másodpercenként!

Bár egyes kontrollerek átlagosan felülmúlják, ez az áramkör mégis a legjobb

egység olyankor, mikor nagy adatblokkokat kell gyorsan mozgatni.

A bevezető software-t nagyon egyszerűen akarták tervezni, és ezért egy intuitív bázisra helyezték, ennek ellenére használata mégsem mondható igazán könnyednek. Hasonlóan a HardFrame-hoz a bevezető program itt is automatikusan szerez információkat a drive-ról, azonban ekkor meg rengeteget kell „bőbelődni” az egyes paraméterek beállításával.

Csakúgy, mint az előző kontrollerhez, a FastCard is alkalmaz handblock-ot, csak hogy ez az ellenőr nem engedi meg a már elkészített blokk egyszerű javítását, törlését, ezért a beállításnál nagyon oda kell figyelni.

Összefoglalva a bevezetési software jobbnak ígérkezik, mint bármely más kontrolleré, nem biztosít semmiféle könnyebbséget a használat során.

A hely megtakarítása érdekében a FastCard handcard-ként is használható, és alacsony ára miatt igen remek ajánlat.

Gyártó:	Xetec Inc.
Ár:	269 DM (csak a kontroller)
Átvitel I/O:	68000 I/O
Megjelenés:	5
Egyszerűség:	3
Különlegesség:	4
Ár:	5
Átlag:	4,25

Supra Drive

A Supra Drive gyönyörű példája az AIO (All-In-One), azaz mindent egy helyen rendszereknek.

A Supra ellenőr azokhoz az áramkörökhez tartozik, amelyeket nem lehetett letesztelni az eddig használt Quantum 40S drive-val. Az ok egészen egyszerű: az automatikus bootblockkal rendelkező Supra SCSI használatakor lehetetlenné vált az aktuális floppy meghajtóról történő bootolás, ezért képtelenség volt drive bevetése. Ez csak első hallásra tűnik olyan rossznak, de mint kiderül, nem jelent nagy problémát.

Az általánosan használt Supra SCSI-hez tartozik még egy Seagate ST-138N tí-

pusú drive. Ennek bevezetése csak annyiból áll, hogy a kártyát behelyezzük a gépbe és bekapcsoljuk a rendszert. Ezután a drive automatikusan betölti a bootlockot és máris sínen van minden.

A Supra Drive-hoz hasonló autobootos rendszereket igen könnyű üzembe helyezni.

Bár mindaddig, amíg lehetővé nem teszik a normál floppy-ról történő bootolást, csak egyedülálló rendszerként használható, mégis a Supra Drive egy nagyon jó megoldás azok számára, akik nem akarnak sokat bajlódni a hardware bevezetésével.

A Supra Drive-val kapcsolatban felmerült egyetlen komoly probléma az a torzulás, amelyet a rendszer az overscan képernyő hatása alatt mutat.

Gyártó:	Supra Corporation
Ár:	949 DM (HD + kontroller)
Átvitel I/O:	DMA
Megjelenés:	3
Egyszerűség:	5
Különlegesség:	4
Ár:	4
Átlag:	4

Impact A2000—20HD/RAM

Egy hard-disk kontrollernek legalább olyan egyszerűnek kellene lennie, mint az Impact A2000-nek.

A rendszer bevezetését szinte mindenki el tudja végezni. Az inicializáló program először információkat szerez az ellenőrző egység segítségével a drive-ról, majd pedig feltesz néhány egyszerűbb kérdést a drive felosztásával és méreteivel kapcsolatban. A válaszokat egyszerűen Mbyte-okba kell megadni, és nem kell foglalatni a high és low cylinder értékeivel.

A válaszadást követően a software automatikusan működésbe hozza a drive-ot.

Bár az Impact A2000 nem igazán viszi a primet a megjelenés szempontjából, azért a legtöbb igénynek tökéletesen megfelel.

Gyártó:	Great Valley Products
Ár:	459 DM (csak a kontroller)
Átvitel I/O:	DMA
Megjelenés:	3
Egyszerűség:	5
Különlegesség:	5
Ár:	3
Átlag:	4

Ami az Impact A2000 hardware-t igazán különlegessé teszi, az az, hogy kártyáján külön foglalatokat találhatunk egy maximum 2 Mbyte méretű RAM-bővítés számára. Így, bár az ára kicsit magasnak tűnhet, tulajdonképpen két bővítő-

kártyát kapunk egynek az áráért. Ez a remek megoldás akkor nagyon előnyös, ha egyszerre kívánjuk bővíteni a RAM és a háttértároló rendszerünket, de nem vagyunk éppen milliomosok.

Flash! Card kontroller

Minden eddig ismertetett ellenőrnél volt valami olyan tulajdonsága, amiért érdemes volt ajánlani, kivéve a Flash Card-ot.

Ez tulajdonképpen egy DMA kontroller akarna lenni, de semmi jelét nem mutatja a DMA gyorsaságának. A leírásokban autobootos rendszerként szerepel, de ezt a funkciót külön kell megvásárolni hozzá.

A Flash Card megjelenése messze elmarad az összes többi ellenőr mögött. A bevezető software-t egy körülbelül két éve elavultnak számító program alkotja, amely már megjelenésekor sem számított tökéletesnek.

Végül a nemlétező autobooting-nak köszönhetően képtelenség elindítani a rendszert a jól bevált Quantum 40S drive-val, ezért az ST-138 drive-val készült eredmények szerepelnek a táblázatban.

Összefoglalva úgy is fogalmazhatnánk, hogy nem lehetünk ettől a kontrollerrel igazán elragadtatva.

Gyártó:	Expansion Technologies
Ár:	389 DM (csak a kontroller)
Átvitel I/O:	DMA
Megjelenés:	2
Egyszerűség:	2
Különlegesség:	2
Ár:	3
Átlag:	2,25

TrumpCard kontroller

A TrumpCard önmagába véve nem egy rossz termék, de ennek ellenére mégsem igazán érdekes.

Minden SCSI interface-től elvárt tulajdonsággal rendelkezik, de ezen felül nem nyújt semmi újat.

Az ára még elég reálisnak tűnik, de ha már egy kevés plusszal is fel akarjuk szerelni, akkor már jócskán rá kell fizetni az eredeti árra.

Hagyományos körülmények között elég biztonságosan üzemel, de mihelyt versenyre kell kelnie egy overscan képernyővel, rövid időn belül teljesítménye jelentősen romlik.

A kontroller gyártói megpróbálták kitűnni a rendszer installáció software-

jével, kialakítva a teljes intuition interface-t.

Ez a program igen remekül működik, még egy check-listet is a rendelkezésünkre bocsát, amelyen nyomkövethetjük a drive minden „rezdülését”. A baj csak az, hogy ez a program elvárja, hogy kitűnően kezeljük a cylinder számokat és egyéb dolgokat, amelyekről egy kezdő felhasználónak halvány fogalma sem lehet.

Gyártó:	Interactive Video Systems
Ár:	239 DM (csak a kontroller)
Átvitel I/O:	DMA
Megjelenés:	3
Egyszerűség:	3
Különlegesség:	3
Ár:	2
Átlag:	3,25

Ha a kontroller megvételénél a pénz nem szempont, akkor legjobb a MicroBotics Hard-Frame-je mellett dönteni, mert ez a legjobb SCSI interface.

Ha nem dúskálunk a pénzben, de mégis jó megjelenésű rendszert szeretnénk, akkor a Xetec FlachCard-ját javasolhatjuk. Ha memóriabővítésre is szeretnénk szert tenni, akkor kicsivel nagyobb befektetésért az Impact A2000 a legjobb vétel.

Ha a könnyű kezelhetőség a fő szempont, akkor az Impact, vagy a Supra ellenőröket kell megvásárolni.

Talán a legfontosabb vásárlási tanács, hogy ne vegyük meg vakon, egymagában az A2090A kontrollerrel, csak azért, mert a Commodore embléma fémjelzi. Sok problémája van az overscan-nel, és mivel olyan plussz tulajdonságai is vannak, amelyeket szinte senki sem tud kihasználni, magas ára miatt csak fölösleges pénzkidobásnak számít.

Ebből következik, hogy hasonló problémák merülnek fel az Amiga 2000HD vagy az ennél nagyobb rendszerekben (A2500), mivel ezek is magukba foglalják az A2090A ellenőrt.

A bemutatott ellenőrökön és a HD-ken kívül még igen sok másfajta, elsősorban az említett konfigurációk különböző képességű változataival találkozhatunk a nyugati standokon, például a Commodore csúcstermékével az 1998 DM-es A2091 106MB Quantum típusal, vagy a Supra 2449 DM-es A500XP 105MB + 2MB RAM gyártmányával.

Végezetül még megemlítjük a nemrég bemutatott és még kint is igen drágának minősített 9490 DM-es SONY-650 OPTICAL DRIVE-ot, amely sebességével és hatalmas tárolókapacitásával igen csak átforgalmazhatja a korszerű háttértárolólól alkotott elképzeléseinket.

(Táblázatok a következő oldalon)

Amiga 500

A PerfTest eredményei:

A második adat egy overscan, hi-res, interlace, 4 bitlapos képernyő bekapcsolt állapotában lett meghatározva.

	Méret (byte)	Read (K/sec)	Write (K/sec)
SupraDrive	50K	52.8/36.3	51.6/34.2
	100K	53.6/37.3	51.5/34.6
	250K	54.9/35.9	51.5/34.5
Impact A500	50K	156.4/105.7	138.4/84.2
	100K	162.9/113.5	141.4/85.4
	250K	168.6/122.0	143.9/87.5
SA500	50K	181.8/72.2	130.4/48.7
	100K	204.8/75.1	140.5/50.5
	250K	223.1/79.8	143.9/51.4
SA500/Bernoulli	50K	91.2/55.2	68.7/40.6
	100K	92.3/61.0	91.9/44.2
	250K	112.7/66.8	103.2/46.1
Fasttrak jr.	50K	126.4/45.9	107.3/42.7
	100K	148.9/49.7	127.3/45.7
	250K	245.7/82.8	196.4/70.9
Fasttrak	50K	322.6/136.3	256.4/107.1
	100K	375.0/148.2	286.4/111.2
	250K	694.4/266.1	415.5/163.4
Escort	50K	52.2/48.6	51.1/44.6
	100K	54.2/50.5	51.6/45.0
	250K	55.0/52.4	51.2/45.0

Amiga 500

Az első Dperf 2.0 teszt eredményei:

DRIVE	File írás	File törlés	Listázás
SupraDrive	11/sec	22/sec	104/sec
Impact A500	12/sec	20/sec	104/sec
SA500	12/sec	28/sec	95/sec
SA500/Bernoulli	7/sec	11/sec	99/sec
Fasttrak	14/sec	31/sec	102/sec
Fasttrak jr.	10/sec	12/sec	88/sec
Escort	14/sec	27/sec	101/sec

Amiga 500

A második Dperf 2.0 teszt eredményei:

DRIVE	Méret (byte)	Read (K/sec)	Write (K/sec)
SupraDrive	32K	54.4	50.0
	64K	55.2	51.0
	256K	55.4	51.4
Impact A500	32K	176.8	142.6
	64K	176.8	142.2
	256K	174.8	141.8
SA500	32K	197.5	125.9
	64K	217.5	137.2
	256K	226.0	145.2

(A táblázat folytatódik)

(A táblázat folytatása)

DRIVE	Méret (byte)	Read (K/sec)	Write (K/sec)
SA500/Bernoulli	32K	116.2	63.2
	64K	123.6	80.6
	256K	124.5	102.2
Fasttrak	32K	360.6	253.7
	64K	400.2	274.5
	256K	740.5	418.1
Fasttrak jr.	32K	161.7	111.3
	64K	165.6	124.1
	256K	274.4	198.3
Escort	32K	54.8	50.1
	64K	54.8	50.8
	256K	55.4	50.8

Amiga 2000

A PerfTest eredményei:

A második adat egy overscan, hi-res, interlace, 4 bitlapos képernyő bekapcsolt állapotában lett meghatározva.

	Méret (byte)	Read (K/sec)	Write (K/sec)
A2090A	50K	327.5/10.0	269.8/82.3
	100K	433.5/17.5	320.2/85.4
	250K	526.7/30.8	343.6/65.1
HardFrame	50K	366.7/246.7	337.1/182.5
	100K	449.8/306.4	367.2/201.5
	250K	567.7/384.8	366.9/206.0
FastCard	50K	343.3/136.1	253.4/103.8
	100K	382.7/148.2	286.8/110.4
	250K	702.2/264.5	413.5/160.6
SupraDrive	50K	271.2/54.9	259.5/49.9
	100K	324.0/56.5	266.9/50.0
	250K	376.3/58.1	275.6/49.9
Impact A2000	50K	218.3/168.9	161.1/173.2
	100K	244.8/176.9	159.6/114.2
	250K	268.0/195.3	159.2/112.9
Flash! Card	50K	51.7/48.5	51.4/44.9
	100K	53.8/50.6	51.3/44.8
	250K	54.8/51.9	51.0/44.9
TrumCard	50K	242.3/103.4	198.9/83.9
	100K	276.2/112.2	222.2/90.3
	250K	310.0/119.9	231.5/92.2

Amiga 2000

Az első Dperf 2.0 teszt eredményei:

DRIVE	File írás	File törlés	Listázás
A2090A	15/sec	45/sec	100/sec
HardFrame	18/sec	54/sec	103/sec
FastCard	14/sec	32/sec	102/sec
SupraDrive	15/sec	20/sec	102/sec
Impact A2000	13/sec	34/sec	105/sec
Flash! Card	13/sec	24/sec	104/sec
TrumCard	14/sec	45/sec	100/sec10

Amiga 2000

A második Dperf 2.0 teszt eredményei:

DRIVE	Méret (byte)	Read (K/sec)	Write (K/sec)
A2090A	32K	463.2	247.3
	64K	537.2	296.9
	256K	583.8	337.4
HardFrame	32K	569.9	289.7
	64K	590.4	330.2
	256K	605.4	352.3
FastCard	32K	314.7	233.6
	64K	381.0	263.7
	256K	390.7	266.9
SupraDrive	32K	314.7	233.6
	64K	381.0	263.7
	256K	390.7	266.9
Impact A2000	32K	265.3	157.3
	64K	269.7	154.3
	256K	274.5	154.2
Flash! Card	32K	54.8	49.9
	64K	54.7	50.3
	256K	55.2	50.4
TrumpCard	32K	291.6	213.6
	64K	293.2	218.6
	256K	291.6	230.4

C64-es tippek kezdőknek

Betöltésjelzés a programban

Ellentétben azokkal a törekvésekkel, amelyek arra irányulnak a tippek/trükkök sorozatban, hogy a számítógép valamilyen visszajelzését elnyomjuk, most arról fogunk gondoskodni, hogy a gép a betöltéseknél extra üzenetet írjon ki. Egy ilyen megoldás különösen akkor nyugtatja meg a felhasználókat, ha több programrész utántöltése történik. Ilyenkor az ember (jelzés híján) nem tudja, éppen melyik rész is van soron. Ahhoz hogy ezt megvalósítsuk, az ismert programozott parancsmód trükköt kell alkalmaznunk. Ennek a trükknek a lényege az, hogy a számítógépet arra készítjük, hogy a parancsmódban automatikusan hajtson végre bizonyos dolgokat anélkül, hogy mi egy gombot is lenyomnánk.

Ehhez először a billentyűzet puffer számláló regiszterébe (ez a 198-as tárolócella) az 1-es értéket írjuk, magába a pufferbe (ami a 631-es címen kezdődik) a RETURN gomb kódja (13) kerül. A számítógép a számlálóregiszter földolgozásakor (erre akkor kerül sor, ha egy program futása befejeződött, vagy a programban egy INPUT vagy GET utasítás kerül sorra) úgy találja, hogy egy billentyű lenyomásra került. Ezért utána néz a pufferben, melyik gomb kódja került oda. Miután mi a RETURN-t választottuk, ez hajtódik végre. Most az INPUT/GET megoldás nem érdekel minket, csak azt jegyezzük meg, hogy a program

befejezésekor indulhat a programozott parancsmód. Nos ha utána gondolunk, akkor kitalálhatjuk, hogy a RETURN gomb hatására a számítógép editorja nekiáll földolgozni azt a sort (80 karakter), ahol a kurzor éppen áll. Mivel mi a programunkat folytatni akarjuk, ezért egy CONT utasítást írunk majd le. Ehhez viszont tudni kell, hol is áll a kurzor, illetve mi van még ezen kívül a képernyőn. A legegyszerűbb, ha töröljük az egészet, a kurzort a harmadik sorba visszük, és oda írjuk a CONT-ot. Hogy az egész működjön, a programot a STOP-pal kell befejezni (és nem END-del). Mihelyt a program futása megszakad, a számítógép nekiáll a billentyűzet puffer kiértékelésének. A fentebb elmondottak szerint, anélkül hogy kézzel egyetlen billentyűt lenyomnánk vagy beírnánk valamit, a gépet a program folytatására utasítjuk. Elméletileg ugyanonnan folytatjuk a munkát, ahol azt a STOP-pal befejeztük, mégis van különbség:

Ha ilyen esetekben egy programot töltünk vagy mentünk, a számítógép operációs rendszere visszajelző szöveget ír a képernyőre. Ha azt szeretnénk, hogy a STOP hatására a BREAK IN... jelzés ne legyen látható, adjuk ki a STOP előtt az alábbi utasítást:

```
POKE 646,PEEK(53281)
```

Ilyenkor a szöveg színe megegyezik a háttérszínnel, így a kírás „láthatatlan” lesz. Az általunk használandó program tehát az alábbiakat tartalmazza:

```
10 POKE198,1:POKE631, 13:POKE646,PEEK(53281):PRINT
" {clr,3 crsr down}CONT{home} " :STOP
```

(A kapcsos zárójelben álló kifejezések azt jelentik, hogy ott az adott felirat szerinti gombot nyomjuk meg.)

A 20. sorban vissza kell állítani a kurzorszínt, és ezután jöhet a betöltési utasítás. Ez a megoldás csak gépi kódú programrészek töltéséhez használható így. (A programok utántöltése témában már írtunk jónéhány cikket.) Ott is figyelembe kell venni, hogy a töltések után mindig egy GOTO(első programsor) utasítás hajtódik végre. Ha tehát nem figyelünk, a töltő rész végtelen ciklusban fönnakad. A 10. sor után tehát az alábbiakat írhatjuk:

```
20 POKE 646,1:A = A + 1
30 IF A = 1 THEN LOAD "1. rész",8,1
40 IF A = 2 THEN LOAD "2. rész",8,1
50 IF A = 3 THEN LOAD "3. rész",8,1
60 IF A = 4 THEN LOAD "4. rész",8,1
70...
```

Ennek a megoldásnak az a különlegessége, hogy a LOAD utasítások a programszövegben szerepelnek. Az eddig ismertett megoldásoknál a programozott parancsmód LOAD utasítását a képernyőre kellett írni PRINT sorokkal. Másrészt pedig a betöltésről szóló üzenetet a gép írja ki, nem kell nekünk erre értékes tárolóhelyet, programsorokat elfecsérelnünk (pl. PRINT „töltöm az 1. részt” stb.).

A 631-es címen kezdődő billentyűzet puffer tíz karaktert képes befogadni. Ennél több programozott parancs utasítás beadása tehát nem lehetséges.

Mindezek segítségével viszonylag egyszerűen elérhető a betöltéseknél a LOADING... vagy a SEARCHING... feliratok kiadása. A billentyűzet pufferrel azonban még sokkal több dolog is végezhető. Betölthetünk programokat, amelyeket azután egy adott sornál indítunk el. Vagy meghatározott tárolóterületet tölthetünk be (mondjuk egy szöveges vagy grafikus képernyőt). Szívesen vennénk minden olyan ötletet, ami ehhez a témához kapcsolódik.

Sz. B.

A jövő beviteli készülékei

Az egerek a C64 és a C128 esetében is egyre érdekesebbek lesznek. De miben különböznek egyáltalán az egerek? Hogyan kell egy programnak lekérdeznie ezt a beviteli készüléket?

A C64 nagyobb kollégái számára, mint amilyen az Amiga és az Atari, egy egér elhagyhatatlan. Ezeknek a számítógépeknek a felhasználói felületei teljesen erre vannak beállítva. De a C64-esnek is egyre érdekesebb lesz ez a beviteli egység, főleg a GEOS, az új operációs rendszer miatt. Az egyenesen kínálja az egérrel való kezelés lehetőségét a felhasználónak.

A Commodore egér

A Commodore cég házi rágcsálója kívülről úgy néz ki mint az Amiga egér. Mivel azonban sem az Amiga egér nem működik a C64-esen, sem pedig a C64-esé az Amigán, azért a belsősegeket tekintve másnak kell lenni a fölépítésnek. Meglepetésre az egér belseje szinte semmiben nem különbözik a C64-hez elsőként kínált NCE-egérétől. Ráadásul több készülék is megjelent a piacon a C64/C128-hoz a Reisware a Schneider és a Rushware cégeket képviselve. (A Schneider cégnek egyébként nincs köze a hasonló nevű CPC számítógépet forgalmazó céghez.) Nos ezeknek a szétszedése is hasonló eredményt hozott. A Rushware kivételével ezek is úgy néztek ki mint az NCE ill. a Commodore egér.

De hogy a dolgot még tovább élezzük, ezért az Amiga és az Atari egér sem került el a boncolást. A rokonságot és az egy fajhoz tartozást egyik sem tudja letagadni. A Schneider egér egyébként egy az egyben azonos a Reisware egérrel, csak az állatka „bőre” különbözik attól (az sem sokban).

Az egérpereputty

A tapasztalt nagy azonosság természetesen indokolta, hogy ennek alaposabban utánajárjunk. Nos a kutatások alapján megtudtuk, hogy a mechanikát, amely egyébként igazán jó, minden egyes egérhez egy japán nagyvállalat állítja elő. Ez a vállalat nem forgalmazza azt a saját nevén, hanem átadja más nagy felvevőknek. Az illesztő-NYÁK gyártását is úgy néz ki egy cég tartja kézben. Az összes vizsgált panelon ugyanis a „Mitsumi” feliratot találtuk. Tehát a Rushware egér kivételével az összes többi egér belső szervei egy műhelyből származnak. Csak az adott számítógépre illő kapcsolás más.

De mit jelent ez a mi C64-esünk esetében? Amikor a C64-es számítógépet fejlesztették, akkor nem is lehetett gondolni egy egér használatára. Ezért abból indultak ki, hogy ahhoz egy joysticket fognak használni. Azonkívül egészen kevés kivételtől eltekintve a programokat is a joystickekkel való kezelésre állították be. Ezt a tényt tehát a C64-hez készülő egér fejlesztésénél is figyelembe kellett venni. Nézzük csak meg a C64 és a C128 control (joystick) portjának ábrázolását:

láb	jel	megjegyzés
1	joy0	logikai 0 = fölfelé
2	joy1	logikai 0 = lefelé
3	joy2	logikai 0 = balra
4	joy3	logikai 0 = jobbra
5	pot Y	paddle
6	button	logikai 0 = tűzgomb
7	+5V	üzemi feszültség
8	GND	föld
9	pot X	paddle

A gép kézikönyvében megtaláljuk a port kivezetéseinek számkiosztását kívülről nézve.

Mivel szinte egyetlen egy C64 program sem képes feldolgozni az egér impulzusokat, amelyek egy megtett útszakaszt jelölnek, ezért a technikusok igyekeztek az egérrel egy joysticket szimulálni. Gyakorlatilag ez azt jelenti, hogy a fölfelé irányuló egérmozgásnál az egér 1-es csatlakozóján a jel logikai nulla szintre áll be impulzusok nélkül. Csupán az igen lassú egérmozgás esetén keletkeznek impulzusok. A C64-es egerek csatlakozóinak kiosztása az alábbi táblázat mutatja.

A Commodore, a Reisware és az NCE egér csatlakozásainak foglaltsága.

láb	jel	megjegyzés
1	fölfelé	logikai 0 = fölfelé
2	lefelé	logikai 0 = lefelé
3	balra	logikai 0 = balra
4	jobbra	logikai 0 = jobbra
5	NC	nincs csatlakoztatva
6	bal bill.	logikai 0 = billentyű megnyomva
7	+5V	üzemi feszültség
8	GND	föld
9	jobb bill.	logikai 0 = billentyű megnyomva

A joystick szimulációnak azonban vannak hátrányai is: Például a képernyőn csakis vízszintes, függőleges és (45 fokos) átlós mozgásokat lehet visszaadni, ahogy azt a joystick esetében megszoktuk. A mozgás sebessége sem felel meg a képernyőn az egér mozgási sebességének. Egy joysticket szimuláló egértől tehát nem várhatjuk el az igazán precíz mozgás leképzést, mint azt az Amigánál vagy az Atari ST-nél megszokhattunk. Sajnos ez még a GEOS operációs rendszer esetében is így volt.

Ettől csak az NCE egér tér el, együtt használva a saját grafikus programmal. De erre később visszatérünk még egyszer. Ehelyütt azonban meg kell jegyezni, hogy a C64-es Commodore, Reisware, Schneider, és bizonyos korlátozásokkal az NCE egerek a belső fölépítésben és a meglévő szoftverekhez való viszonyban abszolút azonosak.

De miért reagálnak az Amiga és az Atari egerei olyan precízen? Nos ennek oka egy más illesztő kapcsolás és az erre kifejezetten beállított szoftverek miatt. Az egerek a vízszintes és a függőleges irányban két négyzögjelet adnak le, amelyek egymáshoz képest fáziseltolásban vannak. A fáziseltolás irányából tudja a számítógép meghatározni, hogy a mozgásirány pozitív vagy negatív-e. Pozitívnak lehet definiálni mondjuk a föl és a jobbra

irányuló mozgást. Negatív lesz akkor a lefelé és a balra. De ennek meghatározása a programozótól függ.

De térjünk vissza az egerekhez, amelyek egy joysticket szimulálnak. Hogyan lehet ezeket a mozgásokat a control porton keresztül lekérdezni?

Egérprogramok

Ehhez néhány alapinformációt kell elsajátítani. A control portot a két Complex Interface Adapter (CIA) közül az első igazgatja. Az a \$DC00—\$DCFF (decimálisan 56320—56575) regisztereket foglalja el. A 2-es control port négy mozgásirányát a \$DC00 (56320) regiszterben a 0—3. bitek segítségével jelezhetjük. Ha a joystick szimuláló egerek egyikét csatlakoztatjuk a joystickportra, és azt nem mozdítjuk meg, akkor minden egyes említett bit a logikai egyes értéket kapja. Ha megmozdítjuk az egeret, akkor a megfelelő bitet nullára visszük. A négy bit kiosztása az alábbi: 0 fölfelé, 1 lefelé, 2 balra, 3 jobbra. A negyedik bit értéke akkor lesz nulla, ha megnyomjuk a bal egérgombot.

A következő \$DC01-es regiszter (56321) hasonló módon kezeli az 1-es control portot.

BASIC-ből az egyes bitállapotokat az AND logikai operátorral lehet ellenőrizni. Kiolvassuk a megfelelő regisztert PEEK utasítással, majd összekapcsoljuk az értéket a megfelelő értékekkel. (1=bit 0, 2=bit 2, 8=bit 3, 16=bit 4 stb.) Ha az eredmény megegyezik az adott értékkel, akkor a vizsgált bit be van kapcsolva. Ez az eljárás teljesen megegyezik a joystick lekérdezésekkel, ezért a konkrét BASIC programozásra most nem térünk ki. Olyan kurzormozgatást, amely a képernyőn többé-kevésbé megfelel az egérmozgásnak, már BASIC-ből nem lehet igazán megvalósítani. Az ilyen időben kritikus folyamatok esetében már a gépi kódú programozáshoz kell folyamodni. Hasznos módszer az interruptvezérelt lekérdezés. Végeredményben azonban minden joystick szimuláló egér esetében megmarad a precízebb és gyorsabb mozgítás igénye. Ez különösen a grafikus programok és a GEOS esetében van így. Csodálatra méltó azonban az NCE egér pontossága a mellé adott grafikus programmal együtt. Ennek oka pedig a saját egér módban keresendő, amely minden egyéb — Commodore, Reisware és Schneider — egérrel való hasonlóság ellenére fennáll.



Ha adatvédelemről esik szó, akkor a lényeg az, hogy találjunk egy megfelelő algoritmust a kódoláshoz. Elsősorban arra kell ügyelni, hogy ezt az algoritmust ne lehessen könnyen föltörni. Vagyis csak nagy nehézségek árán sikerüljön illetékteleneknek a kódolt adatokból az eredetit visszakapni. Ráadásul akkor is, ha netán az információ egy része originált formában valahogy a rendelkezésre áll. Másrészt a kódoló rutin lehetőleg gyorsan dolgozzon, s az sem árt ha nem túl hosszú. A legfontosabb pedig az, hogy legyen egy másik algoritmus, amely a kódolt szöveget visszaalakítja „használatóvá”.

Vannak olyan eljárások, amelyek négy bitből egy táblázat segítségével 5-öt csinálnak, majd ezt fogják össze byte-tá. Ezt alkalmazza a Commodore a lemezinformáció rögzítésére (GCR kódolás). Ezt a kódot (ismeretek hiányában) csak igen nagy nehézségek árán lehet visszaalakítani. A hátrány az, hogy az így nyert kódolt szöveg hossza az eredetihez képest nő (5/4 lesz), 400 byte-ból tehát 500 lesz. Az adattovábbítás vagy a tárolás során ez zavaró. Vagyis általában más megoldást alkalmaznak.

Különböző algoritmusok ismeretese és használatosak. Például az, ahol az adott byte-hoz egy meghatározott értéket adnak hozzá (ADC kód), vagy ahol minden byte meghatározott biteit invertálják (EOR kód). Nagyon kedvelt a byte-okban a bitfelcserélésen alapuló kódolás, vagy ahol a biteket meghatározott rendszerint arrébbcsúsztatják (rotáció). Mindeme algoritmusok teljesen működőképesek és könnyen programozhatók. Hátrányuk

viszont az, hogy megfelelő próbálkozással (gyorsabban vagy lassabban) visszafejthetők, főleg ha az eredeti információ egy része a rendelkezésre áll. Komoly adatvédelemre tehát ezek nem megfelelők.

Természetesen van olyan algoritmus, ahol a rész ismerete esetén sem lehet könnyen föltörni a rejtett információt, s ahol az algoritmus meglehetősen tömör és gyors, miközben a kódolt információ nem lesz hosszabb mint az eredeti. Például a Delta kódolás ilyen. Ezt a módszert még a titkosszolgálatok is alkalmazzák. A Kódolandó byte-okat bitsorozatként írják föl. A lánc első bite változatlan marad. Most a második bitet nézzük. Ha az különbözik az előzőtől, 1-es lesz belőle, ha nem, akkor nulla. Hasonlóan nézzük a harmadik, negyedik stb. bitet is, mindig az előzőt nézve. Új byte-re lépve az előző byte utolsó bite lesz az összevetés alapja. A legjobb ha ezt egy példán keresztül szemlél-tetjük. Kódoljuk az alábbi öt byte-ot ezzel a módszerrel:

Az eredeti:	\$62	\$A4	\$F9	\$07	\$D2
Binárisan:	01100010	10100100	11111001	00000111	11010010
Kódolva:	01010011	11110110	10000101	10000100	01111011
Hex:	\$53	\$F6	\$85	\$84	\$3B

Látható, hogy az eredeti \$62, \$A4, \$F9, \$07, \$D2-ből a \$53, \$F6, \$85, \$84 és a \$3B lett. Ezt „első levezetésnek” szokták mondani. A bineáris értékeket nézve látható, hogy a második sorban akkor lesz a bitek értéke 1, ha fölötte a bitérték változott. Az így kapott kódolt sort természetesen még egyszer vagy akár többször is újrakódolhatjuk ezzel a módszerrel. Így plusz biztonságot teremthetünk:

Az eredeti:	\$53	\$F6	\$85	\$84	\$3B
Binárisan:	01010011	11110110	10000101	10000100	01111011
Kódolva:	01111010	00001101	11000111	01000110	00100110
Hex:	\$7A	\$0D	\$C7	\$46	\$26

A kapott \$7A, \$0D, \$C7, \$46, \$26 értékek jelentik a második levezetést. Ezt tetszőleges számban ismételtethetjük meg. Ha egy így kódolt szöveget továbbadunk, és a „vevő” nem ismeri az eljárást, az eredeti információt nem tudja rekonstruálni. Főleg, ha pluszként a fentebb ismertetett módszerekkel (pl. EOR kód) plusz védelmet iktatunk közbe. Ekkor már valóban szinte lehetetlen lesz a kódtörés.


```

1 REM *****
2 REM *
3 REM *      DELTA CODER      *
4 REM *
5 REM *****
6 :
7 :
8 PRINT CHR$(147)"DATA BEOLVASAS ES ELLENORZES ...";J=00820:VE=00995:P=J
9 FOR B=0 TO 7:READ A$
10 L=ASC(MID$(A$,2,1))
11 H=ASC(MID$(A$,1,1))
12 L=L-48:IF L>9 THEN L=L-7
13 H=H-48:IF H>9 THEN H=H-7
14 PRINT"XXXXXXXXXXXX"X:P=P+1
15 IF H>15 OR L>15 THEN 17
16 A=H*16+L:POKE J+B,A:T=T+A:NEXT B:READ A:IF A=T THEN 18
17 PRINT:PRINT"DATA HIBA ...  SOR:"PEEK(64)*256+PEEK(63):END
18 T=0:J=J+8:IF J<VE THEN 9
19 PRINT"XXXXXXXXXXXX"X:END
20 DATA 4C,47,03,80,40,20,10,08,0398
21 DATA 04,02,01,7F,BF,DF,EF,F7,1034
22 DATA FB,FD,FE,20,F1,B7,8E,58,1444
23 DATA 03,A5,7A,8D,60,03,A5,7B,0818
24 DATA 8D,64,03,A2,00,D0,01,60,0711
25 DATA CE,58,03,A9,00,85,7A,A9,0890
26 DATA 00,85,7B,20,F1,B7,86,F7,1093
27 DATA 20,FD,AE,20,8A,AD,20,F7,1081
28 DATA B7,84,AC,85,AD,20,FD,AE,1252
29 DATA 20,8A,AD,20,F7,B7,84,AE,1111
30 DATA 85,AF,A9,01,85,F8,A0,00,1019
31 DATA B1,AC,85,F9,A5,F9,39,37,1257
32 DATA 03,F0,02,A9,01,A6,F8,D0,1037
33 DATA 2B,A6,F7,F0,0D,AA,F0,06,1125
34 DATA A5,FA,49,01,85,FA,A5,FA,1287
35 DATA 10,04,C5,FA,85,FA,F0,04,1094
36 DATA B9,37,03,2C,A9,00,8D,C3,0792
37 DATA 03,A5,F9,39,3F,03,09,00,0549
38 DATA 85,F9,A5,FA,85,FA,46,F8,1498
39 DATA C8,C0,08,90,BF,A0,00,A5,1060
40 DATA F9,91,AC,20,DB,FC,20,D1,1310
41 DATA FC,90,AB,4C,57,03,00,00,0733
    
```

Ennek hatására a K (kezdő) és V (végcím) között álló információt X-szer delta kódoljuk. A SYS 820,4,0,1024,2024 utasítás hatására a képernyőt a (1024—2023) négyszer kódoljuk. A dekódoló parancs csak abban különbözik az előzőtől, hogy a második paraméter értéke nullától különbözik (pl. 1 lesz):

```
SYS 820,X,1,K,V+1
```

A példánkban a kódolásnál a képernyőből kialakított „salátát” a SYS 820,4,1,1024,2024 utasítással alakíthatjuk vissza „értelmessé”. Az egész természetesen fordítva is működik. Egy tartományt ugyanis kódolhatunk a **SYS 820,1,0,K,V+1**-gyel is, de ekkor a **SYS 820,X,0,K,V+1** lesz a visszaalakító.

Végezetül hadd említsünk meg két korlátozást. A terület vége technikai okokból nem lehet 65535, mert ekkor a V értéknek 65536-ot kellene megadni. Ez azonban már egy 17 bites szám lenne, amit a processzor nem tud földolgozni. Azonkívül a V értéke nagyobb kell legyen mint a K. Mivel pedig a kódoló algoritmus viszonylag bonyolult, a gyorsaság sem olyan óriási. 8 kbyte szöveg egyszeri kódolása kb. 1 másodpercet vesz igénybe. Különböző klassz hatásokat lehet elérni, ha a grafikus tárolót kódoljuk.

A mellékelt Delta Coder program ezt az eljárást valósítja meg. A betöltése az alábbi utasítással történhet:

```
LOAD "DELTA CODER 820", 8,1
```

A betöltést követően korrigáljuk a BASIC mutatót NEW-val. Most az elrejtendő kódot bárhová behívhatjuk a tárolóba:

```
SYS 820,X,0,K,V+1
```

Raszterinterruptok

Tulajdonképpen már számos módszer létezik, ahol a kifinomított interrupt technikával részekre lehet osztani a képernyőt, vagy mondjuk meg lehet azt szabadítani a kereteitől. Gyakorta szükséges egy rasztersoros interruptot pontosan egy definiált X pozíciónál aktiválni, mert mondjuk csak így lehet az oldalsó kereteket eltávolítani. Eleddig csak megfelelő számú NOP paranccsal volt lehetséges az időkonstans úgy megválasztani, hogy ne lépjen föl remegés. Egy mindezzel párhuzamosan dolgozó programra már nem is lehetett gondolni, hiszen a parancsok hosszának különbözősége miatt az interrupt esetleg hét

mikromásodperccel is késhet, amely a Timinget teljesen összekuszálhatja. A mellékelt lista azonban egy olyan rutint mutat, amely a képernyő felső szélén egy rövid színes csíkot hoz létre, amelyet (majdnem) sehogy sem tudunk „kizökkenteni” a helyéről. E fölött egy olyan csíkot találunk, amelyet a „hagyományos” módszerrel hozunk létre, s amely például gombnyomásra ideoda „lengedez”. De térjünk rá a program leírására:

A rasztersoros interrupt inicializálását az alábbi utasítással érhetjük el: **SYS 49152**

Ezután visszaugrunk a főprogramba. Maga az interrupt rutin a \$C020-as címen kezdődik (lásd a forrásszöveget is).

Először töröljük a videochip IRQ regiszterét. Ezután megfelelő számú NOP paranccsal késleltetést végzünk, amely okán a


```

1 REM *****
2 REM"* C= UJSAG          SORSZAM:  *
3 REM"* SZUPER RASZTER INTERRUPT  *
4 REM"* PROGRAM:          *
5 REM *****
6 :
7 :
8 PRINT CHR$(147)"DATA BEOLVASAS ES ELLENORZES ...";J=49152:VE=49327:F=J
9 FOR B=0 TO 7:READ A#
10 L=ASC(MID$(A#,2,1))
11 H=ASC(MID$(A#,1,1))
12 L=L-48:IF L>9 THEN L=L-7
13 H=H-48:IF H>9 THEN H=H-7
14 PRINT"#####P";F=F+1
15 IF H>15 OR L>15 THEN 17
16 A=H*16+L:POKE J+B,A:T=T+A:NEXT B:READ A:IF A=T THEN 18
17 PRINT:PRINT"DATA HIBA ... SOR:"PEEK(64)*256+PEEK(63):END
18 T=0:J=J+8:IF J<VE THEN 9
19 PRINT"#####KESZ":END
20 DATA 78,A9,20,8D,14,03,A9,C0,0846
21 DATA 8D,15,03,A9,01,8D,1A,D0,0710
22 DATA 8D,0D,0C,A9,20,8D,12,00,0950
23 DATA A9,1B,8D,11,D0,5B,60,00,0746
24 DATA AD,19,00,8D,19,00,EA,EA,1248
25 DATA EA,EA,EA,EA,EA,EA,EA,A2,1800
26 DATA FF,A0,00,8E,00,0C,8C,02,0919
27 DATA DC,8E,03,0C,8E,01,0C,8C,1088
28 DATA 01,0C,8E,01,0C,8C,20,00,0964
29 DATA A9,0E,8D,20,D0,AD,13,00,0964
30 DATA 8E,02,0C,8C,03,0C,8E,01,0870
31 DATA 0C,A2,7F,8E,00,0C,4A,4A,1019
32 DATA 4A,8D,68,C0,90,00,B8,50,0919
33 DATA 12,EA,EA,EA,EA,EA,EA,EA,1656
34 DATA EA,EA,EA,EA,EA,EA,EA,EA,1072
35 DATA EA,EA,EA,EA,EA,EA,EA,EA,1872
36 DATA EA,EA,EA,EA,EA,EA,EA,EA,1872
37 DATA EA,EA,EA,EA,EA,EA,EA,EA,1872
38 DATA EA,EA,EA,EA,EA,EA,EA,EA,1872
39 DATA EA,EA,A2,09,CA,D0,FD,EA,1536
40 DATA EA,EA,A9,01,8D,20,D0,A9,1188
41 DATA 0E,8D,20,D0,4C,31,EA,AA,0924
    
```

READY.

színes csík körülbelül a képernyő közepén jelenik meg. Ezt a színes csíkot a \$C045—\$C04C címeken „készül”.

De mire szolgál az a sok, a \$DC00 területre irányuló STX és STY parancs? Nos először itt a CIA1 A portját bevitelre állítjuk, majd a B portot kivitelre. Ezután a B portnál az összes bitet bekapcsoljuk, töröljük majd ismét bekapcsoljuk. „Hát ez meg mire föl szükséges?” — kérdezhetjük. A C64-es hardverének ismerői tudják, hogy az 1-es joystickporton rendelkezésre áll egy Lightpen (fényceruza) bemenet. Ez itt azonos a tűzgomb bemenetével, akkor mód van a CIA-n keresztül impulzusokat adni a VIC Lightpen bemenetére. A videochip egy interruptnál, amelyet mi most a CIA1-gyel hoztunk létre, az elektronsugár aktuális pozícióját a \$D013-as regiszterbe írja. Most helyreállítjuk a billentyűzet port eredeti állapotát.

Az akkumulátorban találjuk azt az X pozíciót, amelynél a B port minden bite nulla értéket kapott. A pontosság két képpont. A legkisebb egység, amellyel a C64-es egy folyamatot irányítani tud, egy mikroszekundum (egy órajelciklus). Ez alatt az idő alatt azonban a VIC nyolc képpontot „ír” a képernyőre. Ezért az akkumulátorban álló értéket négyvel el kell osztani, hogy a kerettől való távolságot órajelciklusban kapjuk meg (a \$C05E és a \$C05F címek).

A következő lépések a processzort egy kis ideig föltartják. A számot úgy kapjuk meg, hogy 64-ből levonjuk az akkumulátor értékét. Így a processzort pontosan szinkronizáljuk a rasztersugárhoz. Mivel nincs olyan parancs, amely pontosan egy ciklusidőt venne igénybe, ismét trükkhöz kell folyamodni. Az akkuban álló értéket ismét elosztjuk kettővel (\$C060), és az eredményt

a \$C068-as címre írjuk. Ott azonban egy BVC parancs operandusa áll, amely az előtte álló CLV miatt közvetlen ugrássá válik. Ez a branch parancs akkuérték/2 byte-tal arrébb ugrik. Mivel az ugrási cél 32 NOP parancson belül van, és egy NOP parancs éppen két órajelciklust vesz igénybe, mindez éppen „beleillik” az elépzeléseinkbe. Például ha az akkumulátorban a 10-es decimális érték áll, akkor 5 NOP-ot ugrunk át, ami éppen a számunkra szükséges 10 órajelciklust vesz igénybe.

De mi történik a páratlan akkumulátor értékekkel? Az osztásnál nem vész el az utolsó bit? Ez nem okoz egy ciklusidőnyi pontatlanságot? Nem igazán, mert a kitöltő bit a processzor státuszregiszter carry flagjébe kerül. Ez egy igen fontos tulajdonsága az LSR parancsoknak. Ha tehát a carry flag be van kapcsolva, akkor a programunknak egy ciklusidővel kevesebbet kell várni, mint a törölnél. A megfelelő parancsokat a \$C064-es címtől kezdve találjuk, s az egy BCC utasítást jelent, amely az őt követő Clear Overflow-ra (CLV) mutat. Ez az értelmetlennek tűnő parancs számunkra fontos jelentést hordoz: Ha a processzor nem ugrik (a feltétel nem teljesült), akkor két, ha ugrik (a feltétel teljesült), akkor három órajelciklust használunk föl. Persze az „ugrás” ige itt kissé zavaró, hiszen a processzor minde esetben a CLV paranccsal folytatja a munkát, de a Timing egzakt betartása miatt fontos a különbség.

A többi már egyszerű. A NOP-ok után egy késleltetőhurok jön, amely a fehér csíkot pozicionálja (\$C09A—\$C01A), majd azt létrehozuk (\$C0A2—\$C0AB) és a rendszerinterruptba lépünk. Ennyi az egész!

Egy tetszőleges billentyű megnyomásakor a felső színsáv remeg, mialatt az új elven elkészített alsó olyan szilárdan áll, mintha bebetonozták volna. Egy kivétel azért van. Ha a szóköz billentyűt, a tűzgombot nyomjuk meg, vagy van csatlakoztatott fényceruza (ez mind egy vezérlő vezeték), akkor a mi rutinunk is akadozni fog, mivel ezek interruptot váltanak ki.

Ez a nagyhatású kis trükk új távlatokat nyit a raszterinterrupt programozásban. Mi lenne új mintákkal a felső és az alsó kereten? (Gondoljunk csak a \$3FFF cellára.) Vagy a képernyő érdekes fölosztására: balra grafika, jobbra szöveg? A profik számára új lehetőségek nyílnak...


```

O      ,c000  78      sei           ; interrupt letiltása           O
      ,c001  a9 20     lda #20       ; interrupt vektor           O
      ,c003  8d 14 03  sta 0314      ; stállítása a               O
      ,c006  a9 c0     lda #c0       ; rutinunkhoz                O
      ,c008  8d 15 03  sta 0315      ;                               O
      ,c00b  a9 01     lda #01       ;                               O
      ,c00d  8d 1a d0  sta d01a      ;                               O
      ,c010  8d 0d dc  sta dc0d      ;                               O
      ,c013  a9 28     lda #28       ;                               O
      ,c015  8d 12 d0  sta d012      ;                               O
      ,c018  a9 1b     lda #1b       ;                               O
      ,c01a  8d 11 d0  sta d011      ;                               O
      ,c01d  58       cli           ; interrupt engedélyezése   O
      ,c01e  60       rts
      -----
      ,c01f  00       brk           ; 0 byte                       O
      -----
      ,c020  ad 19 d0  lda d019      ; interrupt-flag             O
      ,c023  8d 17 d0  sta d019      ; törlése                     O
      ,c026  ea       nop
      ,c027  ea       nop
      ,c028  ea       nop
      ,c029  ea       nop
      ,c02a  ea       nop
      ,c02b  ea       nop
      ,c02c  ea       nop
    
```

Világújdonság



14 nyelv egy szótárban — 182 különféle
szótár egy zsebben
Bármelyik nyelvről bármelyikre.
100 000 szavas memória!

MAGYAR NÉMET
CSEH ANGOL
SZLOVÁK FRANCIA
SZERB OLASZ
LENGYEL SPANYOL
OROSZ PORTUGÁL
SVÉD DÁN

Nyelvenként — 6000 szó — 50 témakörben 550 mondat. Teljes ábécék!
8 számjegyes számológép!

Automatikus kikapcsolás! — 4 darab elemmel működik, mely tartozék!

12 900 Ft (ÁFA-val) Visszonteladóknak nagykereskedelmi ár!

Vásárlási utalvány és megrendelőlap újágunk 34. oldalán található



SZÁMALK-CED
Kereskedelmi Kft.

Magas szintű irodatechnikai szolgáltatások a SZÁMALK-CED-től!

UTAX-Fénymásolók és teljes tartozékok (12-féle géptípus)	39 900 Ft-tól + ÁFA
NIKKAM-pénztárgépek (APEH-engedélyes — visszaigényelhető)	50 000 Ft-tól
SAFAX-telefaxok (francia)	89 000 Ft-tól

KELLÉKANYAGOK:

- festékek, előhívók,
- faxpapírok,
- pénztárgépszalagok.

*Tekintse meg bemutatótermünket!
Várjuk szeretettel!*

Budapest XI., Budafoki út 109.
Tel./fax: 1810-757 vagy 1868-333/137.

CÉL A CED!

KERESZTREJTVÉNY

A Pázmán című epigramma (1830) második zárópentameteréből vettük Vörösmarty humanista hitvallását: — Vízszintes 7., függőleges 1 sorban található meg.

A meghatározásokat nem választottuk külön, hanem a számozás sorrendjében folyamatosan adjuk. Ha egy számtól mindkét irányba indul ki a megfejtés, akkor előbb a vízszintes, majd egy gondolatjellel elválasztva a függőleges meghatározása szerepel.

MEGHATÁROZÁSOK: 1. Váctól keletre község. — Vörösmarty hitvallás második része. 2. Szelén vegyjele. 3. Hangnem. 4. Ósmagyar név. — Kubai és ausztriai gépkocsik jelzései. 5. „A” nyereség. 6. Asztácium vegyjele. 7. Vörösmarty hitvallás első része. — Fizimiska. 8. Szolmizációs skála. 9. Február 13 és november 19-én van a nevenapja. 10. Csen. 11. Ékes burkolatsúly. 11/a. Mesterember. 12. Névelő. 13. Ilyen határidő is van. 14. Állóvíz. 15. Időjelző. — Kémiai elem. 16. Ilyen erőmű is van. 17. Kettőszáz római számmal. 18. Kettős betű. — Következő. 19. Csúk. — Legelő. 21. Ilyen tánc is van. 22. Csak félig apad. 23. Dátumrag. 24. ... Edgar amerikai költő volt. — Ilyen jel is van. 25. Névelős legelő. 26. Ilyen egyszerű is van. 27. Svájci város. 28. Ételfésítőt. 29. Ital. 30. Máltai és olaszországi ugyancsak máltai gépkocsik jelzései. 31. Horoszkóp jel. 32. Idegen vörös. 33. Kimondott betű. 34. Merész. 35. Vidéki vendégmarasztaló. 36. Gubós növény. — Sósav igéje. 37. Azonos a 2 függőlegessel. 38. Becézett női név. 39. Azonos a 36 függőlegessel. 40. Zala megyei helység. 41. Hangszer. 42. A Tisza mellékfolyója a Bánságban. 43. Állami illeték. 44. Helység Pest megyében. 45. Ruha kellék. — Akadály. 46. Hegy Kréta szigetén. 47.

1		2	3		4		5		6		7	8		
		9			10				11	11/a				
12	13		14			15		16		17			18	
19		20		21	22				23			24		
25					26						27			
28				29		30				31		32		
33			34		35				36				37	
		38		39				40				41		
42			43				44				45			
		46				47				48				
49	50				51				52				53	
54				55		56		57		58		59		
60			61			62	63			64	65		66	
		67				68				69			70	
71									72					

Szilaj. 48. Testrés. 49. Visszaér(!) 50. Van ilyen háború is. 51. Titán vegyjele. 52. Irányszó. 53. Azonos a 8 függőlegessel. — E név latinban: oroszlán. 54. ... Éva színművésznő. 55. Vaspálya. — Van ilyen labda is. 56. Nobélium vegyjele. 57. Cinke angolul. — Tantál vegyjele. 58. A légnymás egysége. 59. Talál. 60. Sport-

eszköz. 61. Két magánhangzó. — Nem ül. 62. Időjelző. 63. Tetejére. 64. Állati szállítás. 65. Csen. 66. Azonos a 64 vízszintes-sel. 67. Női név. — Egy másik kimondott betű. 68. Aktínium vegyjele. 69. Kémiai elem. — Kálium és oxigén vegyjele. 70. Fél márk. 71. Város Dél-Lengyelországban. 72. Nógrád megyei helység.

Képernyőtörlés oszloponként

Bizony meglehetősen egyhangú dolog a képernyőt a hagyományos **PRINT CHR\$(147)**-tel törölni. Különleges hatásokra használjuk inkább az alábbi sort:

```
50000 POKE 53281,B:POKE 53280,B:Z=-40:X=40
```

A B változóban 0—15 közötti értéket kell írni. Ez lesz a törlési színünk. A tulajdonképpeni rutin a következő sorban kezdődik:

```
50010 FOR A=1 TO 40
```

Ez lesz az első, negyvenszer végrehajtandó hurok, amely azonban egy másik hurkot is tartalmaz:

```
50020 FOR Y=1 TO 25:Z=Z+40:POKE 55296+Z,B:NEXT
```

Ebben a (huszonötször végrehajtott) hurokban a szín RAM-ot (az 55296-os címtől) a háttérszínnel töltjük föl. Az oszlopokat tehát egyenként töröljük. Mivel negyven oszlop van, ezért a fő hurkot is ennyiszor kell lefuttatni. Ennek a zárása tehát így fog kinézni:

```
50030 Z=-40+A:NEXT
```

Indítsuk el a programot, mire a képernyőt látszólag letöröljük. Azonban itt nem történik más, minthogy a háttérszín és a karakterek színét azonosra módosítjuk. Vagyis a teljes törléshez még egy parancs szükséges:

```
50040 PRINT CHR$(147)
```

Ezt a rutint akkor célszerű használni, ha programjainkat kicsit feltűnőbbé kívánjuk tenni.

Keresem a C64-hez való SUPER GRAFIK; SIMONS' BASIC; SPECTRUM-EMULÁTOR; MUSIC SHOP című programokat. Salamon Péter, 4122 Gáborján, Lenin u. 29.

C64 programok lemezen és 5,25" diszkek 35.-Ft/db ártól eladók. Válaszborítékra listát küldök. Nagy István, 1202 Budapest, Mártírok u. 151.

AMIGÁRA a legújabb programok eladóak! Akció, kaland, pornó stb. Barabás Zsolt, 2230 Gyömrő, Deák F. u. 12.

Totókulcskészítő programok (8 Kbyte-os) C64-re. Válaszborítékért tájékoztató. Nagy Miklós, 4244 Újfehértó, Pf.: 41.

Keresem a Renegade c. C64-es másolóprogram leírását. Tel: 226-1339. Kurdi Szabolcs, 1225 Budapest, Batthyány u. 60. II./4.

Programleírásokat veszek 10 Ft/db-os áron. Lehetőleg Commodore 64-es gépre (pl.: ELITE, LAST NINJA stb.). Bense János, Isaszeg, Szent László u. 45.

YO! AMIGÁRA a legújabb programok külföldről! További információért írj! Soldier, 1683 Budapest, Pf.: 88.

C64-re színvonalas lemezes játék és felhasználói programok kaphatók. Ár 5—15 Ft/program. Lőrincz Endre, 6800 Hódmezővásárhely, Somogyi B. u. 20.

Nyomtatószalagok újrafestése postai utánvétellel is. MPS 803: 155 Ft, MPS 801: 171 Ft, DFX 5000: 799 Ft. Oláhné, 2092 Budakeszi, Erkel u. 31./c. Tel.: 1-121-127.

Keresem C64-re az OXFORD PASCAL című programot, lemezen. Ghete Gábor, 2531 Tokod, Munkácsy u. 14.

Színvonalas C-PLUS/4 programok eladása és cseréje! Listát küldök! 6045 Ladánybene, Gödörállás düllő 55.

C-64+VC1541/II. tartozékokkal eladó. 27 000 Ft. Tel.: 1667-805. Burányi Márk (estefelé). Cím: 1118 Budapest, Budaörsi u. 4/a. F.3.

C+4-re keresek PASCAL programnyelvet kazettán, vagy lemezen. Ha van, dokumentációt is. Baráth Miklós, 7300 Komló, Berek u. 1/a. IV/10. Tel.(fax): (72) 82-728.

Eladom kevest használt COMMODORE-64 típusú számítógépet, meghajtoművel, printerrel és sok leprellő papírral, szakkönyvekkel együtt. Dr. Kaiser. Tel.: 156-25-75 (18 órától).

Figyelem! A Blu Angel Adventure Software márciusban jelenteti meg az eddig egyik legjobb magyar nyelvű kalandjátékát, az A pokol angyalát. C-64, lemez. Magyarországon csak a B.A.A.S.-nél lehet megvásárolni. A program előzetesét küldött lemezre ingyen felvesszük! További felvilágosítás: B.A.A.S., 2600 Vác, Eperfa u. 6.

A BLU ANGEL ADVENTURE SOFTWARE keres programozókat, grafikusokat programírás céljából, vagy kész programokat forgalmazásra átvesz. C-64, lemez. Mindenkinek válaszolunk! Kedvező fizetési és egyéb feltételek. Ha van, a telefonszámodat írd meg. Jelenkezni az alábbi címen: B.A.A.S., 2600 Vác, Eperfa u. 6.

C-64-re felhasználói és játékprogramok eladók lemezen 82—92-ig. Több mint 5000 program. K&C COMPUTERS, 3630 Putnok, Pf.: 25.

AMIGA Action Replay MK3 8500 Ft + postaköltség (német leírással, doboz nélkül). Cím: Derko, 1399 Budapest, Pf.: 701/679.

C-64-hez Action Replay MK5+2900 Ft, MK7 3600 Ft, Atomic Power +MK7 4500 Ft. Cím: Derko, 1399 Budapest, Pf.: 701/679.

C-16 +4 programokat cserélek kazettán. Listát kérek! Siklósi Géza, 3529 Miskolc, Sályi I. út 8. III./2.

Eladó Spectrum 80K + Multiface + joy illesztő + hanggenerátor + 50 kazetta 16 000 Ft-ért, vagy Commodore floppyra cserélhető. Kovácsik Tibor, Budapest XIV., Újvárosparc 4—5. IX./233. 17 óra után.

AMIGA 500 programcsere. Bogár József Zsolt, 1183 Budapest, Széchenyi u. 28.

Olcsón Commodore IC-k elektronikus alkatrészek beszerzése, C-64 és perifériáinak javítása. Tel.: 1-731-783 (üzenet).

Minőségi, izgalmas, gondolkodtató kalandjátékok C-64-re. Adok, veszek, cserélek. Pál Richárd, 6900 Makó, Petőfi u. 15.

Várom (kaland) + játékprogramozók jelentkezését program és tapasztalatcsere céljából. Pál Richárd, 6900 Makó, Petőfi u. 15.

C+4 programok eladók: 8 Ft/db. Válaszborítékért listát küldök. Bujdosó Csaba, 2340 Kiskunlacháza, Tavasz u. 7.

C+4 programokat adok-ve-szek és cserélek, ha újak. 8 Ft egy leírás. Bujdosó Csaba, 2340 Kiskunlacháza, Tavasz u. 7.

Superbase-64 programot keresek. Havasi László, 8111 Seregélyes, Fő u. 86.

AMIGA 500 + 1MB bőv., + RF modulátor, + 40 db lemez, + óra, + könyvek, + joy, + egér. Ára 44 000 Ft. Szegedi Attila, 2131 Göd, Váci M. u. 16.

C-64-re Amőba nevű játékot keresek lemezen, megvételre. Bajári Miklós, 1148 Bp., Adria sétány 9/E.

C-64-es magnósok RECORDER-JUSTAGE magnófejbeállító prg. kazettán eladó. 150 Ft. Lehoczki, 6723 Szeged, Hont F. u. 3/A.

Eladó rengeteg lemez: szuper játék és felhasználói programokkal C-64-re. (80 Ft/db!) Programmásolás már 3 Ft/db-tól lemeze és kazettára. Válaszborítékért listát küldök. Kovács Krisztián, 6750 Szeged II., Bartók B. 6.

Eladó C-64 + 1541 floppy. Érdeklődni az esti órákban a (62) 25-499-es telefonon lehet.

IGEN...

érdekel egy C-64 grafikus képűség-betűreklám program több mint 30 ékezetes betűkészlettel (kisbetű, nagybetű, írásjelek, piktogramok), szerkesztési és kiemelési lehetőségekkel.

- ismertetőt kérek, válaszbélyeget (16 Ft Bp.-re, 22 Ft vidékre) mellékelek.
- demót kérek, lemezt + válaszbélyeget, vagy 80 Ft-ot küldök.
- érdeklődöm a program forgalmazása iránt.

Név:

Cím:

Küldje el a szelvényt, vagy írjon:
Molnár Tibor, 1138 Budapest, Népfürdő u. 21/b.

ÁPRILISI

60 Ft-os
vásárlási utalvány

Beváltható készpénzes vásárlás esetén:
DIGITHALY márkabolt,
Budapest IX., Telep u. 29.
Az utalvány a vásárlási ár 5%-ig
használható fel!

Érvényes: 1992. április 30.

MAKROVILÁG
utazási irodaBeváltható
utazás megrendelése esetén

az Üllői úti főirodában az alábbiak szerint:
5 000 Ft-ig — 200 Ft kedvezmény
10 000 Ft-ig — 400 Ft kedvezmény
20 000 Ft-ig — 500 Ft kedvezmény
20 000 Ft felett — 1000 Ft kedvezmény
Csoportok jelentkezése esetén további
kedvezményekről az irodában lehet tárgyalni

ÁPRILISI

60 Ft-os
vásárlási utalvány

Beváltható készpénzes
vásárlás esetén a 2C Áruházban.
Bp. XIII., Balzac u. 35.

Érvényes: 1992. április 30.

TKD-BUDAPEST

ÁPRILISI
400 Ft-os
vásárlási utalvány

Beváltható: TKD-BUDAPEST
1117 Budapest, Schönhercz Z. u. 21.
Egy szótárgép vásárlásához
egy utalvány használható fel!

Érvényes: 1992. április 30.

A NOVOTRADE SZERVÍZ Kft. az alább felsorolt szervízeiben
mindenféle szervízszolgáltatás munkadíjából 10% kedvezményt ad
az egyesületi tagoknak.

1053 Budapest, Magyar u. 12—14
1083 Budapest, Szigony u. 9.
1191 Budapest, Gábor Á. sétány 3.
3525 Miskolc, Fazekas u. 1—3.
4034 Debrecen, Holló L. u. 14.
5600 Békéscsaba, Bartók B. u. 37.
6724 Szeged, Csongrádi sugárút 76.
7624 Pécs, Jurisics M. u. 17.
8000 Székesfehérvár, Széchenyi u. 15/a.
9700 Szombathely, Szalonok u. 31.

Telefon: 117-3551
Telefon: 134-3153
Telefon: 127-4763
Telefon: 46-17-011
Telefon: 52-32-863
Telefon: 66-27-195
Telefon: 62-13-377
Telefon: 72-11-812
Telefon: 22-12-711
Telefon: 94-13-419

Felvevőhelyek:

9024 Győr, Babits M. 75.
6000 Kecskemét, Széchenyi tér 1—3.

Telefon: 76—23—720

Igazolás: a javítandó berendezés leadásakor egyesületi igazolvánnyal.
A kedvezmény többször is igénybe vehető.

Novotrade
SZERVÍZ Kft.

Az Országos Commodore Egyesület szolgáltatásai

Egyesületi tagoknak 20% kedvezmény:

	kiépitéstől függő
VC—20 memóriabővítés 3—27 kByte-os:	3500 Ft
C—16, C—116 memóriájának bővítése 64 kByte-ra:	1450 Ft
C—16 belső 16 kByte-os EPROM bővítés:	2900 Ft
C—16 belső 32 kByte-os EPROM bővítés:	2800 Ft
C—16 belső 8 kByte-os SOFT—ROM bővítés:	4000 Ft
C—16 belső 32 kByte-os SOFT—ROM bővítés:	2000 Ft
C—16 8 kByte-ról 32 kByte-ra átalakítás:	3200 Ft
C—16 és 1541 kompatibilis lemezegység párhuzamosítása:	5000 Ft
SOFTROM modul 32K, kikapcsoláskor sem felejt C-16, C-116, +4	
FÉK C—16, C—116, +4 potméteres sebességváltoztatás	2000 Ft
0%-tól 100%-ig fokozatmentesen	4300 Ft
TTL IC-teszter (Cartridge+lemezen a program)	
+4, C—16, C—116 UNI—ROM modul különféle kiépítésekben:	
— 8 kByte SOFT—ROM	3400 Ft
— 16 kByte SOFT—ROM	4000 Ft
— 8 kByte SOFT—ROM 16 kByte EPROM	4400 Ft
— 16 kByte SOFT—ROM 16 kByte EPROM	5000 Ft
— 16 kByte EPROM	2200 Ft

Kedvezmény nélkül:

Letilthatatlan RESET C—16, C—116, +4 minden programot megállít	3000 Ft
+4 és 1541 kompatibilis lemezegység párhuzamosítása	1450 Ft

Az EPROM-ba hozott programokat vagy a már kész menük valamelyikét építjük be (bekapcsoláskor és RESET-kor menüvel jelentkeznek, kikapcsoláskor nem törölődik). A SOFT—ROM tetszőleges EPROM menü futtatására alkalmas (RESET-kor menüvel jelentkeznek, kikapcsoláskor törölődik).

A párhuzamosított lemezegységhez jár egy lemezoldalnyi speciális program, melyek az új lehetőséget kihasználják (20-szoros gyorsító, 15 másodperces lemezoldal-másoló stb.).

A fenti bővítések megrendelhetők az OCE. irodájában a Pöttyögőszolgálat napjain 16—18 óra között. Részletes felvilágosítást az 1-363-951-es telefonszámon tudunk nyújtani (főleg este). Árainkat az alkatrészárak változásai befolyásolhatják.

MEGRENDELŐLAP

Megrendelek db 14 nyelvű elektronikus szótárt.

Kérem a készüléket az alábbi címre postai utánvétellel elküldeni:

NÉV:
 irányítószám (város)
 (járás, kerület) utca,
 házszám emelet, ajtó.

aláírás

A megrendelőlapot TKD-BUDAPEST címére kérjük elküldeni:
1117 Budapest, Schönhercz Zoltán u. 21. Tel.: 181-0352

Cserélhető lemezes winchester!



SQ-555 (meghajtó)

SQ-400 (lemez) 44 MB/lemez

SQ 5110 (meghajtó)

SQ-800 (lemez) 88 MB/lemez

Hivatalos magyarországi forgalmazó:

SZERVIZ Kft.

NOVOTRADE

Cím: 1053 Budapest, Henszlmann I. u. 9.
Telefon: 117-4144 Telefax: 117-9692

Meghatalmazott dealereink:

1. Microteam Kft., 1145 Budapest, Róna u. 127.
Tel./fax: 184-1226, 183-2625, 164-2779
2. Professzionál Kft., 1033 Budapest, Kaszásdűlő u. 5.
Tel./fax: 167-0289
3. Miskolci kirendeltség, 3525 Miskolc, Szabó L. u. 37.
Tel.: 46/56-079
4. Békéscsabai Kirendeltség, 5600 Békéscsaba, Andrassy u.
Tel.: 66/28-584
5. Korall Kft., 2800 Tatabánya, Március 15. út 3.
Tel.: 34/11-714
6. 3S Computer Kisszövetkezet, 6723 Szeged, Kemes u. 6.
Tel.: 62/26-277, Fax: 62/26-347
7. Volánelektronika Vercomp Kft., 9024 Győr, Dr. Petz Lajos u.
Tel./fax: 96/12-520
8. Novotrade Miskolc Kft., 3530 Miskolc, Vörösmarty u. 51.
Tel./fax: 46/49-489
9. Data-Elektronik Kft., 8200 Veszprém, Damjanich u. 7/A.
Tel./fax: 80/28-490
10. Elektrosoft Kft., 5000 Szolnok, József A. u. 6-8.
Tel.: 56/42-880, Fax: 56/44-222
11. Digitech Kft., 7101 Szekszárd, Rákóczi u. 6.
Tel.: 74/16-874
12. Számadó Kft., 6000 Kecskemét, Dózsa Gy. u. 29.
Tel.: 76/21-455, Fax: 76/21-462
13. BX-Next Kft., 3434 Mályi, Bercsényi út 50.
Tel.: 46/91-117
14. Palóccomp Kft., 3100 Salgótarján, Bartók B. u. 17.
Tel.: 32/14-114
15. Medorg Rt., 1132 Budapest, Váci út 48/e-f.
Tel./fax: 129-4281
16. Novotrade PC Kft., 1136 Budapest, Sallai u. 25.
Tel.: 149-0798 Fax: 131-0734
17. Almárium Kft., 1137 Budapest, Pozsonyi u. 21-23.
Tel.: 111-2830 Fax: 112-3647
18. Microchip Kft., 8000 Székesfehérvár, Élmunkás u. 47.
Tel.: 22/25-514
19. Trader Kft., 3300 Eger, Bródy S. u. 5.
Tel.: 36/25-006
20. Navigátor Kft., 4400 Nyíregyháza, Kórház út 26/B.
Tel./fax: 42/41-972
21. Alfadat Kft., 2803 Tatabánya, Tóth-Bucsoki út 12.
Tel.: 34/10-234, 10-405, Fax: 34/10-729
22. Interface Kft., 1116 Budapest, Hunyadi J. út 162.
Tel.: 166-5322/58, 55, Fax: 226-3793
23. Netrend Rt., 1089 Budapest, Elnök u. 1.
Tel.: 113-8217, 133-4760, 113-9537, Fax: 113-9537
24. OMNIS Kft., 2840 Oroszlány

input: **MAX** output: **maximum**

A **TUNGSRAM-MAX** mágneslemez japán és amerikai alapanyagból, amerikai technológiával, high-tech berendezéseken készül. Minden egyes mágneslemez hibamentességét a teljes felület számítógépes mérőrendszerrel történő tesztelése garantálja.

TUNGSRAM-MAX mágneslemezek

Standard	Formázott	Színes lemezek műanyag dobozban
5,25" TM 2S2D 53 Ft	5,25" TMF 2S2D 61 Ft	5,25" TMP 2S2D 63 Ft
5,25" TM 2SHD 77 Ft	5,25" TMF 2SHD 87 Ft	5,25" TMP 2SHD 88 Ft
3,5" TM 2S2D 88 Ft	3,5" TMF 2S2D 102 Ft	3,5" TMP 2S2D 99 Ft
3,5" TM 2SHD 155 Ft	3,5" TMF 2SHD 173 Ft	3,5" TMP 2SHD 167 Ft

Árainkhoz ÁFA-t számítunk!

- CSEREGARANCIA: 20 ÉV VAGY 20 MILLIÓ FORDULATI
- Mágneslemezek: no name és bulk csomagolásban is!
- Tárolódobozok, tisztítókészletek 3,5" és 5,25" méretben.
- Szoftvermásolás profi gépeken, írásvédő kivágás nélkül is!
- Viszonteladóknek 20% engedmény!
- Szoftverkészítőknek, nagyfelhasználóknak, diákoknak rendkívüli kedvezmények!
- Színes és formattált mágneslemezek, tárolódobozok, festékszalagok árusítása, szoftvermásolás és csomagolás, címkézés a szoftverkészítő igénye szerint.
- Kérje részletes árlistánkat!
- AKCIÓS ÁRAK az IFABO-n az A pavilonban, 1992. április 27–30.
- Tungstram Magnetic Media Budapest IV., Váci út 77. Tel.: 160-2233 Fax: 160-0925



TUNGSRAM-MAX[®]

KÉPVISELETEINK:

Agromark Kft. — Hódmezővásárhely,
Andrássy út 50. Tel.: 06-62-41695
Comtrade Kft. — Pécs,
Majorossy u. 36. Tel.: 06-72-26063
M és K Bt. — Kecskemét,
Bajcsy-Zsilinszky u. 5. Tel.: 06-76-21878
PGM Computer Kft. — Szeged,
Csongrádi sugárút 22. Tel.: 06-62-14380
Számker Kft. — Zalaegerszeg,
Rákóczi út 4–8. Tel.: 06-92-14500/144
Transzfer Kft. — Nyíregyháza,
Hősök tere 7. Tel.: 06-42-10481

HIVATALOS DEALEREINK:

Novotrade 2C Kft. — Budapest XIII.,
Balzac u. 35. Tel.: 140-2954
Radiant Kft. — Budapest XIV.,
Francia út 11. Tel.: 252-1999/266
Westing Iroda — 1149 Budapest,
Róna köz 12. Tel.: 163-7916

TUNGSRAM MAGNETIC MEDIA RT.
H-1340 Budapest, Váci út 77.
Tel.: 160-2233, Fax: 160-0925



A magyar olimpiai csapat arany fokozatú támogatója