

Az Országos Commodore Egyesület lapja

újság

1989/4

GEOS-TIPPEK

KÖNYVEK
ANIMÁCIÓ
OLVASÓBÓL-
SZERZŐ

TIPPEK — TRÜKKÖK
KEZDŐKNEK, HALADÓKNAK
C-16, PLUSZ 4, C-64
ÉS C-128-AS GÉPEKHEZ,
FLOPPYKHOZ



FIZESSEN ELŐ

A PC-SZALON KATALÓGUS- SÁRA!

A PC-piac aktuális hardver és szoftver információi havonta, 150 oldalon!
50 hazai gyártó, és forgalmazó termékválasztéka:

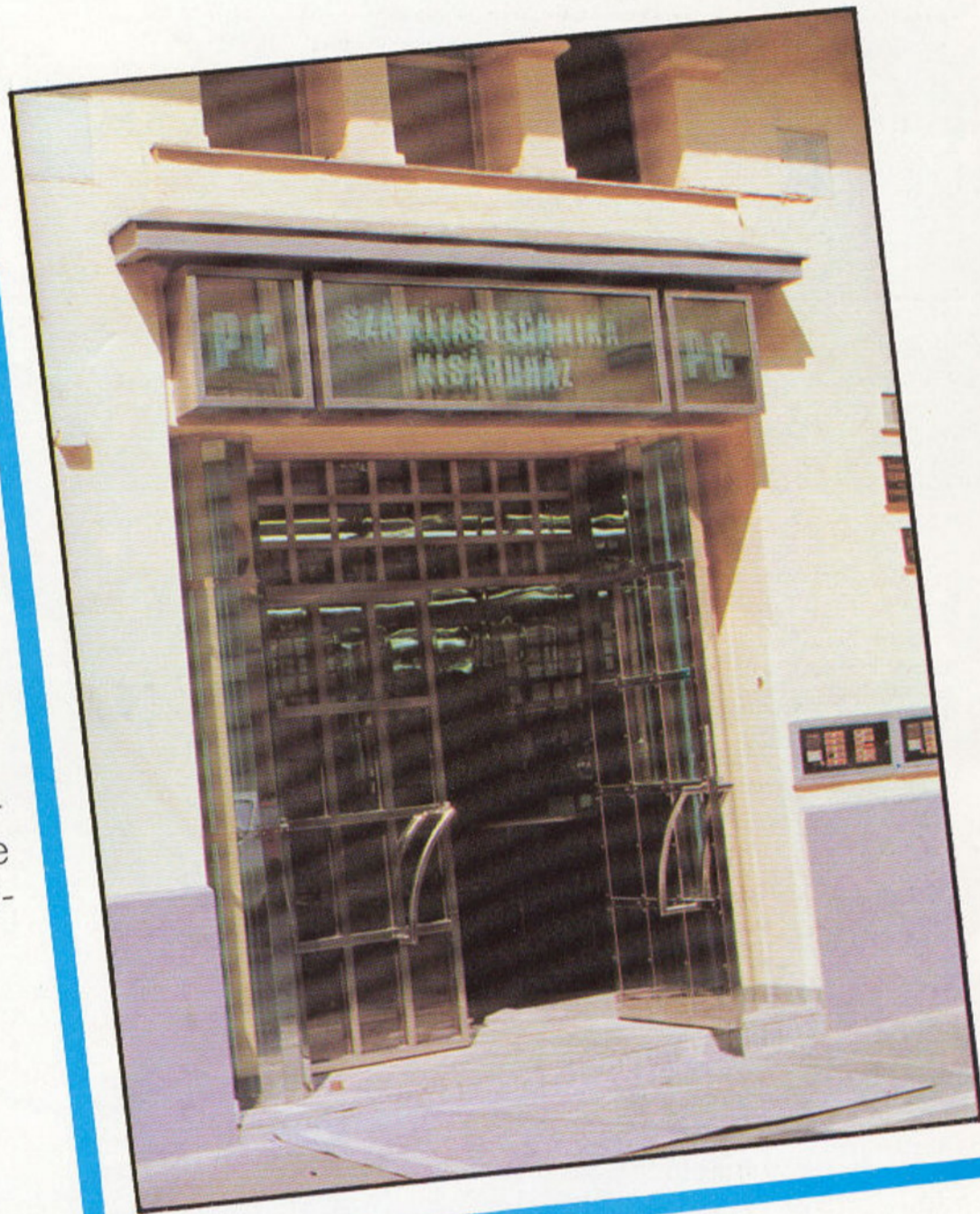
— floppy lemezek, festékszalagok, szakirodalom stb.— egy lehetőség, hogy döntése előtt fáradság nélkül, átfogóan tájékozódhasson!

A katalógust havonta jelentetjük meg!

Előfizetési díja: félévre: 4000,- Ft
egész évre: 6000,- Ft

Cím: Commodore Egyesület
1135 Bp., Kárpát u. 7/a. I. emelet

PC szalon



FESTÉKSZALAGOK
AMSTRAD, APPLE, BRO-
THER, CANON, CITIZEN,
COMMODORE, DATAPO-
INT, DEC, DIABOLO, EP-
SON, TACIT, FUJITSU, HP,
IBM, ITT, MANNESMANN,
NCR, NEC, NIXDORF, OKI,
OLIVETTI, PHILIPS, QUME,
RICOH, RITEMAN, SEI-
KOSHA, SIEMENS, STAR,
TEC, TRIUMPH-ADLER,
WANG stb.
NYOMTATÓKHOZ a PC-
szalonból!

1136 Budapest, Sallai Imre u. 6. Tel.: 310-776, 315-136

NOVOTRADE

MIT, HOGYAN?

EGYESÜLETI ÜGYEK

Az egyesületi tagnyilvántartással, tagdíjfizetéssel, postázással kapcsolatos ügyekben az egyesület irodájában kaphatnak tagjaink felvilágosítást. **1133 Budapest, Kárpát u. 7/a. I. em. 11. Tel.: 497-559**

Figyelem! Kérjük az 1989. évi tagdíjakat befizetni!

Köszönjük!

PÖTYÖGŐ SZOLGÁLAT

A megrendelés módja: a lapban megjelenő megrendelő cédulának vagy másolatának a kitöltésével lehet a megrendelést eljuttatni a C-újság szerkesztőségének címére. Ezzel egyidejűleg kell befizetni a kiszámított összeget az Egyesület számlájára. Amennyiben a megrendelő tud fénymásolatot csatolni a pénzfeladó vevényről, úgy ez elegendő ahhoz, hogy a megrendelés a teljesíthetők közé kerüljön. Amennyiben erre nincs módja, akkor meg kell várunk, míg az OTP-n keresztül megérkezik az Egyesülethez a befizetést igazoló szelvény. Ezután kerülhet csak sor a megrendelés teljesítésére. Jelen pillanatban a pénzbefizetés igazolásának megérkezésétől számított kb. két héten belül kerül sor a másolat elkészítésére és elküldésére.

Az Egyesület számlaszáma, címe: OTP Budapest XIII., Visegrádi u. 7/b. MNB 217-98292, OTP 565-3610.

Természetesen a pötyögő megrendeléseket a kifizetendő összeget leadhatják tagjaink személyesen is.

Pötyögőszolgálatunk a továbbiakban minden páros héten csütörtökön 16 és 19 óra között, a Kárpát utcai irodában várja az érdeklődőket. Természetesen megrendeléseket postán is fel lehet adni. Csekken történő befizetés esetén kérjük tüntessék fel az összeg rendeltetését.

A forintos kedvezmények a következő vidéki könyvesboltok 2C sarkaiban válthatók be.

PÉCS: Zrínyi Miklós Könyvesbolt. 7621 Jókai u. 25. Tel.: 72-12835

DEBRECEN: Szak- és ismeretterjesztő Könyvárúház. 4024 Hunyadi u. 8. Tel.: 52-23237

SZOMBATHELY: Savaria Könyvesbolt. 9700 Mártírok tere 1. Tel.: 94-12341

VESZPRÉM: Kölcsey Ferenc Könyvesbolt. 8200 Cserhát út 7.

BÉKÉSCSABA: Radnóti M. Könyvesbolt. 5600 Tanácsköztársaság út 2. Tel.: 25-207

GYŐR: Pattantyús Á. Géza Szakkönyvesbolt. 9021 Molnár Ferenc u. 9.

SZEGED: Tömörkény Könyvesbolt. 6720 Lenin krt. 48. Tel.: 62-21453

SZOLNOK: Szigligeti Könyvesbolt. 5000 Ságvári krt. 35. Tel.: 56-11133

MISKOLC: Chip-kuckó. 3530 Tanácsház tér 14.

Az 1989-es tagdíjfizetés

Az igények jobb kielégítése érdekében némi változás lesz, ezért kérjük figyelmesen olvassák el, melyik páholy milyen szolgáltatásokat nyújt. Bármelyik páholy tagságát is választja, PÖTYÖGŐSZOLGÁLATUNK, az apróhirdetési lehetőség és a szervizkedvezmény minden tagunk rendelkezésére áll. Az egyesületi tagok részére minden hónapban klubnap a Petőfi Csarnokban.

Vidéki tagjaink részére háromhavi vásárlási tikkett összegyűjtése esetén, postai megrendeléskor, díjmentes csomagküldő szolgálat a NOVOTRADE RT. 2C Áruházában.

DEÁKPÁHOLY: éves tagdíja 666 forint, félévre 366 forint
– a COMMODORE újság havonta megjelenő számai
– vásárlási kedvezmények

PLUSZPÁHOLY: tagdíja egész évre 1777 forint, félévre 911 forint, negyedévre 466 forint

– a COMMODORE újság havonta megjelenő számai
– havonta 120 forint vásárlási utalvány
– pénzkímélő akciók, vásárlási kedvezmények

SZUPERPÁHOLY: tagdíja egész évre 19 100 forint, féléves befizetés esetén 10 000, negyedévenkénti fizetés esetén 5000 forint

- 15 példány a havonta megjelenő COMMODORE újságból
- 1800 forint vásárlási utalvány minden hónapban
- vásárlási kedvezmények

A befizetés módja:

- személyesen az Egyesület irodájában
- átutalással az MNB 217-98292 OTP 565-3610-8 számlára
- megrendelés esetén számlát küldünk

Dr. Nagy Albert

OCE főtitkár

Tisztelt Egyesületi Vezetőség!

Sajnálattal közlöm Önökkel, hogy a Commodore Újságot kénytelen vagyok lemondani, mert hosszabb időre kiküldetésre megyek, és így egyelőre felesleges jártnom az Önök lapját.

Az Önök lapjáról jó véleményem van, mert legalábbis nekem nagyon tartalmas és elég jó információkat tartalmaz.

Bízom benne, hogy majd, ha hazajövök, az elmaradt lapokat visszamenőleg meg tudom vásárolni, és újból tag lehetek.

További sok sikert kívánok az Önök munkájához.

Tisztelettel:
Lovag Imre

Tisztelt Szerkesztőség!

Először is szeretném megköszönni a Commodore Újság 1989/1. számát. Én ugyanis egy ideje már nem vagyok tagja az egyesületnek. Ennek ellenére megkaptam az idei első számot és egy átutalási postautalványt a tagdíj befizetése érdekében. Szeretnék néhány dolgot megkérdezni, mielőtt ismét belépek az egyesületbe!

A Commodore újság 4. oldalán a következőt olvastam: „Vidéki tagjaink részére háromhavi vásárlási tikkett összegyűjtése esetén, postai megrendeléskor, díjmentes csomagküldő szolgálat a NOVOTRADE RT.2C Áruházában”. Ez azt jelenti, hogy a 3

Az Országos Commodore Egyesület módszertani kiadványa.

Egyesületi iroda és szerkesztőség: 1133 Budapest, Kárpát u. 7/a I. em. 11. Tel.: 497-559

Felelős kiadó: Horváth Judit, az egyesület elnöke

Főszerkesztő: Rados Péter

Felelős szerkesztő: Dr. Horváth András

Művészeti szerkesztő: Szulyovszky József

Egyesületi szervező: Winter Júlia

Lapmenedzser: Kovács Gábor

Levélcím: Commodore Újság, 1388 Bp. 62. Pf. 86.

Index: ISSN 0237-756 X

MSZH NYOMDA

tikettnek megfelelő összegért (3×60,— Ft) rendelhetek postán (illetve 3×60,— Ft kedvezményben részesülök drágább cikk esetén)?

Fel lehet-e adni hírdetést, a hírdetésfeladó négyzetháló (20. oldal) másolatával? Ez azért lenne fontos, mert így nem kellene az újságot szétvágni, ezáltal a másik oldalon levő információ nem veszne el az olvasó számára. Ha ez nem megvalósítható, akkor lehetne a másik oldalon olyan dolog, amely nem jelent fontos információt. Szerepelnek az újságban viszonylag nagyobb méretű címek (az egyes írások címei) és grafikák (rajzok, pl. 13. oldal: dobókockák), amelyek lehetnének a hírdetésfeladó hátoldalán is.

A 7. oldalon megjelent néhány játékról egy-két hasznos információ. A cikk végén lévő cím alapján egy olvasó írhatta. Kérdésem: ha egy elég érdekes és hasznos írást beküldenék, leköznék-e?

Végül — hogy valamit dícsérjek is — gratulálok az „Így lesz a Commodore 64 szuperzseni” című rovathoz. Nagyon hasznos információkat tartalmaz és valóban megkönnyíti egy esetleges hardveres kiegészítő kiválasztását.

Remélem kérdéseimre mihamarabb választ kapok!

Köszönettel:

Reiner Péter

Válaszunk mindhárom kérdésre: igen.

Szerkesztőség

Tisztelt Szerkesztőség!

Ezúton szeretnék választ adni az utóbbi számokban, Lugosi Antalné által felvetett C16-os és PLUS/4-es problémákra.

Először is az 1989/1. 3. oldal:

Az MCB-turbót vagy az eredeti HER TURBO-t használók tapasztalhatják, hogy PLUS/4, illetve bővített C16-os géppen BASIC-ból a SAVE „NAME”,7 utasítást használva az elmentett program neve 'szemét' lesz. Ennek oka a gép tárkezelésében keresendő. A felhasznált szöveges változók és konstansok értékét az interpreter a BASIC által használható memória (a \$37/\$38 által mutatott cím, ez C16-on \$3ff6, PLUS/4-en \$fd00) végén helyezi el. Az a tartomány PLUS/4 esetén átfedésben van a \$8000-tól kezdődő ROM-mal. Most tekintsük a turbót! (\$0d2f—\$0d4b).

1. LISTA

```
. 0D2F A0 0F LDY ##0F
. 0D31 A9 20 LDA ##20
. 0D33 99 C0 0E STA $0EC0,Y
. 0D36 88 DEY
. 0D37 10 FA BPL $0D33
. 0D39 A4 AB LDY $AB
. 0D3B C0 11 CPY ##11
. 0D3D 90 02 BCC $0D41
. 0D3F A0 10 LDY ##10
. 0D41 88 DEY
. 0D42 30 03 BMI $0D4C
. 0D44 B1 AF LDA ($AF),Y
. 0D46 99 C0 0E STA $0EC0,Y
. 0D49 4C 41 0D JMP $0D41
```

Magyarázat:

\$ab — filenév hossza,

\$af/\$b0 — filenévmutató,

\$0ec0 — ide másolja át a turbó a mentés előtt a filenevet.

Látható, hogy törlődik a RAM/ROM lapozással, ezért mindig a ROM-ot olvassa. Innen származnak a nem megfelelő karak-

terek. C16-on nem okoz problémát, mert a RAM nincs átfedésben a ROM-mal. Felvetődhet a kérdés, MONITOR-ban miért nem okoz ez problémát. Nos, a MONITOR nem használja a BASIC területet. A filenevet a \$025d—\$026c tartományra helyezi.

Javítsuk ki a hibát!

2. LISTA

```
. 0D2F A0 00 LDY ##00
. 0D31 C4 AB CPY $AB
. 0D33 B0 0F BCS $0D44
. 0D35 A9 AF LDA ##AF
. 0D37 20 94 04 JSR $0494
. 0D3A 99 C0 0E STA $0EC0,Y
. 0D3D C8 INY
. 0D3E C0 11 CPY ##11
. 0D40 D0 EF BNE $0D31
. 0D42 F0 04 BEQ $0D48
. 0D44 A9 20 LDA ##20
. 0D46 D0 F2 BNE $0D3A
. 0D48 EA NOP
. 0D49 EA NOP
. 0D4A EA NOP
. 0D4B EA NOP
```

Magyarázat:

\$0494 — RAM olvasás az A-regiszter tartalma alapján, itt \$af/\$b0 szerint.

A módosítással lehetővé válik 17 mentődő karakteres név megadása. CSAK BASICBEN!

SAVE „12345678901234567”,7

A módosítás után kihasználhatóvá válik az auto-run lehetőség is.

SAVE „NAME”,7,1

A kimentett program visszatöltése mindig a kimentés helyére történik. Ez az információ a turbósított programblokk első két byte-ja. Ezért hatástalan a leírt eljárás. Javaslat: használjon más típusú turbót, pl. COPY'88, ROMTURBO, stb.

Egy másik módosítás:

```
.0e98 20 84 ff jSr $ff84
helyére
```

```
.0e98 ea nop
```

```
.0e99 ea nop
```

```
.0e9a ea nop
```

Ezzel töltés után is megmarad a korábbi színbeállítás.

És még egy módosítás:

A töltés közbeni csíkozás más jellegűvé válik, ha

```
.0f70 69 10 adc # $10
```

helyére

```
.0f70 49 65 eor # $65
```

kerül.

Másik témakör — ékezetesítés

1988/12. 11—15. oldal.

Erősen terjengős cikk. Néhány megjegyzés:

6. pont

\$07f8 — a MONITOR memória olvasása a ROM/RAM-ra vonatkozik (\$00—\$7f:RAM, \$80—\$ff:ROM).

\$ff13 d0-as bitje ROM-státusz, azaz \$8000 felett a ROM vagy a RAM az aktív (0:RAM, 1:ROM).

Tehát, ha \$07f8 tartalma \$80, akkor a monitor az olvasás előtt lekapcsolja a ROM-ot, ezért a d0-es bit 0.

9. pont

Mit jelent az, hogy az 'IRQ rutin eltűnik'?

A \$0600—\$06ff memórialap reset-kor sem törlődik. Így könnyen aktivizálható újra.

MONITOR-ban:

```
>0312 09 06
```

A MONITOR S parancsa NEM menti el a végcímként megadott cím tartalmát, ezért a helyes parancs:

S„Karakterkészlet”,1,f000,f800

Jól használható ékezetesítésre az 'E Z A PLUS/4' demókazettán található 'MAGYAR ABC' program.

László József

Tisztelt Szerkesztőség!

Örömmel olvastam, hogy rajtam kívül mások is írtak az AMIGA számítógépről levelet. Öröömbe egy kis öröm is vegyült, mert úgy veszem észre, hogy lassan kialakul egy értelmetlen harc, amely a COMMODORE 64—128 tulajdonosok egy része és az AMIGA tulajdonosok között dúl. Azt hiszem ennek semmi értelme!

Az AMIGA 500-as tulajdonosok, nem szoktak KickStart-ot lemezzel betölteni, mert nincs rá semmi szükség, ugyanis a sokat emlegetett KickStart nem más, mint a 256 kbyte ROM. Ez tartalmazza a KickStart 1.2-es verziót. Ha valaki egy új, jobb verzióra akarná cserélni, akkor nem kell szerelőt hívni, nem kell hardver tanfolyamra járni, mert egyszerűen és gyorsan lehet a ROM KickStartot kicserélni. A rengeteg hibát tartalmazó 1.0 és 1.1 verziószámú KickStart már a múlté. Az 1.2, illetve az új 1.3, 1.4 nem véletlen van ROM-ban, ugyanis (szinte) teljesen hibamentes. Azért csak szinte teljesen, mert hibamentes operációs rendszert még nem láttam. Mellesleg a multitasking hibák csak a multitasking módban futtatott file-ok esetén jönnek elő (nem valami gyakran), az operációs rendszer nem ragaszkodik ahhoz, hogy minden programot multitasking módon futtassunk. Csak multitaskingban futó program meg elvéve akad...

Én, amikor C-ben, vagy ASSEMBLER-ben programozgatók az AMIGA-n, a WorkBench lemezt még a dobozából sem kell, hogy elővegyem. Nincs rá szükség. A WorkBench lemez nem más, mint felhasználói programok, karakterkészletek, segédprogramok, utilitigyűjteménye. Igaz, hogy egy csomó nyomtatóvezérlő file (meg más library is) található rajta, de ez azért jó, hogy bármilyen nyomtatója is legyen a felhasználónak, tudja majd használni a géphez. Ezek közül a sok segédfile közül csak egy 5—10 kbyte-ra van szükség. Így a BOOT-olás után (mondjuk épp a WorkBench lemezzel) a szabad memória nagysága: 393344 free memory. Ebben az esetben az 512 kbyte-ból már minden lefoglalandót lefoglalt a gép, még a nyomtató alapértelmezést is. (Munkaterület, képernyőmemória, rendszerváltozók, alaprutinok, rezidens programok (pl.: LOADWB), ilyenkor már mind be vannak állítva, lefoglalják a maguk területét a memóriából...)

Abban az esetben, ha a BASIC-et betöltjük, természetesen az is foglal helyet a memóriából. (Mivel interpreter, a program futása alatt a memóriában kell lennie). Nem is keveset, közel 150 kbyte-ot. De így is marad a BASIC programok számára vagy 200—250 kbyte hely!

A különféle C vagy PASCAL fordítók nem foglalnak helyet a memóriából, mert csak a fordítás idejére van szükség rájuk. A forrásszöveget egy egyszerű szövegszerkesztőben is megírhatjuk, azt a fordító a lemezzel részletekben tölti be és fordítja le OBJECT modul(ok)ra. Az object modul(ok)ból a LINKER készít futóképes file-t. Egy 200—300 kbyte nagyságú C vagy ASSEMBLER programot 512 kbyte-os alapgép még simán lefordít! Ennél nagyobb egybefüggő programot ritkán írnak C (esetleg assembler) nyelven!

Például: az AMIGA.COPY 1.2 a PROFIMAT ASSEMBLER segítségével egy menetben lefordítható. Pedig nem egy rövid assembler programról van szó! (A program forráslistáját a Data-Becker nagy floppy könyve közli papíron és a mellékelt lemezen is...!).

A vírusokról csak annyit, hogy egyre terjednek és nemcsak AMIGA gépeken, hanem C64(!), IBM, és más hasonló kategóriájú gépeknél is. Akinek ez újdonság, olvassa el a közelmúltban megjelent, vírusokkal kapcsolatos cikkeket, amelyek nem csak

a számítástechnikai újságokban láttak napvilágot. AMIGA gépekre tucatnyi VÍRUSKILLER és ANTIVÍRUS létezik már, és ahogy jelennek meg az új vírusok, úgy jelennek meg az új KILLEREK és ANTIVÍRUSOK. A helyzet nem olyan reménytelen, az írásvédett lemezre a legügyesebb vírus sem képes felülni, és egy jó VIRUSKILLER-rel és/vagy egy ANTIVIRUSSAL nyugodtan felvehetjük a harcot a legádázabb vírussal szemben is.

Az AMIGA gépen is vannak rejtett (talán egyesek szerint humoros) Copyright feliratok. Azért az AMIGA gépen is, mert más hasonló nagyságrendű gépeken is található. Ezek nem poloskák, ezek rejtett Copyright-ok. Mindenki kipróbálhatja: a bal oldali SHIFT + ALT + jobb SHIFT + ALT + a funkciógombok valamelyikét egyszerre megnyomjuk és nyomva tartjuk a képernyő felső sorában csakugyan megjelenik egy szöveg, de nem viccet tartalmaz, hanem a gép, a szoftver, a grafika készítőinek, tervezőinek nevét. Más kérdés, hogy ezt egyes nyugati újságok, mint viccet tüntették fel...

Akik csak játékokra akarnak vásárolni gépet, azoknak nem ajánlom az AMIGA 500-ast, mert ahhoz túl drága. Akik egyelőre csak játszani akarnak, de a jövőben szeretnének a komolyabb kategóriába tartozó gépekkel foglalkozni vagy esetleg a vállalatuknál van már IBM PC, szeretnék a munkát esetleg hazavinni, azoknak kifejezetten ajánlanám! Tehát nemcsak a profiknak, hanem minden olyan embernek, aki a jövő kihívását elfogadja, és a profibb kategóriától sem riad vissza. Mert akárhogy vesszük, egy kifejezetten játék célra készített gép nem (csak) a profiknak való...

És egy jó hír az érdekelteknek! Megtört a jég! Itt az első AMIGA 500-as program a COMMODORE újság olvasói számára.

A program AMIGA BASIC-ben íródott. Segítségével a magyar ékezetes karaktereket is ismerni fogja az AMIGA! Hogy minél gyorsabb legyen a program futása, az eredményt a RAM discen tárolja. Onnan nekünk kell lemezre menteni. Pl. a WorkBench lemezen az alábbi módosítások szükségesek:

Normál billentyűzet kiosztás:

```

Ű 1 2 3 4 5 6 7 8 9 0 í ' ú [törlés]
[Tab] q w e r t z u i o p ü ó [Enter]
[Ctrl] [CL] a s d f g h j k l ö á é [Ent.]
[Shift] < y x c v b n m , . ő [Shift]
    
```

SHIFT gombal :

```

Ű ! " ^ $ % & / ( ) = f ` ú [Törlés]
[Tab.] Q W E R T Z U I O P Ü Ó [Enter]
[Ctrl] [CL] A S D F G H J K L Ö Á É [Ent.]
[Shift] > Y X C V B N M ; : Ő [Shift]
    
```

Az alternatív (Alt) karakterkészlet :

```

` ' @ # ° ¼ ½ ¾ • « » ? = \ [Törlés]
[Tab] å ° © ® þ ð µ ï ø ¶ [ ! [Enter]
[Ctrl] [CL] æ ß ð . . . . . ð ; ' ^ [Ent.]
[Shift] < ± × ç ¢ ¢ - , , . _ [Shift]
    
```

Másoljuk a RAM-ból a DFO-ra a BASIC program futása után keletkezett „H” nevű file-t a COPY RAM:H TO DFO:DEVS/KEYMAPS utasítás segítségével. Ezek után elegendő az S alkönyvtárban található Startup-Sequence file-ban átírni a SETMAP D parancsot SETMAP H-ra. A H programocská nem egy önálló program, hanem egy adattáblázat, amit a SETMAP utasítás (program) kezel. Ha nem kapcsoljuk be a SETMAP-ot, akkor az alapértelmezésbeli USA billentyűzetkiosztás működik.

A különféle billentyűtérképek a DEVS/KEYMAPS alkönyvtárban találhatóak. (D=német, USA=amerikai, I=olasz, stb...) Ezentúl a H a magyar billentyű „térképet” jelenti majd. Minden olyan programban használhatjuk, ami a SETMAP utasítással együtt is üzemel. (A programok többségénél működik. Ez alól kivételt főleg a játékok képeznek!).

Persze az eredeti karakterek között a teljes magyar ékezetes karakterkészlet nem szerepel. Ezért a BASIC lemezen található

FED (Font Editor) segítségével mindenkinek magának kell megtervezni a hosszú ű, ú, ó, ő karaktert. A többi karakter nagyjából megfelel a magyarnak.

Az ékezetes karakterek ASCII kódjai:

Á = 193 (C1)	á = 225 (E1)
É = 201 (C9)	é = 233 (E9)
Ö = 214 (D6)	ö = 246 (F6)
Ő = 212 (D4)	ő = 244 (F4)
Ú = 218 (DA)	ú = 250 (FA)
Ü = 220 (DC)	ü = 252 (FC)
Ó = 211 (D3)	ó = 243 (F3)
Í = 205 (CD)	í = 237 (ED)
Ű = 207 (CF)	ű = 239 (EF)

Az új billentyűzetkiosztás és a BASIC program a mellékelt listán található.

Malmarits Tamás

Hungary:

```
REM *****
REM * Commodore ujsag 1989 *
REM *****
```

143

```
PRINT "Magyar karakterek "
PRINT "Kis türelmet, dolgozom !"
OPEN "RAM:H" FOR OUTPUT AS #1
h$="0123456789abcdei"
```

```
ide:
  READ a$: IF a$="V" GOTO Vege
  a%=(INSTR(h$,MID$(a$,1,1))-1)*16
  a%=a%+INSTR(h$,MID$(a$,2,1))-1
  b%=(b% AND 16383)+a%:PRINT#1,CHR$(a%);
GOTO ide
```

```
Vege:
  CLOSE 1: READ a%
  IF b%<>a% THEN
    PRINT "Adathiba! Ellenorizd ujra..."
  ELSE
    PRINT "OK!"
    PRINT "Masold at a RAM-bol a DEVS/KEYMAPS alkonyvtarba!"
    PRINT "Hasznalata : SETMAP H "
    REM Keszitette c Malmarits Tamas 1989.
  END IF
END
```

```
DATA 00,00,03,f3,00,00,00,00,00,00,00,01,00,00,00,00
DATA 00,00,00,00,00,00,01,47,00,00,03,e9,00,00,01,47
DATA 00,00,00,00,00,00,00,00,00,00,00,00,05,1a,00,00
DATA 00,4c,00,00,00,c4,00,00,00,2e,00,00,00,3d,00,00
DATA 00,8c,00,00,01,c4,00,00,00,36,00,00,00,45,00,00
DATA ff,07,ff,07,fe,00,00,00,00,00,00,00,00,ff,bf,ff
DATA ef,ff,ef,ff,f7,47,f4,ff,7f,00,00,00,07,03,03,03
DATA 03,03,07,03,03,03,03,07,23,07,80,00,07,07,27,07
DATA 07,07,27,27,27,07,07,07,80,00,00,00,27,07,07,27
DATA 27,27,27,27,07,03,03,07,80,00,00,00,01,27,07,07
DATA 07,07,27,07,03,03,07,80,00,00,00,00,22,00,41,00
DATA 04,02,00,80,80,80,00,80,41,41,41,41,41,41,41,41
DATA 41,41,41,41,41,41,05,05,00,00,00,40,80,80,80,80
DATA 80,80,80,80,80,80,80,80,80,80,80,80,80,80,80,80
DATA 80,80,80,80,7e,60,cf,ef,21,b9,21,31,b2,40,22,32
DATA 23,23,5e,33,a2,b0,24,34,25,bc,25,35,5e,bd,26,36
DATA 26,be,2f,37,2a,b7,28,38,28,ab,29,39,29,bb,3d,30
DATA 5f,3f,cd,ed,00,00,02,a4,7c,5c,da,fa,00,00,00,00
DATA 00,00,00,30,c5,e5,51,71,b0,b0,57,77,00,00,02,fc
DATA ae,ae,52,72,de,fe,54,74,a5,a4,5a,7a,00,00,03,30
DATA 00,00,03,64,00,00,03,98,b6,b6,50,70,7b,5b,dc,fc
DATA 7d,a6,d3,f3,00,00,00,00,00,00,00,31,00,00,00,32
DATA 00,00,00,33,00,00,03,cc,a7,df,53,73,d0,f0,44,64
DATA 00,00,02,ac,00,00,02,bc,00,00,02,cc,00,00,02,dc
```

DATA 00,00,02,ec,a3,a3,4c,6c,3a,3b,d6,f6,22,27,c1,e1
 DATA 5e,5e,c9,e9,00,00,00,00,00,00,00,00,34,00,00,00,35
 DATA 00,00,00,36,00,00,3e,3c,00,00,04,34,f7,d7,58,78
 DATA c7,e7,43,63,aa,aa,56,76,ba,ba,42,62,00,00,04,00
 DATA bf,b8,4d,6d,3c,2c,3b,2c,3e,2e,3a,2e,3f,5f,d4,f4
 DATA 00,00,00,00,00,00,00,2e,00,00,00,37,00,00,00,38
 DATA 00,00,00,39,00,00,04,68,00,00,00,08,00,00,04,7e
 DATA 00,00,00,0d,00,00,0a,0d,00,00,9b,1b,00,00,00,7f
 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,2d
 DATA 00,00,00,00,00,00,04,85,00,00,04,8d,00,00,04,95
 DATA 00,00,04,9e,00,00,04,a7,00,00,04,b2,00,00,04,bd
 DATA 00,00,04,c8,00,00,04,d3,00,00,04,de,00,00,04,e9
 DATA 00,00,04,f4,00,00,04,ff,00,00,05,0a,1b,1b,7b,5b
 DATA 1d,1d,7d,5d,00,00,00,2f,00,00,00,2a,00,00,00,2b
 DATA 00,00,05,15,00,00,00,00,00,00,00,00,00,00,00
 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
 DATA 00,00,00,00,08,61,08,62,00,3d,00,2b,00,66,00,46
 DATA 08,61,08,61,00,06,00,06,00,86,00,86,00,67,00,47
 DATA 08,62,08,62,00,07,00,07,00,87,00,87,00,68,00,48
 DATA 08,03,08,03,00,08,00,08,00,88,00,88,00,6a,00,4a
 DATA 08,04,08,04,00,0a,00,0a,00,8a,00,8a,00,6b,00,4b
 DATA 08,05,08,05,00,0b,00,0b,00,8b,00,8b,01,10,01,22
 DATA 00,a9,00,a9,00,05,00,05,00,85,00,85,65,e9,e8,ea
 DATA 65,eb,e9,e9,ea,e9,e9,e9,e8,ea,e8,e8,e8,e8,45,c9
 DATA c8,ca,45,cb,c9,c9,ca,c9,c9,c9,c8,ca,c8,c8,c8,c8
 DATA 01,10,01,22,00,b5,00,b5,00,15,00,15,00,95,00,95
 DATA 75,fa,f9,fb,75,fc,fa,fa,fb,fa,fa,fa,f9,fb,f9,f9
 DATA f9,f9,55,da,d9,db,55,dc,da,da,db,da,da,da,d9,db
 DATA d9,d9,d9,d9,01,10,01,22,00,a1,00,a5,00,09,00,09
 DATA 00,89,00,89,69,ed,ec,ee,69,ef,ed,ed,ee,ed,ed,ed
 DATA ec,ee,ec,ec,ec,ec,49,cd,cc,ce,49,cf,cd,cd,ce,cd
 DATA cd,cd,cc,ce,cc,cc,cc,cc,01,10,01,22,00,f8,00,d8
 DATA 00,0f,00,0f,00,8f,00,8f,6f,f3,f2,f4,f5,f6,f3,f3
 DATA f4,f3,f3,f3,f2,f4,f2,f2,f2,f2,4f,d3,d2,d4,d5,d6
 DATA d3,d3,d4,d3,d3,d3,d2,d4,d2,d2,d2,d2,01,10,01,22
 DATA 00,e6,00,c6,00,01,00,01,00,81,00,81,61,e1,e0,e2
 DATA e3,e4,e1,e1,e2,e1,e1,e1,e0,e2,e0,e0,e0,e0,41,c1
 DATA c0,c2,c3,c4,c1,c1,c2,c1,c1,c1,c0,c2,c0,c0,c0,c0
 DATA 01,10,01,22,00,ad,00,af,00,0e,00,0e,00,8e,00,8e
 DATA 6e,6e,6e,6e,f1,6e,6e,6e,6e,6e,6e,6e,6e,6e,6e,6e
 DATA 6e,6e,4e,4e,4e,4e,d1,4e,4e,4e,4e,4e,4e,4e,4e,4e
 DATA 4e,4e,4e,4e,01,10,01,22,00,b1,00,ac,00,19,00,19
 DATA 00,99,00,99,79,fd,79,79,79,79,ff,fd,fd,fd,fd,fd,fd
 DATA 79,79,79,79,79,79,59,dd,59,59,59,59,dd,dd,dd,dd
 DATA dd,dd,59,59,59,59,59,59,01,04,00,a0,20,b4,60,5e
 DATA 7e,a8,b4,5e,b4,b4,b4,b4,60,60,5e,60,60,60,01,04
 DATA 02,05,09,9b,5a,02,04,02,06,9b,41,9b,54,02,04,02
 DATA 06,9b,42,9b,53,02,04,03,06,9b,43,9b,20,40,02,04
 DATA 03,06,9b,44,9b,20,41,03,04,04,07,9b,30,7e,9b,31
 DATA 30,7e,03,04,04,07,9b,31,7e,9b,31,31,7e,03,04,04
 DATA 07,9b,32,7e,9b,31,32,7e,03,04,04,07,9b,33,7e,9b
 DATA 31,33,7e,03,04,04,07,9b,34,7e,9b,31,34,7e,03,04
 DATA 04,07,9b,35,7e,9b,31,35,7e,03,04,04,07,9b,36,7e
 DATA 9b,31,36,7e,03,04,04,07,9b,37,7e,9b,31,37,7e,03
 DATA 04,04,07,9b,38,7e,9b,31,38,7e,03,04,04,07,9b,39
 DATA 7e,9b,31,39,7e,03,02,9b,3f,7e,64,00,00,00,03,ec
 DATA 00,00,00,27,00,00,00,00,00,00,02,40,00,00,02,28
 DATA 00,00,02,24,00,00,02,20,00,00,02,1c,00,00,02,18
 DATA 00,00,02,14,00,00,02,10,00,00,02,0c,00,00,02,08
 DATA 00,00,01,f4,00,00,01,cc,00,00,01,c4,00,00,01,9c
 DATA 00,00,01,88,00,00,01,60,00,00,01,5c,00,00,01,58
 DATA 00,00,01,54,00,00,01,50,00,00,01,44,00,00,01,24
 DATA 00,00,01,20,00,00,01,1c,00,00,01,0c,00,00,00,f4
 DATA 00,00,00,2a,00,00,00,26,00,00,00,22,00,00,00,1e
 DATA 00,00,00,1a,00,00,00,16,00,00,00,12,00,00,00,0e
 DATA 00,00,00,0a,00,00,00,00,00,00,03,f2
 DATA V,7815

PROGRAMOK

Keret csíkozása

A program a rasztorsor-megszakítási lehetőségeket kihasználva a képernyő keretén jelenít meg színes csíkokat. Az effektus játékprogramokban, hosszabb várakozási időt kívánó tevékenységeknél használható jól. Az assembler forráslista lefordítása után (PROFI-ASS) a rutin SYS 49152-vel indítható.

```

1000 ;
1010 ;
1020 ; KERET CSIKOZAS
1030 ;
1040 ; SOMOS PETER, 1989.
1050 ;
1060 ;
1070 SYS 36864
1080 .OPT 00
1090 *= $C000
1100 IRQVECT = $0314 ; IRQ VEKTOR
1110 IRQRET = $EA31 ; IRQ BELEPESI CIME
1120 CIA1 = $DC0D ; CIA1 ICR
1130 VIC = $D000 ; VIDEOVEZERLO KEZDOCIME
1140 IRQCALL = $FEBC ; IRQ HIVAS
1150 ; +-+--+--+--+--+
1160 SEI
1170 LDA #Q1:STA IRQVECT ; IRQ UJ KEZDOCIMENEK
1180 LDA #>Q1:STA IRQVECT+1 ; BEALLITASA
1190 LDA #32:STA VIC+18:STA Q4+1 ; RASZTENSOR BEALLITASA
1200 LDA #27:STA Q4+2
1210 LDA VIC+17:AND #$7F:STA VIC+17
1220 LDA #$81:STA VIC+26 ; RASZTERMEGSZAKITAS BEALLITASA
1230 CLI
1240 RTS
1250 ; +-+--+--+--+--+
1260 Q1 LDA VIC+25:STA VIC+25:BMI Q2 ; IRQ-HIVAS ELLENORZESE
1270 LDA CIA1 ; NINCS
1280 CLI
1290 JMP IRQRET
1300 ; +-+--+--+--+--+
1310 Q2 LDA VIC+18:CMP Q4+1:BEQ Q3 ; RASZTENSOR ELLENORZESE
1320 LDA Q4+1:STA VIC+18
1330 JMP IRQCALL
1340 ; +-+--+--+--+--+
1350 Q3 LDA Q4:STA VIC+32 ; SZINVALTAS
1360 INC Q4 ; SZINKOD NOVELESE
1370 LDA Q4+2:BPL Q5 ; UJ RASZTENSOR KISZAMITASA
1380 LDA Q4+1:CMP #38:BCC Q5
1390 LDA #$1B:STA Q4+2:LDA #32:STA Q4+1
1400 Q5 CLC:LDA Q4+1:ADC #8:BCC Q6
1410 LDX #$9B:STX Q4+2
1420 Q6 STA Q4+1:STA VIC+18
1430 LDA Q4+2:STA VIC+17
1440 JMP IRQCALL ; VISSZA AZ EREDETI IRQ-HOZ
1450 Q4 .BYTE 0,0,0 ; ADATOK

```

READY.

Mozgó felirat a kereten

A rutin három sprite-ból (13, 14 és 15-ös blokk) egy feliratot rak össze, és a keret alsó részén azt mozgatja oda-vissza,

minden egyes irányváltatásnál módosítva a felirat színét. A mellékelt sprite adatok egy 'C Újság' feliratot eredményeznek. A fordítás (PROFI-ASS) után SYS 49152-vel indítsuk a rutint! Mivel a mozgatás a megszakítórutin segítségével történik, ezért ettől függetlenül a számítógép más tevékenységet is végezhet ezzel egyidőben.

SPRITE-ADATOK:

```

832 : 47 , 55 , 0 , 170 , 177 , 145 , 179 , 34 , 34 , 0 , 0 , 76
844 : 0 , 0 , 0 , 4 , 0 , 0 , 0 , 4 , 20 , 0 , 25 , 22
856 : 0 , 1 , 92 , 8 , 0 , 0 , 0 , 0 , 0 , 0 , 137 , 8
868 : 179 , 189 , 144 , 0 , 0 , 0 , 0 , 1 , 8 , 128 , 8 , 142
880 : 8 , 142 , 8 , 0 , 160 , 0 , 160 , 0 , 160 , 40 , 0 , 30
892 : 0 , 89 , 8 , 0 , 0 , 0 , 8 , 0 , 0 , 74 , 0 , 137
904 : 8 , 137 , 8 , 255 , 0 , 0 , 0 , 0 , 25 , 0 , 0 , 3
916 : 76 , 13 , 184 , 0 , 142 , 7 , 135 , 7 , 0 , 0 , 0 , 135
928 : 0 , 135 , 0 , 0 , 0 , 101 , 0 , 0 , 0 , 135 , 192 , 0
940 : 0 , 0 , 16 , 80 , 0 , 94 , 8 , 230 , 122 , 208 , 2 , 230
952 : 123 , 173 , 103 , 8 , 201 , 58 , 176 , 10 , 201 , 32 , 240 , 239
964 : 56 , 233 , 48 , 56 , 233 , 208 , 96 , 128 , 79 , 199 , 82 , 88
976 : 0 , 255 , 0 , 0 , 170 , 44 , 0 , 0 , 1 , 0 , 4 , 0
988 : 0 , 0 , 0 , 0 , 0 , 68 , 83 , 85 , 0 , 0 , 0 , 0
1000 : 0 , 0 , 0 , 0 , 0 , 0 , 255 , 3 , 0 , 0 , 60 , 3
1012 : 0 , 0 , 0 , 0 , 4 , 255 , 4 , 250 , 62 , 0 , 0 , 0

```

```

1000 SYS 36864
1010 .OPT 00
1011 ;
1012 ;
1013 ; MOZGO FELIRAT A KERETEN
1014 ;
1015 ;
1016 ; SOMOS PETER, 1989.
1017 ;
1018 ;
1020 *= $C000
1030 IRQVECT = $314 ; MEGSZAKITASVEKTOR
1040 IRQRET = $EA31 ; EREDETI IRQ KEZDOCIME
1050 SPRPTR = $7F8 ; SPRITE-MUTATO
1060 VIC = $D000 ; VIC KEZDOCIME
1070 SEI
1080 LDA #Q1:STA IRQVECT ; MEGSZAKITASVEKTOR ATIRASA
1090 LDA #>Q1:STA IRQVECT+1
1100 LDA #0:STA VIC+18 ; RASZTERMEGSZAKITAS BEALLITASA
1110 LDA #1B:STA VIC+17
1120 LDA #81:STA VIC+26
1130 LDA #7F:STA $DC0D
1140 LDA #50:STA VIC ; SPRITE-OK X KOORDINATAJA
1150 LDA #68:STA VIC+2
1160 LDA #80:STA VIC+4
1170 LDX #0:STX VIC+32:STX VIC+33 ; KERET- ES HATTERSZIN
1180 STX VIC+1:STX VIC+3:STX VIC+5 ; SPRITE-OK Y KOORDINATAJA
1190 STX Q100
1200 INX:STX VIC+39:STX VIC+40:STX VIC+41 ; SPRITE-OK SZINE
1210 LDX #13:STX SPRPTR ; SPRITE-OK KEZDOCIMEI
1220 INX:STX SPRPTR+1 ; (13,14,15-OS BLOKK)
1230 INX:STX SPRPTR+2
1240 CLI
1250 RTS
1260 Q1 LDA VIC+25:STA VIC+25 ; UJ MEGSZAKITASI RUTIN
1270 LDA VIC+18:CMP #F9:BNE Q2 ; RASZTERSOR ELLENORZESE
1280 LDA #32:LDY #13
1290 LDX VIC+33:STX #B0 ; HATTER- ES KERETSZIN
1300 LDX #0:STX VIC+33
1310 JMP Q3
1320 Q2 LDX #B0:STX VIC+33
1330 LDA #F9:LDY #1B ; UJ RASZTERSOR BEALLITASA
1340 Q3 STA VIC+18:STY VIC+17
1350 LDA #21111:ORA VIC+21:STA VIC+21 ; SPRITE-OK BEKAPCSOLASA
1360 LDA Q100:BEQ Q10 ; MOZGASIRANY ELLENORZESE
1370 DEC VIC:DEC VIC+2:DEC VIC+4 ; Y KOORDINATAK CSOKKENTESE
1380 LDA VIC:CMP #50:BNE Q11 ; POZICIO ELLENORZESE
1390 LDA #0:STA Q100:BEQ Q12
1400 Q10 INC VIC:INC VIC+2:INC VIC+4 ; Y KOORDINATAK NOVELESE
1410 LDA VIC+4:CMP #FF:BNE Q11 ; POZICIO ELLENORZESE
1420 LDA #1:STA Q100
1430 Q12 LDX Q100+1:INX:CPX #16:BNE Q13 ; SZINVALTAS
1440 LDX #1
1450 Q13 STX Q100+1:STX VIC+39:STX VIC+40
1460 STX VIC+41

```

145

```

○ 1470 Q11 CPY #13:BEQ Q4
1480 JMP IRQRET ; VISSZATERES A MEGSZAKITASBOL ○
1490 Q4 JMP $FEBC
1500 Q100 .BYTE 0,1

○ READY.

```

A 26. sor a képernyőn

A 26. sor nevű program a C64-es video chipjét ismét egy fokkal jobban kizsákmányolja. Amint azt a név szinte sugallja, a programocska a képernyőn létrehoz egy 26. sort. Az indítás az alábbi módon történjen:

```

LOAD „GENERATOR 26”,8,1
NEW
SYS 49152

```

```

○ 1 REM *****
○ 2 REM * C= UJSAG SORSZAM:146 * ○
3 REM * 26.SOR DEMO C64 *
4 REM * PROGRAM: KOVACS ZOLTAN *
○ 5 REM ***** ○
8 SYS 49152:REM 26 SOROS UZEM BEKAPCS.
10 A=17:T$="DEMO":SYS 49358,A,T$
○ 11 POKE 198,0:WAIT 198,1:REM VARAKOZAS EGY BILLENTYURE ○
20 SYS 49168:C=SQR(2*PI)+SIN(.865):SYS 49369,C ○
21 POKE 198,0:WAIT 198,1
30 SYS 49168:SYS 49369,"DEMO SZOVEG"
○ 31 POKE 198,0:WAIT 198,1 ○
40 SYS 49168:SYS 49358,10,"***** FELADAT *****"
41 POKE 198,0:WAIT 198,1
○ 50 POKE 1,53:PRINT"J":REM KEPERNYO + 26. SOR TORLESE ○
51 POKE 198,0:WAIT 198,1
52 FOR I=0 TO 50:POKE 1024+I,I:NEXT
○ 53 SYS 49369,"DEMO SZOVEG" ○
54 POKE 198,0:WAIT 198,1
60 POKE 1,55:PRINT"J":REM KEPERNYO TORLESE A 26. SOR NELKUL
○ 61 POKE 198,0:WAIT 198,1 ○
70 SYS 49168:REM A 26. SOR TORLESE ○
71 POKE 198,0:WAIT 198,1
80 SYS 49179:REM 26 SOROS UZEM KIKAPCS.

○ READY.

```

Ekkor a C64-es 26 soros üzemmódjába kerültünk. A felső és az alsó szél egy kicsit remeg, bizonyos körülmények között ezeken a kereteken fekete merőleges csíkokat látunk, de erről később még szólunk. A plusz sort a következőkben nulladik sornak fogjuk nevezni.

SYS 49152

Inicializálja a programot. A 0. sort ekkor töröljük. Ezt a parancsot mindig elsőként adjuk be a betöltés után.

SYS 49155

Ugyanaz mint a SYS 49152, de a 0. sor törlése nélkül.

SYS 49161

Ugyanaz mint a SYS 49155, de itt a BASIC ROM-ot és a Kernel ROM-ot nem másoljuk át az alattuk álló RAM-ba. Ekkor vgyázunk, mert így a nulladik sort csak korlátozottan használhatjuk (lásd ehhez a POKE 1,53-at is).

A SYS 49179 kikapcsolja a 26 soros üzemmódot. Erre akkor van szükség, ha lemezes műveleteket végzünk (pl. betöltés). Ha ugyanis bekapcsolt 26 soros üzemmódban próbálunk valamit betölteni, a számítógép menthetetlenül lemerevedik.

SYS 49168: törli a nulladik sort.

SYS 49369, sztring

A maximum 40 karakter hosszúságú sztringet a 0. oszloptól beírjuk a nulladik sorba.

SYS 49358, oszlop, sztring

A sztringet a megadott oszloptól kezdve írjuk a 0. sorba.

POKE 1,55

Ez az utasítás a 0. sor tartalmát „odaszögezi”, így ezt a képernyőről sem görgetéssel, sem törléssel nem tudjuk eltávolítani. Azt azután csak a nevezett SYS parancsokkal manipulálhatjuk.

POKE 1,53

A POKE 1,55-tel rögzített állapot feloldása. Vigyázat: Azt biztosítani kell, hogy a ROM alatti RAM érintetlen maradjon.

POKE 49504, szám

Ha a szám=128, akkor kikapcsoljuk a felső és az alsó keretet. Ha a paraméterek 0 és 15 közé esnek, akkor a felső és az alsó keret ezeknek a kódoknak megfelelő színre vált.

Azt tartuk szem előtt, hogy a 26 soros üzemmódban a programjaink kb. 50 százalékkal lassabban dolgoznak. Ezen kívül nem szabad megkísérelni aktiválni ezt az üzemmódot, ha az egyszer már aktív. Ebben az esetben a gépünk le fog merevedni.

Előfordulhat, hogy a felső és az alsó képernyőkereten fekete csíkok jelennek meg. A fő bűnös ebben a \$3FFF cella. Segítséget



egy POKE 4*4096—1,0 utasítás jelent, ami azonban egy BASIC programot fölülírhat. De ezen segíthet egy hosszabb REM sor, amely közvetlenül a \$4000-res címen kezdődik, így a POKE utasítás csak a kommentárt rombolja szét. Ha a program kisebb mint 14 kbyte, akkor a program vége mutatót (45/46) a \$4000-re kell állítanunk, hogy a változókat megvédjük.

Fontos tudni még azt is, hogy a program a sztringek kezelésekor nem ellenőrzi a hosszúságot. Azok a programozók, akik a saját fejlesztéseiket a „26. sor” mögé helyezik a tárolóba (a \$C328 után), akkor különösen ügyeljenek, hogy a SYS 49369-es utasítással feldolgozott sztring hossza ne legyen 40-nél nagyobb legyen. A SYS 49358,X, (sztring) utasításnál a sztring hossza értelemszerűen kisebb kell hogy legyen (40 minusz X)-nél.

A továbbiakban azt is tudnunk kell, hogy a rutin a sztringekben az <RVS ON> és az <RVS OFF> vezérlő kódokat NEM kezeli karakterként, de azokat végrehajtja. Például az alábbi alkalmazás gondolkodás nélkül elfogadható:

```
A$="0123456789 <RVS ON> 01234567890123456789 <RVS OFF> 0123456789"
```

Ezen kívül a <CRSR RIGHT> egy sztringben megtartja a funkcióját, azaz átugrik egy helyet a képernyőn (egy karakternek számít). A <CRSR LEFT>, a <CRSR UP> és a <CRSR DOWN> funkciókat nem hajtjuk végre, de a kódokat kijelezzük. Ha a sztringben megtalálható a CHR\$(13) vagy a CHR\$(141), akkor a kijelzést megszakítjuk a 0. sorban.

A PROGRAM MŰKÖDÉSE

A 26. sor megjelenítésének elve azon alapul, hogy a C64-es képernyőjét függőlegesen 7 bittel el lehet tolni. A fő rutin a rasztorosos interruptban „lóg”. A plusz sor láthatóvá tételéhez ki kell kapcsolni a felső és az alsó képernyőkeretet, de kívánságra létre lehet hozni egy mesterséges keretet. A SYS-szel való felhívás után inicializáljuk a rasztorosos interruptot. Az első ilyen a \$F9-es rasztercellánál aktiváljuk. A következőkben bemutatjuk, mi történik az egyes interruptoknál.

\$F9 sor:

Előkészítjük a keret kikapcsolásának feltételeit, és a következő interruptot a \$FF-nél rögzítjük.

\$FF sor:

Kikapcsoljuk a felső és az alsó keretet, a képernyőt 7 bittel feljebb toljuk. Az 1-es sor adatait (\$0400—\$0427) elrakjuk (a \$C2D0—\$C2F7 területre), és azokat az adatokat, amelyek a 0. sort reprezentálják, kiolvassuk a \$C300—\$C327 területéről beírjuk az 1. sorba. Ezután a program egy várakozóhurokba lép, s addig vár, amíg a képernyőn a rasztorsugár fönt a \$30-as sorba nem lép (emiat kell a kb. 50%-os lassubbodással számolni, mert az idő alatt BASIC program feldolgozás nem lehetséges).

\$30 sor:

Ez a rasztorsor a VIC számára fontos. Itt olvassa ki ugyanis az adatokat a képernyőtárolóba a következő nyolc rasztorsorhoz. Mihelyt az megtörtént, a képernyőt 7 bittel lejjebb toljuk, és az 1-es sor köztes tárolóban álló adatait (\$C2D0—C2F7) visszaírjuk az 1-es sor tárolójába, ahol eddig a VIC a nulladik (a plusz) sor adatait találta. Ez sajnos csak egy LDA—STA láncban történhet, mivel egy indikált hurokban ez túl sokáig tartana. A képernyő most 7 bittel lejjebb áll, és a VIC hirtelen azt tapasztalja, hogy ismét fel kell építenie egy képernyő sort a \$30-tól kezdve. Ezért hát ismét beolvassa a sor adatait a \$0400-as címtől. A trükk tehát abban áll, hogy a \$0400-tól kezdődő munkatárolót kétszer használjuk. A lényeg pedig az adatok elég gyors cseréje a megfelelő időpontban.

A program azon része, amelyik az interruptban dolgozik, a \$C162—\$C2C7 területen áll. Mivel a keretet lekapcsoltuk, a képernyőszínt kell felhasználni erre a „célra”, ha a \$C160-ban a 7. bit nincs bekapcsolva. Akkor a képernyőszínt a \$C161-ben egy időre letároljuk, és a \$C160-as cella értékét a \$D021-be visszük. Ezzel lehet azután felső és alsó keretet szimulálni. Az interruptba való be- és kikapcsoló rutinok, valamint a 0. sor adatait kezelő részek a \$C000—\$C15F területen állnak. Ezekben nincs semmiféle programozástechnikai különlegesség azon kívül, hogy a szövegek kiértékelésének és kijelzésének rutinjainál az ASCII-képernyőkód váltás „kézzel” kell, hogy történjen, mivel az operációs rendszer ilyen rutint nem bocsát a rendelkezésünkre.

Arra szigorúan ügyeltünk, hogy a \$C000 alatt ne foglaljunk el semmilyen tárolóterületet, sem a nulláslapon, sem máshol. Ezzel pedig biztosítjuk a kompatibilitást minden olyan BASIC programhoz, amelyek nem raknak adatokat a \$C000 területre.

```

1 REM *****
2 REM"* C= UJSAG          SORSZ. : 146  *
3 REM"*          A 26.  SOR  C64      *
4 REM"* PROGRAM:        KOVACS ZOLTAN *
5 REM *****
6 :
7 :
8 PRINT CHR$(147)"DATA BEOLVASAS ES ELLENORZES ...";:J=49152:VE=49863:P=J
9 FOR B=0 TO 7:READ A$
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 "███" P;: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"███KESZ":END
20 DATA 20,10,C0,20,1B,C0,20,3B,0582
21 DATA C0,20,7D,C0,20,A2,C0,60,1023
22 DATA A2,27,A9,20,9D,00,C3,CA,0956
23 DATA 10,FA,60,78,A9,00,A2,31,0862
24 DATA A0,EA,8D,1A,00,8E,14,03,0934
25 DATA 8C,15,03,A9,1B,A2,40,A0,0746
26 DATA 37,8D,11,D0,8E,05,DC,84,0920
27 DATA 01,58,60,78,A2,61,A9,9F,0892
28 DATA 8D,48,C0,8D,4B,C0,B9,00,0998
29 DATA 00,99,00,00,C8,D0,F7,EE,1046
30 DATA 48,C0,EE,4B,C0,CA,D0,EE,1417

```



○ 31 DATA 58,A9,35,85,01,A9,4C,8D,0830 ○
 32 DATA 44,E5,8D,EA,E8,A9,C5,A2,1432 ○
 ○ 33 DATA C0,8D,45,E5,8E,46,E5,A9,1241 ○
 34 DATA B4,A2,C0,8D,EB,E8,8E,EC,1520 ○
 35 DATA E8,20,0D,FD,60,78,A9,99,1276 ○
 ○ 36 DATA A2,C0,A0,81,8D,14,03,8E,0949 ○
 37 DATA 15,03,8C,1A,D0,A9,1B,A2,0756 ○
 38 DATA F9,8D,11,D0,8E,12,D0,58,1071 ○
 ○ 39 DATA 60,AD,19,D0,8D,19,D0,4C,0952 ○
 40 DATA 31,EA,78,A9,62,A2,C1,A0,1185 ○
 ○ 41 DATA 2F,8D,14,03,8E,15,03,8C,0517 ○
 42 DATA 05,DC,58,60,A2,27,8D,00,0799 ○
 43 DATA 04,9D,00,C3,CA,10,F7,A5,0986 ○
 ○ 44 DATA AC,48,4C,ED,E8,20,10,C0,1029 ○
 45 DATA AD,88,02,4C,47,E5,20,FD,0972 ○
 ○ 46 DATA AE,20,9E,B7,8A,E0,27,90,1092 ○
 47 DATA 02,A9,00,18,69,00,8D,37,0496 ○
 48 DATA C1,20,FD,AE,20,9E,AD,24,1051 ○
 ○ 49 DATA 0D,30,06,20,DD,8D,20,87,0676 ○
 50 DATA B4,20,A6,B6,AA,A0,00,E8,1122 ○
 51 DATA CA,F0,42,B1,22,30,3F,C9,1031 ○
 ○ 52 DATA 0D,D0,03,4C,3D,C1,C9,1D,0784 ○
 53 DATA F0,2F,C9,12,D0,0A,85,C7,1056 ○
 ○ 54 DATA E6,22,D0,E7,E6,23,D0,E3,1403 ○
 55 DATA C9,92,D0,04,A9,00,F0,EE,1206 ○
 56 DATA C9,20,90,0A,C9,60,90,04,0832 ○
 ○ 57 DATA 29,DF,D0,02,29,3F,E6,C7,1007 ○
 58 DATA C6,C7,F0,02,09,80,99,00,0929 ○
 ○ 59 DATA C3,C8,4C,F8,C0,60,29,7F,1175 ○
 60 DATA C9,7F,00,02,A9,5E,C9,12,1020 ○
 61 DATA F0,02,C9,20,90,05,09,40,0905 ○
 ○ 62 DATA 4C,2E,C1,C9,0D,00,F9,F0,1226 ○
 63 DATA E4,00,00,00,00,00,00,00,0220 ○
 ○ 64 DATA 0E,F6,AD,19,D0,8D,19,D0,1040 ○
 65 DATA 30,07,AD,0D,DC,58,4C,31,0674 ○
 66 DATA EA,AD,12,D0,C9,F9,D0,0D,1304 ○
 ○ 67 DATA A9,17,8D,11,D0,A9,FF,8D,1123 ○
 68 DATA 12,D0,4C,7E,EA,AD,60,C1,1124 ○
 69 DATA 30,0C,AD,21,D0,8D,61,C1,0905 ○
 ○ 70 DATA AD,60,C1,8D,21,D0,A9,F9,1262 ○
 71 DATA 8D,12,D0,A9,18,8D,11,D0,0926 ○

○ 72 DATA A2,27,8D,00,04,9D,D0,C2,0953 ○
 73 DATA 8D,00,C3,9D,00,04,CA,10,0763 ○
 ○ 74 DATA F1,AD,12,D0,C9,2E,D0,F9,1344 ○
 75 DATA AD,11,D0,29,80,D0,F2,AD,1190 ○
 76 DATA 60,C1,30,06,AD,61,C1,8D,0947 ○
 ○ 77 DATA 21,D0,A2,17,CA,D0,FD,A9,1258 ○
 78 DATA 1F,8D,11,D0,AD,D0,C2,8D,1113 ○
 79 DATA 00,04,AD,01,C2,8D,01,04,0726 ○
 ○ 80 DATA AD,D2,C2,8D,02,04,AD,03,1108 ○
 81 DATA C2,8D,03,04,AD,04,C2,8D,1062 ○
 ○ 82 DATA 04,04,AD,05,C2,8D,05,04,0738 ○
 83 DATA AD,D6,C2,8D,06,04,AD,07,1120 ○
 84 DATA C2,8D,07,04,AD,08,C2,8D,1070 ○
 ○ 85 DATA 08,04,AD,09,C2,8D,09,04,0750 ○
 86 DATA AD,DA,C2,8D,0A,04,AD,0B,1132 ○
 ○ 87 DATA C2,8D,0B,04,AD,DC,C2,8D,1078 ○
 88 DATA 0C,04,AD,DD,C2,8D,0D,04,0762 ○
 89 DATA AD,DE,C2,8D,0E,04,AD,DF,1144 ○
 ○ 90 DATA C2,8D,0F,04,AD,E0,C2,8D,1086 ○
 91 DATA 10,04,AD,E1,C2,8D,11,04,0774 ○
 92 DATA AD,E2,C2,8D,12,04,AD,E3,1156 ○
 ○ 93 DATA C2,8D,13,04,AD,E4,C2,8D,1094 ○
 94 DATA 14,04,AD,E5,C2,8D,15,04,0786 ○
 ○ 95 DATA AD,E6,C2,8D,16,04,AD,E7,1168 ○
 96 DATA C2,8D,17,04,AD,E8,C2,8D,1102 ○
 97 DATA 18,04,AD,E9,C2,8D,19,04,0798 ○
 ○ 98 DATA AD,EA,C2,8D,1A,04,AD,EB,1180 ○
 99 DATA C2,8D,1B,04,AD,EC,C2,8D,1110 ○
 ○ 100 DATA 1C,04,AD,ED,C2,8D,1D,04,0810 ○
 101 DATA AD,EE,C2,8D,1E,04,AD,EF,1192 ○
 102 DATA C2,8D,1F,04,AD,F0,C2,8D,1118 ○
 ○ 103 DATA 20,04,AD,F1,C2,8D,21,04,0822 ○
 104 DATA AD,F2,C2,8D,22,04,AD,F3,1204 ○
 ○ 105 DATA C2,8D,23,04,AD,F4,C2,8D,1126 ○
 106 DATA 24,04,AD,F5,C2,8D,25,04,0834 ○
 107 DATA AD,F6,C2,8D,26,04,AD,F7,1216 ○
 ○ 108 DATA C2,8D,27,04,4C,7E,EA,00,0814 ○

○ READY.

PÖTYÖGŐ
AKCIÓ

AKCIÓ!

OLCSÓBB
PROGRAMOK

**A FLOPPY
ÖNNEK
INGYEN
VAN!**

8 db program VC20-ra	220,—
12 db program Plus/4-re	320,—
16 db program C128-ra	420,—
20 db program C16-ra	520,—
24 db program C64-re	620,—
32 db program C64-re	820,—

— Megrendelhető
utánvétellel!
— Megvásárolható
az egyesület
irodájában kedden
és csütörtökön!

GEOS

TIPPEK KEZDŐKNEK

Kérdések és válaszok

Ebben a rovatban segítünk a GEOS problémák megoldásában. Szeretnénk megmutatni, hogyan használjuk ezt a rendszert még hatékonyabban.

Van-e arra lehetőség, hogy a GEOS-t egy EPROM-ba égessem, abból a célból, hogy az a bekapcsolás után azonnal a rendelkezésemre álljon?

Mivel a GEOS a maga rendkívül komplex lemezes hivatkozásaiból „él”, ezért semmilyen mód nincsen a floppy nélküli munkára. Mivel ezen kívül ebben a rendszerben a tárolóterületeket állandóan megváltoztatjuk, ezért a GEOS-t képtelenség EPROM-ba égetve használni.

Nálam problémák jelentkeznek a Commodore 1351-es proporcionális egerének használatakor. Mialatt a joystickkel hiba nélkül tudok dolgozni, a kurzort (a nyilat) az egérel nem tudom mozgatni

Sajnos azt nem írta meg, melyik GEOS változattal dolgozik. A GEOS 1.2-vel ugyanis egy proporcionális egeret (pl. a 1351-eset) nem lehet teljesen kihasználni, mivel erre a célra nincs meghajtóprogram. Ugyanakkor ez az egér magától a joystick üzemmódba vált, ha a C64-es bekapcsolásakor megnyomjuk a jobb oldali egérgombot.

A GEOS 1.3-mal a <CBM I> billentyűkombinációval kell beállítani az egérmeghajtót. Ez egyszerűen egy ablakban történik. Hogy ez mindjárt a boot után a rendelkezésre álljon, a megfelelő ikont a Desktopban a joystick meghajtó elé kell vinni.

A GEOS meghajtólemezén található a Koala Pad és a Lightpen állomány. Mi célra szolgálnak ezek?

Ezek esetében a GEOS beviteli meghajtó programról van szó. A Koala Painter egy ma már nem forgalmazott, de igen népszerű festőprogram. Ennek a programnak a beviteli eszköze az a grafikus tábla, amely kezelését az az állomány teszi lehetővé. A Lightpen állomány egy fényceruzát kezel beviteli eszközként.

Szeretném a 1541-es floppymat felgyorsítani a Prologic DOS rendszer segítségével. Ez a rendszer együtt tud működni a GEOS-szal és a 1750-es RAM bővítővel?

Nehézségekre ebben az esetben nem kell számítani. Ugyanakkor a floppygyorsító a GEOS-ra semmilyen hatással nincsen. A GEOS ugyanis a lemezegység kezelésére saját rutinokat használ. A speedert tehát már a GEOS betöltésekor lekapcsoljuk, így annak semmilyen hatása nem lesz. A RAM bővítő használatakor sem jelentkezhetnek problémák.

HA ÉRVÉNYESÜLNI KÍVÁN, TANULJON MEG LEGALÁBB KÉT IDEGEN NYELVET!

A SYSTEM GEORGE NYELVOKTATÓ PROGRAMOK SEGÍTSÉGÉVEL GYORSABB, ALAPOSABB A TANULÁS!

C64 gépre ANGOL, NÉMET, OROSZ, SPANYOL, PC/XT, AT gépekre ANGOL és NÉMET programok állnak rendelkezésünkre.

Az ANGOL, NÉMET és OROSZ programokhoz együtt futtatható hanganyag is kapható!

A vállalatoknak érdeke, hogy dolgozóik nyelvismerettel rendelkezzenek. Tegyük tehát lehetővé számukra, hogy a mikroszámítógépek szabadidejében tanuljanak!

A PROGRAMOK ALKALMASAK AKÁR MAGÁNOKTATÁSRA ÉS KEZDŐK, VALAMINT HALADÓK SZÁMÁRA IS!

KÉRJENEK RÉSZLETES ISMERTETŐT ÉS ÁRJEGYZÉKET!

INFORMÁCIÓ, RENDELÉS: Országos Commodore Egyesület
1133 Budapest
Kárpát u. 7/a. I/11.
Telefon: 497-559

TANFOLYAMOKAT IS RENDEZÜNK!!!

SYSTEM GEORGE

TIPPEK TRÜKKÖK

kezdőknek

Biztonságos kommentárok

Aki nekilát BASIC programok írásának, fogadja meg egy gyakorlott programozó jótanácsát, és lássa el BASIC programját elegendő kommentárral. A BASIC erre a célra a REM utasítást használja. Minden, ami ezután a parancs után áll, hidegen hagyja a számítógépet. A jól választott kommentárok (REMARKS) ugyanakkor hetekkel később nagy segítséget adhatnak az elfeledett programfunkció felidézésére. Sajnos azonban ezen a téren van egy interpreter (ez a program a gépben, amely a BASIC utasításokat „értelmezi”) hiba. Adjuk be a kisbetű/nagybetű karakterkészlettel az alábbi BASIC sort:

```
10 rem Halló Fiúk!
```

A dolog rendben megy. Legalábbis a következő LIST utasítás kiadásáig! Próbáljuk csak ki, azaz adjuk be a LIST parancsot (és persze, mint minden más esetben nyomjuk meg a (RETURN) gombot). A képernyőn ekkor az alábbi sor jelenik meg:

```
10 rem left$allo asciuk!
```

Persze ez nekünk egyáltalán nem állt szándékunkban. Az interpreter ilyenkor a nagybetűket (vagy a másik betűkészletben a grafikus jeleket) BASIC parancsként „értelmezi”, azaz a H-ből left\$, az F-ből asc lett. Persze van egyszerű megoldás: Ha mindenképpen ilyen kommentárokat akarunk írni, rakjuk a REM után egy idézőjelet:

```
10 rem "Halló Fiúk!"
```

A sor végén „szokásos” második idézőjelet nem is kell kitenni. LIST esetén a sor az eredetinek megfelelően jelenik meg.

Basic program helyreállítása

Milyen gyakran történik meg az emberrel az, hogy óvatlanul beadja a NEW utasítást, majd megnyomja a (RETURN)-t. Hosszú órák munkája válhat látszólag köddé egy szempillantás alatt, ha a programot buta módon nem tároltuk be. Ugyanilyen hatást vált ki egy esetleges reset gomb megnyomása: A C64-es a bekapcsolási felirattal jelentkezik vissza.

De álljon meg a menet! A programunk nem vész el. Szokjunk hozzá ahhoz a gondolathoz, hogy a számítástechnikában csak igen ritkán történik fizikai törlés is. Gondoljunk csak bele: A NEW után hamar jön a READY jelzés. Lenne ideje a gépnek valóban „kitörölni” esetleg több ezer byte-ot? Ezek az adatok pedig ott állnak a tárolóban, csak tudni kell előcsalogatni őket. A legfontosabb az, ne-hogy beadjunk egy újabb BASIC sort, de változókat sem szabad definiálni (pl. A=35).

```
POKE 2050,8: SYS 42291
```

```
POKE 46, PEEK(35)—(PEEK(781)>253)
```

```
POKE 45, PEEK(781)+2 AND 255:CLR
```

Ezt, vagy hasonló parancskombinációkat sokszor megtaláljuk a BASIC bővítésekben, ahol azokat RENEW vagy OLD névvel illetik, mivel ezek „visszacsinálják” a NEW utasítást.

Van még itt egy szabad byte?

Amikor a számítógépes újonc programozni kezd, s hosszú gépelés után meg szeretné tudni mennyi BASIC byte szabad még, beadja a PRINT FRE(0) utasítást. Mire az eredmény ekkor esetleg -26627 lesz!!!

Nos, ismét csak egy interpreter hiba van a dologban, amit azonban nem nehéz kijavítani. A helyes értéket az alábbi módon kapjuk meg:

```
PRINT FRE(0)+216
```

A PRINT helyett használhatjuk a „?”-et, úgy gyorsabban megy.

Segítség az Action Cartridge plushoz

Az embert egy idő után bosszantja, hogy az Action Cartridge a reset gomb minden egyes megnyomása után a modul saját reset menüjébe ugrik, ahol kiválaszthatom, milyen resetet is akarok egyáltalán. Véletlenül találtam két használható billentyűkombinációt, amelyek segíthetnek.

A C= (Commodore, balra lent) és a reset gomb egyszerre történő megnyomásakor automatikusan az „Exit to Fastload” menübe kerülünk. Hasonlóképpen viselkedik a (CTRL)+reset is. Ebben az esetben a modult kikapcsoló reset hajtódik végre („Normal Reset” menü).

Üzenetek lemezen

Ha szeretnénk egy rövid üzenetet írni a lemezre (pár betűből álló kommentárt egy programhoz, vagy a lemezhez), nincs szükség mindjárt szövegszerkesztőre vagy diszkmonitorra. Egyszerűbb a dolgunk, ha az alábbiakat tesszük: SAVE „az üzenet szövege”, 8. Ekkor „az üzenet szövege”

bekerül a lemez tartalomjegyzékébe. A szöveg maximum 16 karakter lehet, pl. DON'T VALIDATE (a lemezt validálni tilos), vagy JÁTÉKPROGRAMOK, stb. Ha nincs BASIC program a tárolóban, akkor a „szöveg” csak 1 blokkot fog elvenni a szabad 664-ből. Ha ezt valaki be akarja tölteni a LOAD és LIST után csak a READY köszön vissza. Ez a trükk minden gépnél alkalmazható, azaz a C128-nál és a PLUS/4-esnél is.

PEEK, POKE és SYS

Egy rövid paranccsal elérhető az, amit különben csak nehezen tudnánk megvalósítani. Például:

POKE 19,64: Az INPUT esetén nem jelenik meg a kérdőjel. A parancsot feltétlenül az adott INPUT előtt kell kiadni, majd ezután azt a POKE 19,0-val hatástalanítani kell.

POKE 22,35: Listázásnál nem jelennek meg a sorszámok.

POKE 204,0: A kurzort bekapcsolhatjuk futó programból vagy a GET alatt is.

POKE 207,0: POKE 204,1: A kurzort ismét kikapcsoljuk.

PRINT PEEK(214): Kiadja a kurzor aktuális sorpozícióját. A kurzort abba a sorba (0-24) és oszlopba (0-39) visszük, amelyiket paraméterként megadtuk.

PRINT PEEK(646): Kiadja a pillanatnyi karakterszint (0-16).

POKE 650,128: Minden billentyű ismételt (repeat).

POKE 650,64: Egyetlen billentyű sem ismételt.

POKE 650,0: Csak az <INST/DEL>, a <SZÓKÖZ> és a kurzorbillentyűk ismételtnek.

POKE 775,200: LIST elleni védelem bekapcsolás.

POKE 775,167: LIST elleni védelem kikapcsolás.

POKE 788,52: Kikapcsolja a <RUN/STOP> gombot.

POKE 788,49: Bekapcsolja a <RUN/STOP> gombot.

POKE 792,193: A <RUN/STOP>+<RESTORE> (de nem a <RUN/STOP> kikapcsolása).

POKE 808,225: A <RUN/STOP>+<RESTORE> letiltás, LIST védelem.

A <RUN/STOP> gomb funkciójának letiltásakor egyben logikusan <RUN/STOP>+<RESTORE> is blokkolva van.

SYS 64738: A C64-es a bekapcsolási képpel jelentkezik vissza.

SYS 64767: Rövidebb és gyorsabb reset.

Utóbbi esetben a képernyő alapszínét nem állítjuk vissza. Számos BASIC bővítést csakis ezzel az utasítással lehet kikapcsolni, mert egy SYS 64738 vagy egy hardveres reset során végrehajtódik a modul teszt (van-e modul a bővítőportban?), amit kihasználva a bővítések (pl. Simon's BASIC) bekapcsolódnak.

Itt a vége!

Miért is nem adja ki a C64-es az editálható sor végén, vagy a vége előtt egy figyelmeztető hangjelzést? Ez csak egy kis segédprogram alkalmazásával oldható meg, amelyet a mellékelt listában (Warn-

ton néven) közreadunk. A tárolás után a betöltés a LOAD „WARNTON”, 8-cal történhet, az indítás pedig RUN-nal. Néhány másodperces várakozás után a hangjelzés használható lesz. Ekkor minden BASIC sor 70. és 75. karakterének beadása után figyelmeztető hangjelzést kapunk. A SYS 49256 lekapcsolja a hangot, a SYS

49156 pedig ismét be. A POKE 49194,s1 és a POKE 49198,s2 utasításokkal adhatjuk meg a hang megszólalásának s1 és s2-es pozícióját. Az s1 és az s2 értéke 40 és 79 közé eshet. A POKE 48212,a és a POKE 49225,b a hangmagasságot változtatja, itt 1 és 255 közé eső értékeket lehet megadni.

Ügyeljünk arra, hogy a képernyőn egészen jobbra a 39. pozícióban feltétlenül állnia kell egy karakternek, legyen az akár a szóköz is, különben a sort nem tudjuk nyomon követni. Nem szabad tehát a kurzorral a második 40. képernyői sorba lépni, ha hallani akarjuk a sipolást.

```

1 REM *****
2 REM *           C= UJSAIG           *
3 REM *           WARNTON           *
4 REM *           C64               *
5 REM *****
10 A=0
15 READ B:IF B=-1 THEN SYS 49152:NEW
20 POKE 49152+A,B:A=A+1:GOTO 15
25 DATA 120,162,21,160,192,142,20,3,140,21,3,88,32,163,253,169,15,141,24,212
30 DATA 96,8,72,165,213,201,79,240,10,169,32,141,4,212,104,40,76,49,234,165
35 DATA 211,201,69,240,14,201,74,240,23,169,32,141,4,212,104,40,76,49,234,169
40 DATA 35,141,1,212,32,85,192,104,40,76,49,234,169,55,141,1,212,32,85,192
45 DATA 104,40,76,49,234,169,0,141,0,212,141,5,212,169,250,141,6,212,169,33
50 DATA 141,4,212,96,120,162,49,160,234,142,20,3,140,21,3,162,0,142,4,212,142
55 DATA 24,212,88,32,163,253,96,-1

```

READY.

INPUT komforttal

Segítség a C16(PLUS)4 és C128-ashoz

Felejtjük el az INPUT parancsunk igencsak sajátos tulajdonságait! Meg fogunk lepődni, milyen komfortot nyújt cserébe az Optinput program.

Az INPUT parancs egy sor kellemetlen tulajdonsággal rendelkezik, és mindegy, hogy a BASIC 2.0, 3.5 vagy 7.0-tól van szó. Így többek között a bevitelnél mindenféle adat lehetséges, amely főleg a nem profik számára okozhat meglepetéseket. Például egyáltalán nincs értelme a képernyő törlésének (mely ugye lehetséges) a beviteleknél. Azután az se jó, hogy a számítógépünknek alapjában véve mindegy, hogy milyen hosszú is a bevétel. Először a <RETURN> megnyomásánál jelentkezik hiba, ha a lehetséges hosszúságot túlléptük. Na, és a tilos karakterek, mint a „:” vagy a „;”, s a nem éppen sokatmondó és hibameghatározó angol hibajelzések (pl. ?REDO FROM START) úgyszintén hozzájárulnak ahhoz, hogy a felhasználó a haját tépje.

Ezért mi sem áll hozzáink közelebb, minthogy egy olyan INPUT rutint használjunk, amely a nevezett hiányosságokat mind kiküszöböli. S az Optinput egy ilyen rutin, amelyet ráadásul, mint BASIC alprogramot használhatunk, amelyet természetesen bárki illeszthet a saját igényeihez. Ez pedig egy nem lebecsülendő előny.

Adjuk be az 1-es listát. De vigyázzunk. Az Optinput egy alprogram, tehát önmagában nem használható. Ga azt RUN-nal elindítanánk, kis idő múlva ILLEGAL QUANTITY ERROR hibajelzést kapunk. Az alprogramot a saját BASIC programunkban GOSUB 8000-rel kell felhívni. A kísérletezésekhez kiválóan alkalmas a mi demo programunk (2. lista). Írjuk az 1100–1155-ös sorokat egyszerűen az alprogram (Optinput) elé. A demo az alprogramot többször is felhívja. Az eredményt minden esetben az INP\$-ban találjuk. Előtte azonban az alábbi paramétereknek megfelelő jelentést kell adnunk:

Változó	Jelentés
LG	A bevétel hossza.
ZS	A jelkészlet (1=alfanumerikus).
VG\$	Előre megajánlott bevétel (ha nincs VG\$="").
TAB	Tabulátor (rendesen 8).

Elképzelhető például az alábbi felhívás:

```
100 LG = 15:VG$ = „PROBA”:ZS = 1:GOSUB 8000:NEV$ = INP$
```

Ekkor a bevétel hossz maximum 15 karakter lehet, a bevételnél ajánlott „értékként” megjelenik a PROBA, és csak az alfanumerikus jelkészlet engedélyezett. Végezetül pedig az eredmény a NEV\$-be kerül.

Maga a kezelés igen egyszerű: Egy nem villogó kurzor jelzi a beviteli pozíciót. Itt lehet beadni minden engedélyezett karaktert. Ha a ZS értéke 1, akkor az alfanumerikus karaktereket (betűk, számok) használhatjuk, illetve a legtöbb különleges jelet. (Pl. idézőjel, vessző, pontosvessző is!). Ha más értéket állítunk be, akkor csak a számok, a + és a -, illetve az E (a kitevő jelzése) beadása lehetséges. Mindezek mellett az editálás is sokkal komfortosabb lesz, a normál INPUT paranccsal szemben:

```

1000 :
1005 REM *****
1010 REM * --- OPTINPUT --- *
1015 REM * ===== *
1020 REM * DEMO 147 *
1025 REM *****
1030 :
1100 VOL 8:KEY 1,CHR$(133):KEY 2,CHR$(134):SCNCLR:PRINT CHR$(9)CHR$(14)CHR$(8)
1105 PRINT " *** DEMOPROGRAMM ***":PRINT:PRINT
1110 PRINT "EREM A KOVETKEZO ADATOKAT BEIRNI :":PRINT
1115 PRINT "XEZETEKNEV: ";LG=15:ZS=1:VG$="":GOSUB 8000:VN$=INP$
1120 PRINT "/TONEV : ";LG=15:GOSUB 8000:NN$=INP$
1125 PRINT "/TCA ES HAZSZAM : ";LG=20:GOSUB 8000:ST$=INP$
1130 PRINT "\R.SZ. IELYSEG : ";VG$="1111 IUDAPEST":LG=20:GOSUB 8000:OT$=INP$
1135 PRINT "/LETKOR : ";ZS=2:VG$="":LG=5:GOSUB 8000:KOR$=INP$
1140 PRINT:PRINT
1145 PRINT VN$;" ";NN$;" (<";KOR$;">)"
1150 PRINT ST$:PRINT OT$:PRINT:PRINT
1155 PRINT "JRAKEZDESHEZ KEREM EGY BILLENTYUT":GET KEY T$:RUN
1160 :

```

READY.



kurzor le	tiltva
kurzor fel	tiltva
kurzor jobbra	mint rendesen
kurzor balra	mint rendesen
INS	mint rendesen
DEL	mint rendesen
HOME	kurzort az első írási pozícióra
CLR	a teljes bevétel törlése
F1	tabulátor (8 hellyel jobbra)
F2	tabulátor (8 hellyel balra)

A tabulátorok használata csak akkor lehetséges, ha előtte az F1 és az F2 funkciók billentyűkhöz a KEY paranccsal a

CHR\$(133) és a CHR\$(134) értékeket rendeltük (a demóban az 1100-as sorban). A tabulálás hosszát a TAB>0) alakíthatjuk. Természetesen itt az értelmes lehetőségnél a beviteli hossz (LG) is szerepet játszik. A bevitelt a szokott módon a (RETURN)-nel kell lezárni.

Az Optinput az alábbi változókat használjuk, ezért ezeket a főprogramban nem szabad alkalmazni.
Z1\$, Z2\$, Z\$, XP, YP, CP, K, K\$, I, ML, LT\$, RT\$, EIN\$

A nyitott koncepció miatt az Optinputot kiválóan használhatjuk a saját programjainkban, feltéve, ha a rutint, például a beviteli sztring megváltoztatásával a saját igényeikhez illesztjük.

```

8000 :
8001 REM *****
8002 REM * C= UJSAG SORSZAM: 147 *
8003 REM * OPTINPUT - C16 *
8004 REM * PROGRAM: KOVACS ZOLTAN *
8005 REM *****
8010 :
8015 Z1$="ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789+-=*"
8020 Z1$=Z1$+"!#$%&'()*+,-./:;<>@,./?":Z2$="1234567890+-E":XP=PEEK(205):YP=POS(X)
8025 IF ZS=1 THEN Z$=Z1$:ELSE Z$=Z2$
8030 INP$=VG$:CP=1:IF VG$="" THEN 8040
8035 IF LG<LEN(VG$) THEN LG=LEN(VG$)
8040 ML=40-YP:IF LG>ML THEN LG=ML
8045 K$=" " + CHR$(20) + CHR$(148):IF LEN(INP$)>ML THEN INP$=LEFT$(INP$,ML)
8050 K$=K$+CHR$(133)+CHR$(134):IF LEN(INP$)<LG THEN INP$=INP$+" ":GOTO 8050
8055 CHAR,YP,XP,INP$:CHAR,(YP-1)+CP,XP," " + MID$(INP$,CP,1) + " " :IF TA=0 THEN TA=8
8060 GETKEY EIN$:IF EIN$=CHR$(13) THEN 8130
8065 IF INSTR(Z$,EIN$) THEN 8080
8070 K=INSTR(K$,EIN$):ON K GOTO 8085,8090,8095,8100,8105,8110,8120,8125
8075 SOUND 1,100,5:GOTO 8060
8080 MID$(INP$,CP,1)=EIN$
8085 IF CP=LG THEN CP=1:GOTO 8055:ELSE CP=CP+1:GOTO 8055
8090 IF CP=1 THEN CP=LG:GOTO 8055:ELSE CP=CP-1:GOTO 8055
8095 CP=1:GOTO 8055
8100 INP$="" :FOR I=1 TO LG:INP$=INP$+" " :NEXT I:GOTO 8095
8105 LT$=LEFT$(INP$,CP-1):RT$=RIGHT$(INP$,LG-CP):INP$=LT$+RT$+" ":GOTO 8055
8110 IF CP=LG THEN 8075:ELSE LT$=LEFT$(INP$,CP-1):RT$=RIGHT$(INP$,LG-CP+1)
8115 INP$=LT$+" "+RT$:INP$=LEFT$(INP$,LEN(INP$)-1):GOTO 8055
8120 IF (CP+TAB)>LG THEN CP=LG:GOTO 8055:ELSE CP=CP+TAB:GOTO 8055
8125 IF (CP-TAB)<1 THEN CP=1:GOTO 8055:ELSE CP=CP-TAB:GOTO 8055
8130 CHAR,YP,XP,INP$:PRINT:I=LG:DO UNTIL MID$(INP$,1,1)<>" " OR I=1:I=I-1:LOOP
8135 IF I=1 THEN INP$="" :RETURN:ELSE INP$=LEFT$(INP$,I):RETURN

```

READY.

Illesztés a C128-ashoz

Az Optinput egyetlen hátránya, hogy az csak a C16, C116 és PLUS/4-es gépre készült. De legalábbis a C128-as tulajdonosok szintén illeszthetik azt a komputerükhöz. Itt csupán azt a tárolócímet kell megváltoztatni, ahol a kurzorpozíció áll. A C16, C116, PLUS/4-es gépeknél ez a cím a 205-ös, a C128-asnál azonban a 235-ös.

A pozíció lekérdezése a 8020-as sor végén történik:
XP = PEEK(205):YP = POS(X)

A C128-as géptulajdonosok ezt a részletet így írják át:
XP = PEEK(235):YP = POS(X)

Ennyi az egész. Ettől a pillanattól kezdve az Optinput a C128-ason is futni fog.

Turbo-block

Ez a rövid assembly program a nagyfelbontású grafikus képernyőre való elforgatás nélküli, kitöltött téglalapok rajzolását (BOX) gyorsítja meg.

A BASIC töltő lefutása után a rutin a SYS 1536, TÍPUS,X1, Y1,X2,Y2 utasítással hívható. Itt a téglalap bal felső sarkát az X1, Y1, a jobb alsó sarkát az X2, Y2 pontkoordináta párok határozzák meg. A TÍPUS háromféle értéket vehet fel. 0 esetén az utasítás törli, 1 esetén rajzolja, 2 esetén invertálja a kijelölt területet.

A sebességét jellemzi, hogy 1000 darab véletlenszerűen választott téglalap rajzolása 21 másodpercet vett igénybe, s egy teljes képernyő méretű blokk kirajzolással kb. 0,7 másodpercig tart. Összehasonlítva az eredeti BOX utasítással, ez kb. 100-szoros sebesség.

Az X koordinátáknak a 0—139, az Y koordinátáknak a 0—199 tartományba kell esniük, továbbá az első koordinátáknak kisebbeknek kell lenniük a másodikaknál (X1<=X2 és Y1<=Y2). Mivel a program ellenőrzést nem tartalmaz, ezeket a kikötéseket vegyük komolyan. A 199-nél nagyobb Y koordináta például garantálja, hogy a rutin beleír a BASIC programunkba.

A BASIC töltőben a kezdőcímet megváltoztathatjuk, a rutin a memória bármely (32768-nál alacsonyabb) szabad területén fut. Az új indítási cím azonos a töltőbeli kezdőcímmel. A rutin hossza 240 byte.

Nagy Tamás

```

1000 REM *****
1001 REM * C= UJSAG SORSZAM: 148 *
1002 REM *****
1003 REM * TURBO-BLOCK *
1004 REM *****
1005 REM * C-16,C-116,PLUS/4 *
1006 REM *****
1007 REM * NAGY TAMAS 1989.01.20. *
1008 REM *****
2000 DATA 20,91,94,20,84,90,E0,03,0369
2001 DATA 90,03,4C,1C,99,86,D3,20,0300
2002 DATA 91,94,20,E1,90,A5,14,80,0409

```



```

2003 DATA 02,85,D0,A5,15,8D,AE,03F9
2004 DATA 02,85,D1,20,91,94,20,84,0341
2005 DATA 9D,86,D4,A2,00,8E,00,02,03D9
2006 DATA 20,91,94,20,E1,9D,18,A5,03A0
2007 DATA 14,69,01,85,D5,A5,15,69,02FB
2008 DATA 00,85,D6,20,91,94,20,84,0344
2009 DATA 9D,86,D7,A5,D4,85,D2,8D,0557
2010 DATA AF,02,20,64,C2,84,09,86,03DA
2011 DATA DA,85,D8,E6,D0,D0,02,E6,05A5
2012 DATA D1,A5,D5,C5,D0,A5,D6,E5,0640
2013 DATA D1,90,2B,A6,DA,A5,D8,10,04A6
2014 DATA 89,C2,85,D8,E8,86,DA,E0,05D0
2015 DATA 08,D0,E0,A2,00,86,DA,38,03F2
2016 DATA 00,14,18,A5,8C,69,08,85,0303
2017 DATA 8C,A5,8D,69,00,85,8D,A9,03E2
2018 DATA 00,85,D8,18,90,C5,08,A5,0377
2019 DATA 8C,48,A5,8D,48,A4,09,A5,0470
2020 DATA D4,85,D2,A5,D8,A6,D3,D0,05F1
2021 DATA 07,49,FF,31,8C,18,90,09,02BD
2022 DATA E0,01,D0,03,11,8C,2C,51,02CE
2023 DATA 8C,91,8C,E6,D2,A5,07,C5,05A2
2024 DATA D2,90,16,D8,C0,08,00,0B,0483
2025 DATA A0,00,18,A5,8C,69,40,85,0317
2026 DATA 8C,A5,8D,69,01,85,8D,D0,040A
2027 DATA CA,68,85,8D,68,85,8C,28,03E5
2028 DATA 00,A0,60,48,41,52,44,57,0326
2029 DATA 4F,52,48,53,20,27,38,37,01F5
2030 BA=1536:REM START CIM
2031 PRINT"KONTROL INDUL !"
2032 FORK=0TO29
2033 C=0
2034 FORB=0TO7
2035 READA$
2036 C=C+DEC(A$)
2037 NEXTB
2038 READA$
2039 IF C<>DEC(A$) THEN PRINT"ADATHIBA
A"2000+K"-ADIK SORBAN !":STOP
2040 NEXTK
2041 PRINT"KONTROL KESZ !"
2042 PRINT"BEIRAS INDUL !"
2043 RESTORE
2044 FORK=0TO29
2045 FORB=0TO7
    
```

```

2046 READA$
2047 POKEBA+K*8+B,DEC(A$)
2048 NEXTB
2049 READA$
2050 NEXTK
2051 PRINT"OK."
2052 PRINT"START :
2053 PRINT" SYS"BA",TYP,X1,Y1,X2,Y2
2054 END

READY.
    
```

```

1000 GRAPHIC1,1
1010 CHAR1,13,12,"TURBO BLOCK DEMO"
1020 GOSUB1230
1030 SYS1536,2,100,92,235,107
1040 GOSUB1230
1050 FORI=0TO360STEP20
1060 X=20+I/9*7
1070 Y=100-SIN(I*PI/180)*60
1080 SYS1536,2,X-10,Y-20,X+10,Y+20
1090 NEXT
1100 GOSUB1230
1110 GRAPHIC1,1
1120 FORI=0TO1000
1130 A=INT(RND(0)*320)
1140 B=INT(RND(0)*200)
1150 C=INT(RND(0)*(320-A))
1160 D=INT(RND(0)*(200-B))
1170 S=INT(RND(0)*3)
1180 SYS1536,S,A,B,A+C,B+D
1190 NEXTI
1200 GOSUB1230
1210 GRAPHIC0
1220 END
1230 T=TI
1240 GETX$:IFX$=""ANDTI-T<150THEN1240
1250 RETURN

READY.
    
```

A SZÁMALK oktatási irodája a következő tanfolyamokat indítja

A tanfolyamok megnevezése	tartama	ára	időpontok
COMMODORE TANFOLYAMOK			
C-64 gépkezelés	3 nap	3500,—	máj. 2-4.
C-64 alapozó I. (BASIC)	5 nap	5500,—	máj. 8-12.
C-64 alapozó II. (fájlok)	5 nap	5500,—	jún. 5-9.
C-64 Assembly	5 nap	5500,—	jún. 12-16.

**Számítástechnika-alkalmazási
Vállalat Oktatási iroda**
Bp., XI. Szakasits Árpád u. 68.
Levél cím: 1502 Budapest 112
Pf. 146 Telex: 22-4498



IBM PC ALAPSZOFTVEREK

ASSEMBLER I.	5 nap	7450,—	ápr. 17-21.
ASSEMBLER II.	5 nap	7450,—	máj. 1-5.
OS 2 rendszerprogramozó ismeretek	5 nap	7950,—	ápr. 10-14.
OS 2 Assembler programozás	5 nap	7950,—	máj. 15-19.

IBM PC PROGRAMOZÁSI NYELVEK

Turbo Pascal I. (4. o.), kezdő	5 nap	7500,—	máj. 8-12.
Turbo Pascal II. (4. o.), haladó	5 nap	7500,—	máj. 15-19.

BALATONKENESE:

Professional COBOL	5 nap	7500,—	máj. 22-26.
BASIC az IBM PC-kre	5 nap	7450,—	jún. 12-16.

PROGRAMCSOMAGOK

Műhelygyakorlat dBASE III-ban (programozóknak!)	5 nap	7950,—	ápr. 10-14. máj. 22-26.
--	-------	--------	----------------------------

CLIPPER (dBASE-t nem ismerőknek)
(kezdő + haladó)

5+5 nap	15000,—	máj. 8-19.
---------	---------	------------

BALATONKENESE:

CLIPPER (dBASE-t ismerőknek)	5 nap	7950,—	máj. 22-26.
AUTOCARD grafikai programcsomag	5 nap	7500,—	ápr. 24-28.

KORSZERŰ ADATBÁZISKEZELŐ

SOL a relációs adatbázisok nyelve	5 nap	4950,—	ápr. 10-12.
-----------------------------------	-------	--------	-------------

Tanfolyamszervezők: Mészárosné Nagy Erzsébet
Hettigné Böszörményi Éva
Telefon: 853-111 220 vagy 229 mellék



Finom SCROLL-program

A képernyő legalsó sorában képpontonként haladó feliratot állít elő. A program működése közben BASIC programot lehet szerkeszteni, futtatni, valamint olyan gépi kódú programot futtatni, amelyik nem írja át a megszakítás vektort. Az osztott képernyő (GRAPHIC 2 és GRAPHIC 4) nem működik, helyette a grafikus képernyő látszik. Ha szükség van az osztott képernyőre, ki kell kapcsolni a program működését (SYS 4153). A lemezegység használata közben a kép ugrál, rángatózik. Ezt a „szépség-hibát” az okozza, hogy a KERNAL a pontos időzítés miatt letiltja a rásztermegszakítást.

Amint a felirat fut, az alsó sorba nem lehet írni.

Az F1 gomb megnyomása vagy a SYS 4153 utasítás ki-, illetve bekapcsolja a programot.

RESET után újra indítható: SYS 4136.

A felirat \$ 1100-tól helyezkedik el a memóriában. Az indítás után a BASIC terület kezdete \$ 1600 lesz, tehát a program és a felirat betűi \$ 15FF-ig foglalják el a tárat. A betűk képernyőkóddal vannak tárolva. A kiírás a teljes memóriát tudja kezelni. Ha a felirat hosszabb, a BASIC terület kezdete mutatót (\$2B, \$2C) át kell írni. A szöveg végére egy \$FF byte-ot kell elhelyezni, az a véggel.

A szöveg betöltésének megkönnyítésére készült a FINOM SCROLL BEL program.

```

0 10 REM *****
1 20 REM *
2 30 REM * C= UJSAO SORSZAM: 149 *
3 40 REM * FINOM SCROLL C16-PLUS/4 *
4 50 REM *
5 90 REM * PROGRAM:PERCZE ISTVAN *
6 100 REM *
7 110 REM *****
8 115 ;2PASS ASSEMBLER INDITASA
9 120 SYS 4352
10 130 .OPT P,0
11 140 *= $1001
12 150 LINE = $C2 ;IRQ KEPERNYOSOR
13 160 VECT = $CE0E ;EREDETI IRQ VECTOR
14 170 LD2 = $1022
15 180 .BYT $20,$10,$C4,$07,$9E
16 190 .ASC "4136"
17 200 .BYT $3A,$F9,$31,$2C,$22
18 210 .ASC "SYS4153"
19 220 .BYT $22,$FA,$C7,$28,$31
20 230 .BYT $33,$29,$3A,$A2,$0,$0
21 240 .BYT $0,$0,$0,$0,$0
22 250 : LDX #$01 ;BASIC KEZDET
23 260 : LDY #$16 ;MUTATO = $1601
24 270 : STX $2B
25 280 : STY $2C
26 290 : LDA #$00
27 300 : STA $1600
28 310 : JSR FF
29 320 : RTS
30 330 FF LDA $0315
31 340 : CMP #$CE ;EREDETI IRQ
32 350 : BEQ MOD ;HA IGEN
33 360 : LDA #$00
34 370 : STA $FF07 ;40 KAR
35 380 : LDA #<VECT ;EREDETI
36 390 : LDX #>VECT ;IRQ VECTOR
37 400 : BNE LERAK
38 410 MOD LDA #<TABL ;TABLAZAT KEZDE-
39 420 : LDX #>TABL ;TENEK BETOLTESE

```

```

430 : STA SZAML+1
440 : STX SZAML+2
450 : LDA #$08 ;SZAMLALO
460 : STA LD2
470 : LDA #<LIRQ ;UJ IRQ VECTOR
480 : LDX #>LIRQ
490 LERAK SEI
500 : JSR $FF4F ;ABLAK,TORLES
510 .BYT $13,$13,$93,$00
520 : STA $0314 ;VECTOR ATIRASA
530 : STX $0315
540 : CLI
550 : RTS
560 :
570 :
580 :
590 LIRQ LDA $FF09 ;MEGSZAKITAS FORR-
600 : AND #$02
610 : STA $FF09 ;RAS TILTASA
620 : LDA $FF1D
630 : CMP #$C2 ;UTOLSO SORBAN
640 : BCS RASZT ;HA IGEN
650 :
660 :
670 : LDA #$17 ;KEPERNYOABLAK
680 : CMP $07E5 ;ALSO SZELE=25
690 : BCS IRQ1 ;HAC25
700 : DEC $07E5 ;HA=25 CSOKKENTI
710 IRQ1 LDA #$08
720 : LDX #LINE
730 : STA $FF07 ;30 KARAKTER
740 LIRQ STX $FF08 ;KOVETKEZO MEGSZ
750 : JSR $CFBF ;KAZETTA,100
760 : JSR $CECD ;HANG
770 : LDA LD2
780 : AND #$02 ;BILLENTYU KESL
790 : BNE VEGE
800 : LDA $FB ;ROM BANK MENTES
810 : PHA ;
820 : LDA #$00
830 : STA $FB
840 : PHP
850 : CLI
860 : JSR $0B11 ;BILLENTYU
870 : PLP
880 : PLA ;ROM BANK VISSZA
890 : STA $FB
900 VEGE JMP $FCBE ;IRQ VEGE
910 :
920 RASZT LDX LD2 ;FINOM SCROLL
930 : DEX
940 : STX $FF07
950 : BNE RESCR ;HA MEG NEM 0
960 : LDY #$00
970 SHIFT LDA $0FC1,Y ;KEPERNYO ALSO
980 : STA $0FC0,Y ;SOR ELTOLASA
990 : INY ;BALRA EGY KA-
1000 : CPY #$27 ;RAKTERREL
1010 : BNE SHIFT
1020 KAR LDY #$00
1030 : STA $FF3F
1040 SZAML LDA TABL,Y
1050 : STA $FF3E
1060 : CMP #$FF ;VEGE
1070 : BNE FOLYT ;HA NEM VEGE
1080 : LDA #<TABL
1090 : LDX #>TABL
1100 : STA SZAML+1 ;SZOVEGET ELOL-
1110 : STX SZAML+2 ;ROL KEZDENI
1120 : JMP KAR
1130 FOLYT STA $0FE7 ;UJ KARAKTER
1140 : INC SZAML+1 ;+1
1150 : BNE UJSCR ;
1160 : INC SZAML+2 ;
1170 UJSCR LDX #$08 ;SCROLL SZAMLALO
1180 RESCR STX LD2
1190 : LDX #$00
1200 : BEQ LIRQ ;IRQ FOLYTATASA
1210 .BYT $0
1220 *= $1100
1230 TABL .ASC "EZ A SZOVEG HELYE * * * "
1240 .BYT $FF ;EZ A SZOVEG
1250 .END ;VEGJEL

```

READY.

Finom SCROLL BEI program

A FINOM SCROLL által kiírandó szöveg betöltését segíti elő. Összesen 1280 karaktert lehet beírni a 0–4 számú lapokra. A kiírás mindig a 0 lap első karakterével kezdődik, és az első \$FF byte előtt fejeződik be. A \$FF byteot ez a program automatikusan elhelyezi a szöveg végére. A program indítása után ki kell választani, hogy melyik lapra írjon (általában 0). A beírás közben a képernyő bal felső sarkában az aktuális betű sorszáma látható. A lap írása befejeződik a RETURN gomb lenyomásával, vagy a 256-ik karakter elérésével. A „VÉGE I/N” kérdésre N-nel válaszolva lehet folytatni, javítani. Ha a válasz I és a lap betelt, a következő lapon folytatható a beírás.

A PROGRAMOK BEÍRÁSA:

A \$ 2000–10FF-ig terjedő hex dump listát helyesen beírva a program kész, de üres. Célszerű futtatás előtt elmenteni monitorból: S, „név”, 8, 1001, 10FF vagy szalagra mentés esetén a 8 helyett 1. Ezután RESET, majd a FINOM SCROLL BEI programot kell beírni és elmenteni. A FINOM SCROLL programot betölteni, majd futtatni a következő lépésben. A FINOM SCROLL BEI program betöltése és elindítása után a kiírni kívánt szöveg beírható. A hibátlan szöveg beírása után monitorból a következőképpen lehet elmenteni a kitöltött programot: S, „név”, 8, 1001, 1600 vagy szalagra mentés esetén a 8 helyett 1-et írva.

Az így elkészített program a gép bekapcsolása után DLOAD vagy LOAD utasítással betölthető és RUN-nal elindítható.

Percze István

```

○ 1 REM *****
○ 2 REM * C= UJSAG             149 *
○ 3 REM * FINOM SCROLL BEIRO C16-PLUS4 *
○ 4 REM * PERCZE ISTVAN           *
○ 5 REM *****
○ 10 INPUT "MELY HANYADIK LAPRA IRJAK 0 - 4 ";L
○ 20 IFL<0ORL>4THENGOTO10:ELSEI=DEC("1100")+L*255
○ 30 PRINT "I":K=DEC("0008"):C=0:B$="":C$=""
○ 40 PRINT "A KIIRANDO SZOVEG MAX. 256 KARAKTER."
○ 50 DO UNTIL A=130RC=257
○ 60 PRINT "C":PRINT "B#C#CHR$(27)"
○ 70 GETKEYA$:A=ASC(A$)
○ 80 IFAC>200RC=0THEN100
○ 90 C=C-1:IFC<128THENB$=LEFT$(B$,C):ELSEC$=LEFT$(C$,C-128):GOTO130
○ 100 IFAC<32OR(A>128ANDRC<160)THEN130
○ 110 IFC<128THENB$=B#+A$:ELSEC$=C#+A$
○ 120 C=C+1
○ 130 LOOP
○ 140 PRINT "VEGE I/N ":POKE239,0:GETKEYA$
○ 150 IFA$="N"THEN:PRINT "I":A=0:GOTO50:ELSE PRINT "OK!"
○ 160 FORD=0TODC-1:POKEI+D,PEEK(K+D):NEXT
○ 170 IFL=5THENGOTO220
○ 180 IFC<256THENGOTO210
○ 185 PRINT "KOVEKTEZO LAPRA IRJAK I/N "
○ 190 POKE239,0:GETKEYA$:IFA$<"N"ANDR$<"I"THEN190
○ 200 IFA$="I"THENL=L+1:GOTO20
○ 210 POKEI+D,255
○ 230 END
○
○ READY.
```

Szeretettel várjuk minden kedves Vásárlónkat május 17–25-ig, a BNV „D” szabad területén lévő, árusítással egybekötött kiállításunkon.

Ez idő alatt Parrot lemezeinket 10% kedvezménnyel, a C 1802 színes, 36 cm-es monitort 30 000 Ft-ért árusítjuk.

ENTERPRISE akció a BNV ideje alatt is.

Még mindig megvásárolható az ENTERPRISE 128 Kbyte-os számítógép.

Ára az összes tartozékkal 11 900 Ft.

20 árúház

Tel.: 402-954 *
Viszontlátásra
a BNV-n!

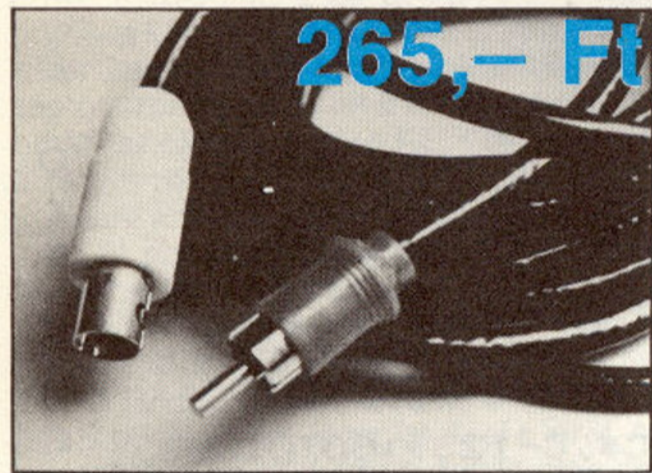
1136 Budapest XIII., Balzac u. 35.

Hardver kiegészítők

Printer Basic **8 459,- Ft** Prodem folyamattábla **2 700,- Ft**

RS 232 szalagkábel PC **2 469,- Ft** CCX-Epson interface

Antennakábel



Videokábel **300,- Ft**

PC-Centronix kábel



RS232 interface



CCX-Epson interface



C64-256K tárbővítő



Zero power



Basic emlékeztető C64 **153,- Ft**

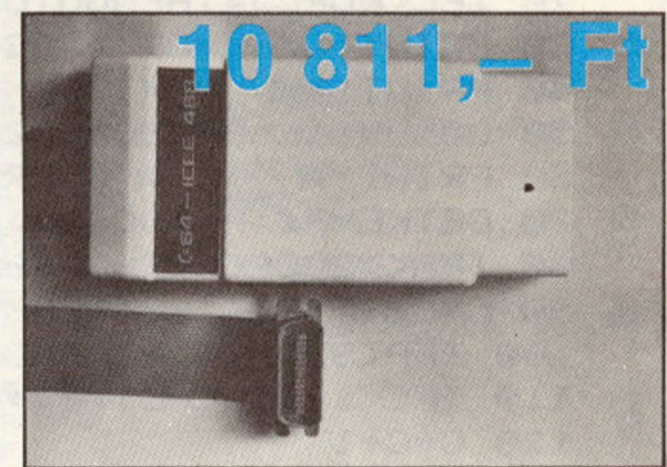
Basic emlékeztető C16 **102,- Ft**

Deltex-Robotron
(6011, 6120, 6125) **24 152,- Ft**

C64 és ZX epromégető



IEE488 interface



Fényceruza **2 500,- Ft**

Resetgomb **300,- Ft**

Oktatótábla C64 **4 380,- Ft**

Oktatótábla C16 **3 600,- Ft**

Oktatótábla C + 4 **3 240,- Ft**

Nemzetközi
tízes billentyűzet



MPS 801 festékszalag



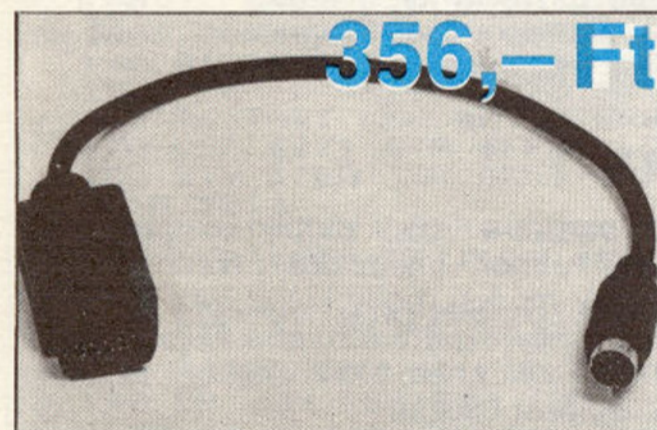
MPS 803 festékszalag



Videoton párhuzamos interface **3 326,- Ft**

Commodore multifele transzfer **2 545,- Ft**

P/4 Interface kábel (joy)



Seikosha SP180-as festékszalag



Videoton soros interface **4 220,- Ft**



Porvédő (C64, 1541, +4, SP-180) **41,- Ft**

Toldó kábel EPSON printerhez **5 500,- Ft**

Felhívjuk Egyesületi tagjaink figyelmét arra, hogy a NOVOTRADE RT. bonok nem beválthatók, hanem a 2C Áruház áruira levásárolhatók.

Számítástechnikánk a jövő technikája

**Örömmel látjuk a
2C
áruházban**

Íme néhány ajánlat az áru kínálatból:

ENTERPRISE 128 Kbyte + magnó	11 900,—
C 64 + GEOS	19 960,—
1530 Datasette	3 500,—
Nyomtatószalag	700,—

A gépekhez sokféle játék- és oktatóprogram, kiegészítő berendezés kapható.

**2C
áruház**

1136 Budapest XIII., Balzac u. 35. Tel. 402-954
Nyitva: hétfőtől péntekig 9-től 18 óráig

Számítástechnikánk a jövő technikája



ANIM +4, avagy egy kis animáció számítógéppel

Gondolom, mindenki készített már pörgetős filmet, például a tankönyve sarkába. Ez adta az ötletet, hogy számítógépes animációval foglalkozzam. Az elv mindenki számára ismerős. Kis képeket készítünk, amelyek csak minimálisan térnek el egymástól, majd ezeket gyors egymásutánban levetítjük. A könyv sarkában lévő figurák kb. 10 kép/mp-es sebességgel egymásra vetítve folyamatos mozgás látszatát keltik.

Számítógéppel ez a sebesség könnyen túlléphető. A mi esetünkben a grafikus képernyő egy darabját használjuk vetítésre. Ennek mérete, bizonyos szempontok miatt, 40×24 képpont.

Ezekből 400 darabot tárolunk, ami 30-40 mp filmet tesz lehetővé. A szerkesztést nagytárolva végezzük, ez könnyebb, mintha a grafikus képernyő kicsiny pontjaira meresztgetnénk a szemünk.

A program üzembe helyezése: Fordításhoz írjuk át a 20. sort: 20. OPT 00,P

Fordítás után Monitorban adjuk ki a következő parancsot:

F 1800 189F F1

F 18A0 1BFF FF

F 1C00 1C9F 01

F 1CA0 1FFF FF

F 2000 25FF 00

Ezután mentjük ki a programot:

S"ANIM G2600", 1,1800,4000.

illetve ha menünk is van (lásd alább), akkor S"ANIM G2600", 1(1000,4000).

A program indítása monitorból G2600-zal történik.

A program működése:

Induláskor egy üres képernyőn egy ?-et látunk. Ez az aktuális rajzpozíciónk. Ha ez normál ?, akkor üres, ha inverz, akkor teli mezőn állunk. Egy karakter-hely egy grafikus pontnak felel meg. A képernyőn a kurzor vezérlő billentyűkkel mozoghatunk. Egy pontot a * karakterrel írhatunk és a SPACE gombbal törölhetünk. Ha folyamatos vonalat szeretnénk húzni, akkor ez a módszer túl lassú. Erre szolgál a SHIFT+CRSR kombináció, törölni a C=CRSR együttes használatával lehet. A teljes képet a CLEAR billentyűvel törölhetjük.

Ha filmünkben az egész képet scrollozni (mozgatni) akarjuk, akkor használjuk az S parancsot. Ekkor a szerkesztő scroll üzemmódba tér át. Ezt a ? eltűnése jelzi. A képet a kurzor billentyűkkel mozgathatjuk a kívánt irányba. Szerkesztésbe a SPACE-val térhetünk vissza.

Az elkészített képet az E gombbal megnézhetjük a grafikus képernyőn. Ezt nem árt gyakran megtenni, mert a nagyításban sok minden másképp látszik, mint eredetiben. Szerkesztésbe itt is a SPACE visz vissza.

Ha a kép elnyerte tetszésünket, akkor eltesszük a memóriába a K parancssal. Ekkor a kép nem tűnik el a szerkesztőből azért, hogy a következő fázist meg lehessen szerkeszteni. Vegyük figyelembe a lejátszási sebességet is! Ez könnyen megrézfálhat bennünket, mert úgy érezzük, hogy már sokat dolgoztunk, pedig még csak 2 mp film (20-40 kép) készült el.

Ellenőrzéskor és lejátszáskor több képmező közül választhatunk. Ezek a grafikus képernyőn, egymás mellett helyezkednek

el. Összesen hat darab közül (0-5) választhatunk az X paranccsal. Ehhez az X megnyomása után egy 0-5 közti számot nyomjunk meg. A kiválasztott képmező száma a jobb alsó sarokban látszik (az egyedül álló szám). A másik három szám az éppen szerkesztés alatt álló kép sorszáma.

A filmünket az L paranccsal játszhatjuk. Ha vége a filmnek, a SPACE-val térhetünk vissza szerkesztésbe. Ha a filmet többször egymás után szeretnénk látni, akkor a D parancsot használjuk. Ez a demo üzemmód. A szerkesztésbe a SPACE visz vissza, ha lejátszódott a film. Ezt úgy lehet legegyszerűbben elérni, hogy a film végéhez közeledve lenyomjuk a SPACE-t és nyomva tartjuk.

Lehetőség van a lejátszási sebesség állítására is. Ezt a szám-billentyűkkel tehetjük meg. A 0=10 kép/mp 1=12 kép/mp 2=15 kép/mp 3=20 kép/mp.

Ha a lejátszás során kiderül, hogy valamit elrontottunk, akkor van lehetőség a javításra. A már elrakott filmkockák közt előre-hátra a > és < gombokkal lépkedhetünk. Ha nem állunk a film végén, akkor a K parancs nem a megszerkesztett képet hagyja a képernyőn, hanem a memóriából előveszi a következőt, és ezt rajzolja a szerkesztőbe. Ha egy képen módosítunk, de nem a K gombbal jövünk le róla, hanem a <vagy a> gombbal, akkor a változtatás nem őrződik meg!

Ha sok képet kell mozognunk, akkor használjuk a G parancsot. Ekkor a kép jobb alsó sarkából eltűnnek a számok, helyettük egy ? jelenik meg. Most kell beírni a keresett kép sorszámát. A RETURN lenyomására megjelenik a keresett kép. Ha az utoljára szerkesztett kép számánál nagyobb számot írunk be, akkor automatikusan az utolsó kép kerül a képernyőre. Ezt kihasználhatjuk akkor, ha az utolsó képre akarunk állni, de nem tudjuk a számát. Ekkor egy 400-nál nagyobb számot írunk be. A G lenyomásakor a bal felső sarokban megjelenő @jelek semmilyen hibát nem okoznak. A ROM rutin miatt nem lehet a 0. képre ugrani. Ezt úgy tehetjük meg, hogy az 1. képre ugrunk, és utána kiadunk egy <parancsot.

Ha ki akarjuk menteni a kész filmünket, akkor nyomjuk a Q-t. Ekkor kiíródik a film kezdő és vég címe, és kilépünk a monitorba. Itt az S"név",8,4000, <vég cím> paranccsal menthetjük lemezre a filmet. A programba kétféleképpen térhetünk vissza: G2600 — törli a filmet, G260D — nem törli a filmet.

Ha nem elég a kép alján mozgó menü, akkor írhatunk saját menüt. Ezt a rendes képernyőn megírjuk, majd egy T 800 FFF 1000 paranccsal elhelyezzük a programba. Programból az M paranccsal kapcsolhatunk a menüre.

A program leírása:

Azért kellett az egész programot gépi kódban megírni, mert a "BASIC + gépi kódú rutinok" megoldás nem működött. A hibát a ROM okozta, oly módon, hogy hiába határoltam le a BASIC munkaterületet, a ROM alatti területre akkor is bizonyos „hulladékok” kerültek. Sajnos nem tudom, honnét. Ez pedig a filmkockák megsemmisülését okozta.

A program indításakor a változó terület feltöltése, a kép törlése után a főprogram rutin következik. Ez végzi a billentyűzet figyelését és a megfelelő rutinokra való ugrást.

Az első ilyen rutinok az irányítást végzik. Itt döntjük el, hogy csak mozgatás volt, vagy mozgatás és rajzolás. Itt található az egy pont törlését vagy írását végző rutin is. A kép törlésénél beállítunk egy 40×24-es ablakot, és ezt töröljük egy ROM rutinnal.

A következő az E parancs rutinja. Ez meghívja a grafikus képernyőt bekapcsoló, törlő rutint, majd azt a rutint, mely a karakteres képernyő grafikus képernyőre másolását végzi. Ezután egy billentyűzet figyelés történik. Ha megnyomtuk a SPACE billentyűt, visszaugrunk a főágba. A következő nagyobb rutin a kép grafikára történő másolását végzi. A másolás soronként történik, ehhez x, y koordináta szükséges. A pont megjelenítéséhez ki kell számolni a pont társbeli címét. Ezt ROM rutin végzi.

A kép megőrzését végző rutin két lépést hajt végre. Először a képet a grafikus képre másolja, az első rutinnal, majd innét teszi a memória megfelelő helyére. Ha a képjelző (\$EO) tar-

talma nulla, akkor ugrik a főprogramba. Ha nem, akkor a következő képet előveszi a memóriából, és a képernyőre másolja. Ezt a NOR rutin végzi el.

A lejátszást végző rutin a filmkockákat az aktuális képmezőbe másolja, majd vár a lejátszási sebességtől függő ideig, utána veszi elő a következő kockát. Ha vége a filmnek, várakozik a SPACE megnyomásáig. A lejátszási sebesség konstansait a TAB1 táblázat tárolja. Ha ide 1-est írunk, akkor a 400 képet kb. 2 mp alatt lejátsza a filmező rutin. A konstansok értékét a 870/N képlettel kapjuk meg, ahol N a képek száma mp-enként.

A következő két rutin a > és a < parancs rutinja. A < rutin az EO címre FF-et ír, ezzel jelzi, hogy nem a film végén állunk.

A scrollozás fő rutinja hasonlít a főprogramra. Itt csak a CRSR és a SPACE billentyűt kell figyelni. A fel és le scroll ROM rutinnal történik, a jobbra-balra pedig saját rutinnal.

Az interrupt rutin végzi a billentyűzet puffer feltöltését és a menü scrollozását. Ennek sebességét a 3F8-ra írt 4-es érték állítja.

A film kimentését segítő rutin a kezdőcím kiírása után egy ROM rutinnal kiszámítja a végcím értékét és hexadecimális formában kiírja. A visszatérési cím kiírása után kilép a monitorba.

A NOR rutin a kép memóriából a szerkesztőbe másolja a képet. Ez két részben történik: először a memóriából a grafikusra, majd a grafikusról a karakteres képernyőre másolja át a képet.

A menü kiiratását végző rutin az FF14-es TED regisztert állítja át. Ez a képernyő kezdőcímét tárolja a színmemóriával együtt, tehát ha ide az eredeti 8-as értéket írjuk, akkor a színmemória 800-, a képmemória C00- helyezkedik el. Ezt úgy állítja

be, hogy a színmemória 1000-, a képmemória 1400- legyen. A SPACE gomb megnyomására visszaállítja az eredeti képernyőt és visszatér a főprogramba.

A következő egység a G parancsot hajtja végre. Itt beállítunk egy kis ablakot a jobb alsó sarokba, majd a „parancssor beolvasása” (870N) ROM rutinnal beolvasunk egy sorszámot. Egy ciklus segítségével beállítjuk a kép kezdőcímét és a képernyőre másoljuk.

A képmező beállításánál a beolvasott szám alapján beállítjuk az ablak x koordinátáját. (TABX táblázat). A képmező címét is kiszámoljuk, méghozzá mindhárom sorát, erre a grafikus képmezőt törlő rutinnak van szüksége. Ezután visszaállítjuk az eredeti ablakot és ugrunk a főrutinra.

A program elején azért definiáltam címkeként a leggyakrabban használt címeket, hogy gyorsabban és biztonságosabban lehessen a programot beírni. A fordításhoz a 2-PASS ASSEMBLER-t használtam, de a BASIC kezdőcímet 2100-ról 3100-ra írtam át.

Korábban elkészített film betöltése

Vagy az ANIM indítása előtt, vagy kilépés (Q) után történhet. LOAD" <név> ",1,1, illetve lemezről LOAD" <név> ",8,1. Ezután írjuk be a film végcímének alsó, illetve felső byte-jának decimális értékét a D4-D5 címre (POKE 212, <alsóbyte>: POKE 213, <felsőbyte>). Ezután monitorból G260D paranccsal indítsuk az ANIM-ot.

Hering László

```

○ 1 REM *****
2 REM *      C= UJSAG      SORSZAM: 150 *
3 REM *      ANIMACIO      C=+4 *
4 REM *      PROGRAM: HERING LASZLO *
○ 5 REM *****
10 SYS 4352
20 .OPT
○ 30 *= $2600
40 D0 = $D0:D1 = $D1;CRSR POZICIO
50 D2 = $D2:D3 = $D3;KEPEK POINTERE
○ 60 D4 = $D4:D5 = $D5;FILM VEGE
70 D6 = $D6:D7 = $D7;SZOV. MUTATO
80 D8 = $D8:D9 = $D9
90 DA = $DA:DB = $DB
○ 100 E0 = $E0:SEB = $E1;FILM SEBESSEG
110 X1 = $DC:X2 = $DE:X3 = $E2;KEP MEZO CIME
120 KK = $E4;KEP MEZO KOORDINATAJA
○ 130 ; ----- START -----
140 START SEI:LDA #$40:LDY #0:STA D3:STY D2:STA D5;KEPEK VEGE:STY D4;MUTATO
150 ; ----- RESTART -----
○ 160 RESTART LDA #4;INTERRUPT SCROLL:STA $3F8;SEBESSEG:LDA #$73:STA SEB;FILM SEB.
170 LDA #$71:STA $FF15:LDY #0:STY $53B
180 STY $7ED:JSR GRKI:JSR $D888:LDY #0:LDA #$C:STY D0:STA D1:LDA #<INT:LDX #>INT
:STA $314
○ 190 STX $315:LDA #<SZOV:LDX #>SZOV:STA D6:STX D7:CLI:LDA #$30:STA $FE3:STA $FE4
200 STA $FE5:SEI:LDA $FF0A:AND #$FE:STA $FF0A:LDA #$04:STA $FF0B:CLI:LDA #$FF:ST
A E0
○ 210 LDA #0;0.KEPMEZO:JMP KM1;BEALLITASA
220 ; ----- FO PROGRAM -----
230 FOPRG LDY #0:LDA (D0),Y:STA $3F7:CMP #$20:BEQ FP:LDA #191;.BYT $2C
○ 240 FP LDA #63:STA (D0),Y:LDA D0:LDY D1:STA D8:STY D9:LDA #$D5:STA $FF13:JSR GRK
I
250 GOMB JSR $EBD9:CMP #0:BEQ GOMB:PHA:LDY #0:LDA $3F7:STA (D0),Y:PLA
○ 260 CMP #"I":BEQ JOB:CMP #"II":BEQ BAL:CMP #"X":BEQ LE:CMP #"J":BEQ FEL:CMP #"*"
270 BEQ IRD:CMP #" ":BEQ TOR:CMP #"E":BEQ LES:CMP #"K":BEQ KES:CMP #"J":BEQ CLR
280 CMP #".":BEQ KOV:CMP #",":BEQ ELO:CMP #"L":BEQ LEJ:CMP #"Q":BEQ SAV:CMP #"M"
290 BEQ MEN:CMP #"D":BEQ DEM:CMP #"S":BEQ SCR
○ 300 CMP #"X":BEQ MEZO
310 CMP #"G":BEQ GOX:TAX:AND #$F0:CMP #$30:BEQ SEBE:JMP FOPRG.
320 ; ----- UGRD TABLA -----
○ 330 JOB JMP JOBRA:BAL JMP BALRA:FEL JMP FFEL:LE JMP LLE:IRD JMP BEIR:TOR JMP TOR
OL
340 LES JMP NEZ:KES JMP ELRAK:CLR JMP CLS:KOV JMP KNOV:ELO JMP KCSOK:SAV JMP SAV
E.
○ 350 MEN JMP MENU:LEJ JMP FILM:DEM JMP DEMO:SCR JMP SCROL:GOX JMP XGOTO:SEBE JMP
SEBES
360 MEZO JMP KEPMEZ
○ 370 ; ----- CRSR LE -----
380 LLE LDA D0:CLC:ADC #40:STA D0:LDA D1:ADC #0:STA D1:LDA D1:CMP #$F:BNE OK:LDA
D0
○ 390 LSR:CMP #$60:BMI OK:JMP VISZ
400 ; ----- CRSR FEL -----
410 FFEL LDA D0:SEC:SBC #40:STA D0:LDA D1:SBC #0:STA D1:CMP #$B:BEQ VISZ
420 ; ----- JO KOORDINATAK -----
○ 430 OK LDA $543:CMP #1;VOLT SHIFT:BNE OK1:JMP BEIR
440 OK1 CMP #2;VOLT C=:BNE OK2:JMP TOROL:OK2 JMP FOPRG

```

```

450 ; -- KOORDINATAK VISSZA ALLITASA --
460 VISZ LDA D8:LDY D9:STA D0:STY D1:JMP FOPRG
470 ; ----- CRSR BALRA -----
480 BALRA LDA D0:SEC:SBC #1:STA D0:LDA D1:SBC #0:STA D1:CMP #B:BEQ VISZ:BNE OK
490 ; ----- CRSR JOBRA -----
500 JOBRA JSR NOV0:LDA D0:CMP #C0:BNE OK:LDA D1:CMP #F:BNE OK:BEQ VISZ
510 ; ----- EGY PONT BE -----
520 BEIR LDA #160:LDY #0:STA (D0),Y:JMP FOPRG
530 ; ----- EGY PONT KI -----
540 TOROL LDA #32:LDY #0:STA (D0),Y:JMP FOPRG
550 ; ----- KEP TORLES -----
560 CLS LDA #23:STA #7E5:JSR #D88B:JMP FOPRG
570 NOV0 LDA D0:CLC:ADC #1:STA D0:LDA D1:ADC #0:STA D1:RTS
580 ; ----- KEP ELLENORZES -----
590 NEZ JSR GRBE:JSR GRMAS:CLI:BV JSR #EBD9:CMP #20:BNE BV:JSR GRKI:JMP FOPRG
600 ; ----- GR. BEKAPCS -----
610 GRBE LDA $FF06:ORA #20:STA $FF06:LDA #8:STA $FF12:LDA #18
620 STA $FF14:RTS
630 ; ----- GR. KIKAPCS -----
640 GRKI JMP #C7C9
650 ; ----- KEP GR.-RA -----
660 GRMAS JSR GRCLS:LDY #0:LDA #C:STA D0:STY D0:GR3 LDY #0:GR2 LDA (D0),Y:CMP #
160
670 BEQ POBE:GR1 INC #2AD:INY:CPY #40:BNE GR2:INC #2AF:CMP #24:BEQ GRE:LDA XK
680 STA #2AD:LDA D0:CLC:ADC #40:STA D0:LDA D0:ADC #0:STA D0:CMP #10:BNE GR3:RTS
690 ; --- GRAFIKUS PONT RAJZOLASA ---
700 POBE TYA:PHA:LDA #19:MEK. ERTEK:STA #88:JSR #C264:HELY SZAMITASA
710 ORA (#8C),Y:RA PONT:STA (#8C),Y:MEGJELENITESE:PLA:TAY:JMP GR1
720 GRE RTS
730 ; ----- GR. TORLES -----
740 GRCLS LDA #0:LDY #39:GC1 STA (X1),Y:STA (X2),Y:STA (X3),Y:DEY:BPL GC1
750 LDA XK:STA #2AD:KOORDINATAK:LDA #0:BEALLITASA:STA #2AE:STA #2AF:STA #2B0
760 RTS:ELF JMP FOPRG
770 ; ----- KEPSZAM NOVELES -----
780 KSZN LDX #2:KSZ INC #FE3,X:LDA #FE3,X:CMP #3A:BNE GRE:LDA #30:STA #FE3,X:D
EX
790 BPL KSZ
800 ; ----- KEP MEMORIABA -----
810 ELRAK LDA D2:CMP D4:BNE KOK:LDA D3:CMP D5:BNE KOK:LDA #0:STA E0:KOK LDA #FE3
820 CMP #34:BNE KM:JMP FOPRG:KM JSR GRBE:JSR GRMAS:LDA X1:LDX X1+1:STA D0:STX D
B
830 SEI:STA $FF3F:LDX #2:EL0 LDY #0:EL1 LDA (D0),Y:STA (D2),Y:INY
840 CPY #40:BNE EL1:LDA D0:CLC
850 ADC #40:STA D0:LDA D0:ADC #1:STA D0:LDA D2:CLC:ADC #40:STA D2:LDA D3:ADC #0
:STA D3
860 DEX:BPL EL0:LDA E0:CMP #0:BNE U2
870 LDA D2:LDX D3:STA D4:STX D5:JSR KSZN:STA $FF3E:CLI:JSR GRKI
880 JMP FOPRG
890 U2 JSR KSZN:JMP NOR
900 ; ----- FILM LEJATSZASA -----
910 FILM JSR LEJAT:FI JSR #EBD9:CMP #20:BNE FI:JMP FOPRG
920 LEJAT LDA D4:BNE LET:LDA D5:CMP #40:BNE LET:RTS:LET JSR GRBE:JSR GRCLS:SEI
930 STA $FF3F:LDY #0:LDA #40:STA D9:STY D8:L3 JSR VIMAS:LDX SEB:LDY #0:LV DEY
940 BNE LV:DEX:BNE LV:LDA D8:CMP D4:BNE L3:LDA D9:CMP D5:BNE L3:LDY #0:LDX #0
950 LW DEX:BNE LW:DEY:BNE LW:STA $FF3E:CLI:RTS
960 ; ----- KOVETKEZO KEP -----
970 KNOV LDA D2:CMP D4:BNE NO1:LDA D3:CMP D5:BNE NO1:LDA #0:STA E0:JMP FOPRG
980 NO1 JSR KSZN:LDA D2:CLC:ADC #120:STA D2:LDA D3:ADC #0:STA D3
990 ; ---KEP MEMORIABOL SZERKESZTOBE---
1000 NOR LDA D2:CMP D4:BNE HU:LDA D3:CMP D5:BNE HU:LDA #0:STA E0:STA $FF3E:CLI:J
SR GRKI
1010 JMP FOPRG
1020 HU LDA #1:STA E0:JSR ATMAS:HOVA JSR GRBE:JSR GRCLS:SEI:STA $FF3F:JSR VIMAS
1030 STA $FF3E:CLI
1040 LDY #0:LDA #C:STY D8:STA D9:NO4 LDY #0:NO3 STY #3F9:LDA #19:STA #88:JSR #
C264
1050 JSR #C264:PONT CIME
1060 AND (#8C),Y:PONT VIZSGALATA
1070 BEQ NO2:LDA #160:BYT #2C:NO2 LDA #32:LDY #3F9:STA (D8),Y:INC #2AD:INY:CPY
#40
1080 BNE NO3:LDY XK:STY #2AD:LDA D8:CLC:ADC #40:STA D8:LDA D9:ADC #0:STA D9:INC
#2AF
1090 LDA #2AF:CMP #24:BNE NO4:LDA D0:LDX D1:STA D8:STX D9:JSR GRKI
1100 JMP FOPRG
1110 ; ----- ELOZO KEP -----
1120 KCSOK LDA #FE4:ORA #FE5:ORA #FE3:CMP #30:BNE CS1:JMP FOPRG:CS1 LDA #1:STA
E0
1130 LDA D2:SEC:SBC #120:STA D2:LDA D3:SBC #0:STA D3:LDX #2:CS3 DEC #FE3,X:LDA #
FE3,X
1140 CMP #2F:BNE CS2:LDA #39:STA #FE3,X:DEX:BPL CS3:CS2 JMP NOR
1150 ATMAS LDA D2:LDX D3:STA D8:STX D9:RTS
1160 ; ----- SCROLLOZAS -----
1170 SCROL JSR #EBD9:CMP #0:BEQ SCROL:CMP #10:BEQ JSC:CMP #90:BEQ BSC:CMP #11
:BEQ LSC
1180 CMP #91:BEQ FSC:CMP #20:BNE SCROL:JMP FOPRG:JSC JMP SCRJ:BSC JMP SCR8
1190 ; ----- FEL SCROLL -----
1200 FSC LDA #23:STA #7E5:JSR #DEF6:ESC+V:JMP SCROL
1210 ; ----- LE SCROLL -----
1220 LSC LDA #23:STA #7E5:JSR #DF04:ESC+W:JMP SCROL
1230 ; ----- JOBRA SCROLL -----
1240 SCRJ LDA #0:LDX #C:STA D0:STX D0:LDY #38:J1 LDA (D0),Y:INY:STA (D0),Y:D
EY:DEY
1250 BPL J1:LDA #20:INY:STA (D0),Y:JSR NOVDA:LDA D0:CMP #C0:BNE J0:LDA D0:CMP #F

```



```

1260 BNE J0:JMP SCROL
1270 ; ----- BALRA SCROLL -----
1280 SCRB LDA #0:LDX #FC:STA DA:STX DB:B0 LDY #1:B1 LDA (DA),Y:DEY:STA (DA),Y:IN
Y:INY
1290 CPY #40:BNE B1:LDA #20:DEY:STA (DA),Y:JSR NOVDA:LDA DA:CMP #C0:BNE B0:LDA
DB
1300 CMP #F:BNE B0:JMP SCROL
1310 NOVDA LDA DA:CLC:ADC #40:STA DA:LDA DB:ADC #0:STA DB:RTS
1320 ; ----- INTERRUPT -----
1330 INT LDA $FF09:STA $FF09:JSR $DB11;BILLENYUZET:DEC $3F8;SCROLL:BEQ INT0:JMP
$FCC3
1340 INT0 LDA #4:STA $3F8:LDY #1:INT2 LDA $FC0,Y:STA $FBF,Y:INY:CPY #34:BNE INT2
1350 LDY #0:LDA (D6),Y:STA $FE1:LDA D6:CLC:ADC #1:STA D6:LDA D7:ADC #0:STA D7
1360 LDA (D6),Y:BNE INTE:LDA #C0V:LDX #>SOV:STA D6:STX D7:INTE JMP $FCC3
1370 ; ----- FILM MENTESE -----
1380 SAVE SEI:JSR $D888:JSR $FF4F
1390 .ASC "SAVE $4000-$":.BYT 0
1400 LDA D4:LDX D5:JSR $FAFF;VEGCIM HEX-BEN:JSR $FF4F:.BYT 13
1410 .ASC "M VISSZATERES A PROGRAMBA : 02600":.BYT 0,0,0
1420 ; ----- KEP MEMORIABOL GR.-RE -----
1430 VIMAS LDA X1:LDX X1+1:STA DA:STX DB:LDA #3:V2 LDY #0:V1 LDA (D8),Y:STA (DA),Y
1440 INY:CPY #40:BNE V1:LDA DA:CLC:ADC #40:STA DA:LDA DB:ADC #1:STA DB:LDA D8:C
LC
1450 ADC #40:STA D8:LDA D9:ADC #0:STA D9:DEX:BNE V2:RTS
1460 ; ----- MENU KIIRASA -----
1470 MENU LDA #10;KEPERNYO:STA $FF14;$1400-RA
1480 MM JSR $EBD9:CMP #20:BNE MM:LDA #8;KEPERNYO:STA $FF14;$C00-RA
1490 JMP FOPRG
1500 ; ----- FILM DEMOZASA -----
1510 DEMO JSR LEJAT:JSR $EBD9:CMP #20:BNE DEMO:JMP FOPRG
1520 ; ----- SCROLLOZANDO SZOVEG -----
1530 SZOV .ASC "
1540 .ASC " RAJZFILM IRO PROGRAM "
1550 .ASC "
1560 SOV .ASC "
1570 .ASC "ANIM ANIM ANIM ANIM ANIM ANIM (C)1988"
1580 .ASC " HLA-SOFT
1590 .ASC "IRANYITAS:CRSR.....EGY PONT BEIRASA:*.
1600 .ASC "EGY PONT TORLESE:SPACE.....KEP TORLESE:CLEAR"
1610 .ASC ".....KEP ELLENORZESE:E.....KEP ELRAKASA:K"
1620 .ASC ".....FILM LEJATSZASA:L.....FILM KIMENTESE"
1630 .ASC " :G.....EGY KEPPEL VISSZA:<.....EGY KEPPEL"
1640 .ASC " ELORE:>.....INFORMACIOK:M....."
1650 .ASC "KEP SCROLLOZASA:S..... X. KEP ELOVETELE"
1660 .ASC " :G.....SHIFT+CRSR:CSIK HUZASA....."
1670 .ASC "C+=CRSR:CSIK TORLESE.....FILM TOBBSZORI"
1680 .ASC " LEJATSZASA (DEMO):D.....LEJATSZASI"
1690 .ASC " SEBESSEG:0-10 K/S 1-12 K/S 2-15 K/S 3-20 K/S"
1700 .ASC "
1710 .ASC "A FILM TAROLASA $4000- TARTENIK....."
1720 .ASC "BETOLTOTT FILM LEJATSZASAHOZ A $D4-$D5 CIMET"
1730 .ASC " A FILM VEGERE KELL ALITANI ES A PROGRAMOT "
1740 .ASC "$2600-TOL INDITANI!!....."
1750 .ASC "RAJZOLASI MEZO ALLITASA:X ES EGY "
1760 .ASC "SZAM (0-5)....."
1770 .BYT 0,0,0,0,0
1780 ; ----- X. KEP KERESESE -----
1790 XGOTO LDY #24;KEPERNYO:LDX #35;ABLAK:STY $7E5;BEALLITASA:STY $7E6:STX $7E7
1800 LDX #39:STA $7E8:JSR $D88B;CLEAR
1810 LDA #FF;DIREKT MOD:STA $3A:JSR $895A;SOR PUFFERBE:STX $3B;OLVASASA:STY $3C
1820 JSR $473;CHRGET:JSR $8E3E;SORSZAM $14/$14:JSR $D88B;ABLAK TORLESE
1830 LDA #0:STA $7E7;MAGY KEPERNYO:STA $7E6;VISSZALLITASA
1840 LDA #23:STA $7E5:LDA #30:STA $FE4:STA $FE5:STA $FE3
1850 LDA #39:STA $7E8
1860 INC $15:LDA #0:LDY #40:STA D2:STY D3
1870 GX1 LDA D2:CMP D4:BNE GX:LDA D3:CMP D5:BNE GX:JMP ODA
1880 GX JSR KSN:LDA D2:CLC:ADC #120:STA D2:LDA D3:ADC #0:STA D3:DEC $14:BNE GX1
:DEC $15
1890 BNE GX1:LDA D2:CMP D4:BNE UFR:LDA D3:CMP D5:BNE UFR
1900 ODA LDA #0:STA E0:LDA D2:SEC:SBC #120:STA D8:STA D2
1910 LDA D3:SBC #0:STA D9:STA D3:JMP HOVA
1920 UFR LDA #1:STA E0:JMP NOR
1930 ; ----- LEJATSZASI SEB -----
1940 SEBES TXA:AND #F;ASC->SZAM:CMP #4:BPL JIJ
1950 TRY:LDA TABI,Y:STA SEB:JIJ JMP FOPRG
1960 ; ----- RAJZ MEZO KIVALASZTASA -----
1970 KEPMEZ JSR $EBD9:CMP #"X":BEQ KEPMEZ:TXA:AND #F0
1980 CMP #30:BNE KEPMEZ:TXA:AND #F;ASC->SZAM
1990 KM1 CMP #6:BPL KEN:TXA:ORA #30:STA $FE7;KIIRAS:LDA TABX,X:STA XK;X KOORDIN
ATA
2000 LDA #D8:STA X1:LDA #1F:STA X1+1
2010 KP1 LDA X1:CLC:ADC #40:STA X1;KEP MEZO:LDA X1+1;CIMENEK:ADC #0;KISZAMITASA
2020 STA X1+1:DEX:BPL KP1
2030 LDA X1:CLC:ADC #40:STA X2:LDA X1+1:ADC #1:STA X2+1:LDA X2:CLC:ADC #40:STA
X3
2040 LDA X2+1:ADC #1:STA X3+1;KEN JMP FOPRG
2050 TABX .BYT 0,40,80,120,160,200,240
2060 TABI .BYT 87,73,59,44;SEBESSEGEK
2070 .END

```

READY.

EGYESÜLETI TAGOK FÓRUMA

Vegyes

C 64

C 64-re FASTLOAD és TURBOTAPE CARTRIDGE egyenként 650,— Ft-ért és leltíthatatlan RESETGOMB 214,70 Ft-ért!
ENTERPRISE-ra JOYSTICK átalakító 351,— Ft-ért! Rende-
lés csak levélben!
Répás Sándor, 1116 Buda-
pest, Fegyvernek u. 10. VI/61.

1987/88-as programokat adok és cserélek kazettán. 1 program másolási díja 20,— Ft. Válaszlevélre részletes tájékoztatót küldök.
ifj. Varga Péter, 4400 Nyíregyháza, Óvoda u. 49.

C 64-hez epromégetőt keresek 2—32K-ig és magyar nyelvű használati leírást.
Eladó: OCEAN—118N FLOPPYDRIVE! Kompatibilis C 16, PLUS/4, C 64, C 128 gépekkel.
Ára: 12 000,— Ft.
Csonki András, 1144 Buda-
pest, Füredi út 7/c.

C 64-es gépemhez C 16-os szimulátort keresek, hogy grafikai és zenei utasításokat fogadni képes legyen.
Gelencsér Zoltán, 3300 Eger, Kolozsvári u. 38.

C 64-es programokat cserélek lemezen. Keresem a DESERT FOX című programot. Listát kérek és küldök.
Matallik Krisztián, 3035 Gyöngyöspata, Szöcsi út 11.

C 64-es játékprogramokat másolok kazettára. 15,— Ft/db. Kérésre listát is küldök.
Horváth Tibor, Győr, Tas vezér út 35/A.

C 64-es játék- és felhasználói programokat cserélek lemezen. Listát kérek és küldök.
ifj. Sipos István, 2098 Piliszentkereszt, Pf.: 10.

Eladó magyar nyelvű ZOOM-PASCAL-64 leírás.
Lehócz Rudolf, 1118 Buda-
pest, Torbágy u. 14. I/3.

Epromégetés 2716—27256-ig, masterből vagy lemezzel 24 órán belül!
Továbbá eladó egy DATA-SETTE és egy 32K—SCARTRIDGE.
Muzsik József. Telefon:
587-835

MPS 801 nyomtató eladó!
Kecskeméti Tibor, 6723 Sze-
ged, Gáz u. 10/8. Telefon:
62/29-968.

Jutányos áron ELADÓ!
500 db C 16, illetve PLUS/4 program. 40 db kazettán, katalógussal, leírásokkal, könyvekkel.
ifj. Sipos István, 2098 Piliszentkereszt, Pomázi út 76.

Vadonatúj Commodore 801 mátrixprinter, kábelekkel, sok papírral rendkívül olcsón eladó! Ajándék: 5 db floppy, 1988-as játékokkal!
Nagy Gábor. Telefon: 655-225

A mindenkori legfrissebb árak ismerete elengedhetetlen a helyes döntéshez.
A havonta megjelenő

PC szalon katalógus

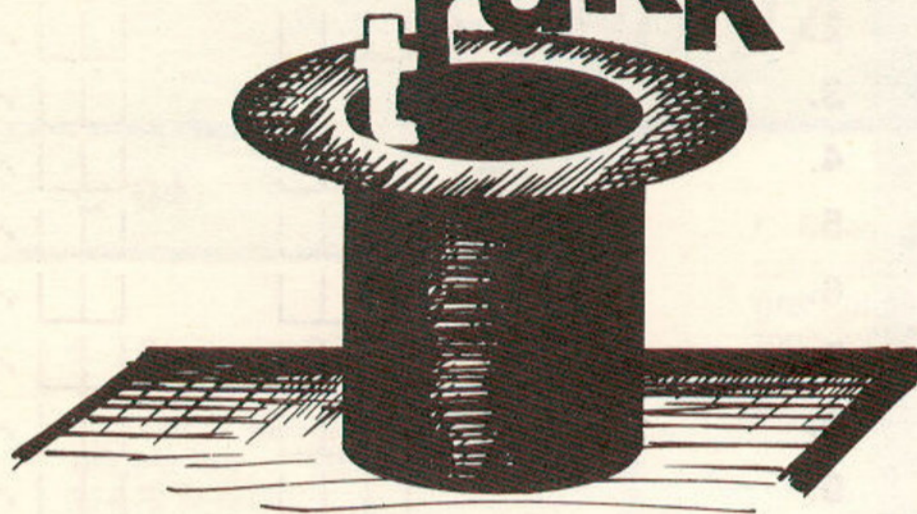
több mint 200 oldalon, közel 100 gyártó és forgalmazó teljes választékának adataival tartalmaz aktuális számítástechnikai árinformációkat a hardver és szoftver területén.

Előfizetési díj: 6 hónapra 4000,— Ft
12 hónapra 6000,— Ft

Ha előre kívánja jegyezni cége számára, kérjük, az alábbi címre küldje el megrendelését!

PC SZALON 1136 Budapest, Sallai u. 6.

Tippek trükkök



a C128-ashoz

A plusz billentyűk

A C-128-as plusz billentyűit (tizesblokk, stb.) a 64-es módban is lehet használni. Hívjuk be (a 64-es módban) a „plusz billentyűk” programot az alábbi módon:

```
„PLUSZ BILLENTYŰK”,8 (RETURN)
```

Az indítás RUN-nal történhet. Mintegy három másodperc múlva a számítógép READY-vel jelentkezik vissza. Ekkor a plusz billentyűzet aktiválva van. A billentyűzet kiosztása az alábbiak szerint alakul:

bill.	ASCII kód	bill.	ASCII kód	bill.	ASCII kód
0	48	9	57	.	46
1	49	—	49	+	43
2	50	NO SCROLL	25	TAB	23
3	51	ENTER	13	ALT	24
4	52	ESC	27	HELP	22
5	53	CRSR LEFT	157	LINE FEED	10
6	54	CRSR RIGHT	29		
7	55	CRSR UP	145		
8	56	CRSR DOWN	17		

A HELP, TAB, ALT és NO SCROLL billentyűk ASCII kódjait úgy választottam, hogy azok ne kapjanak különleges értelmet, és ezért azokat könnyen lekérdezhethetjük programjainkban. Pl.: GET A\$: IF A\$=CHR\$(22) THEN ...: REM HELP

A billentyűzet (ASCII) foglaltságát meg lehet változtatni. Ehhez csak a 220–450. sorok DATA értékeit írjuk át.

A program rendes körülmények között a kazettapufferben áll, amely azonban a Datasette használatát kizárja. A 10-es sorban álló kezdőcím értékének megváltoztatásával a programot bárhová áthelyezhetjük. A helyigény 189 byte.

A PROGRAM LEÍRÁSA

Ha szeretnénk megérteni a működést, azt kell megnéznünk, hogyan történik a billentyűk lekérdezése. A „leosztás” egy táblázatnak felel meg sor-oszlop elrendezésben. Ha szeretnénk tudni, melyik billentyűt nyomtuk meg, akkor ki kell választani a \$DC00 (56320) címen a megfelelő bit törlésével azt a sort, amelyikbe az adott gomb tartozik. Minden ehhez a sorhoz tartozó billentyű esetében a \$DC01-es (56321) címen szintén le kell nullázni a megfelelő bitet. (Arra ügyelni kell, hogy a \$DC00

```

○ 1 rem *****
○ 2 rem *
○ 3 rem * , <<< c - 128 >>> *
○ 4 rem * pot-funkcios billentyuk *
○ 5 rem * - a 64'er alapjan - *
○ 6 rem * 151 *
○ 7 rem *****
○ 8 :
○ 9 :
○ 10 kc=828:rem kezdocim
○ 20 for i=0 to 188:read b:poke kc+i,b:next i
○ 30 poke kc+157,peek(788):poke kc+158,peek(789):rem normal irq
○ 40 hi=int((kc+165)/256):lo=(kc+165)-256*hi
○ 50 poke kc+147,lo:poke kc+148,hi:rem az ascii tablazat kezdete
○ 60 poke 56334,peek(56334)and 254:rem irq kikapcsolasa
○ 70 hi=int(kc/256):lo=kc-256*hi
○ 80 poke 788,lo:poke 789,hi:rem uj irq
○ 90 poke 56334,peek(56334)or 1:rem irq visszakapcsolasa
○ 100 end
○ 110 data 120,165,198,201, 10,240, 27,169,255,141, 0,220,173, 47,208
○ 120 data 41,248,141, 47,208,172, 1,220, 9, 7,141, 47,208,192,255
○ 130 data 240,127,208, 2,240,120,160, 0,173, 47,208, 41,254,141, 47
○ 140 data 208,174, 1,220, 9, 7,141, 47,208,138,201,255,208, 46,160
○ 150 data 8,173, 47,208, 41,253,141, 47,208,174, 1,220, 9, 7,141
○ 160 data 47,208,138,201,255,208, 23,160, 16,173, 47,208, 41,251,141
○ 170 data 47,208,174, 1,220, 9, 7,141, 47,208,138,201,255,240, 54
○ 180 data 201,254,240, 31,200,201,253,240, 26,200,201,251,240, 21,200
○ 190 data 201,247,240, 16,200,201,239,240, 11,200,201,223,240, 6,200
○ 200 data 201,191,240, 1,200,196, 2,240, 12,132, 2,185,225, 3,164
○ 210 data 198,153,119, 2,230,198, 76, 49,234,169,255,133, 2,208,247

```

```

○ 220 data 022:rem help ○
230 data 056:rem 8 ○
○ 240 data 053:rem 5 ○
250 data 023:rem tab ○
○ 260 data 050:rem 2 ○
270 data 052:rem 4 ○
○ 280 data 055:rem 7 ○
290 data 049:rem 1 ○
○ 300 data 027:rem esc ○
310 data 043:rem + ○
320 data 045:rem - ○
○ 330 data 017:rem line feed ○
340 data 013:rem enter ○
○ 350 data 054:rem 6 ○
360 data 057:rem 9 ○
○ 370 data 051:rem 3 ○
380 data 024:rem alt ○
○ 390 data 048:rem 0 ○
400 data 046:rem . ○
○ 410 data 145:rem crsr up ○
420 data 017:rem crsr down ○
○ 430 data 157:rem crsr left ○
440 data 029:rem crsr right ○
○ 450 data 025:rem no scroll ○

○ ready. ○
    
```

címen egyszerre csakis egy bit lehet törölve, hiszen különben nem lehet kiválasztani a billentyű „sort”!)
\$DC01 (56321)

	bit 0	1	2	3	4	5	6	7
0	DEL	RET	CRSR L	F7	F1	F3	F5	CRSR D
\$1 D	3	W	A	4	Z	S	E	SH L
C2 0	5	R	D	6	C	F	T	X
03	7	Y	G	8	B	H	U	V
(4 5	9	I	J	0	M	K	O	N
65 3	+	P	L	-	.	:	@	,
26 0	£	*	;	HOME	SH R	=	↑	/
) 7	1	←	CTRL	2	SPACE	C=	Q	R/S

(R/L=bal/job, SH=shift, R/S=run/stop)

A „plusz” billentyűk lekérdezése alapján véve ezzel a módszerrel történik. Ugyanis ezek is fel vannak osztva oszlopra és sorra.

\$DC01 (56321)

	bit 0	1	2	3	4	5	6	7
\$0 D	HELP	8	5	TAB	2	4	7	1
C1 0	ESC	+	-	LINEF	ENTER	6	9	3
O2	ALT	0	.	CRSR U	CRSR D	CRSR L	CRSR R	NS

U/D/L/R=fel/le/balra/jobbra, NS=no scroll)

Ugyanakkor a sorkiválasztás a \$D02F (53295-ös) cím 0–2. bitjeivel történik. Ha egy billentyű meg van nyomva, akkor a \$D01-ben töröljük a megfelelő bitet. Ha szeretnénk tehát lekérdezni a plusz billentyűk valamelyikét, akkor arra kell figyelni, hogy a \$DC00 címen minden egyes bit be legyen kapcsolva, mivel ekkor egy „normál” gomb megnyomása törölhetne egy bitet a \$DC01 címen! (Ugyanez érvényes fordítva a „normál” billentyűk lekérdezésére!)

A PLUSZ BILLENTYŰK program az alábbiak szerint dolgozik. Generálunk egy gépi kódú programot (10–50. és DATA sorok), amelyet azután beépítünk az interruptba (60–90. sorok). Ez a program azután másodpercenként mintegy hatvenszor lekérdezi a plusz billentyűket, és azok ASCII kódjait a billentyűzet pufferbe írja.

Egyszerű másolásvédelem

Van egy igen egyszerű másolásvédelmi eljárás, amely azonban csak a 1571-es floppyval, és annak 1541-es üzemmódjával lehetséges. A módszer a behelyezett lemez B oldalára történő programmásolás. Az első oldalra egy betöltőt kell vinni, amely neve lehet ugyanaz, mint a behívandó programé. A program hossza egy blokk lesz, és azaz alábbi egy BASIC sorból áll:

```
0 OPEN 1,8,15,"U0>M1": LOAD"FOÓPROGRAM",8
```

Az első utasítás a floppyt átállítja a másik író/olvasó fejre. Ezután a B oldalról behívjuk a FÓPROGRAMOT, és azt automatikusan el is indítjuk. Az általam ismert egyetlen egy 1571-es másolóprogram (mindegy, hogy a 1541-es vagy a 1571-es módban) sem képes a B oldalról másolni. A Thirty Seconds olvasási hibákat ad, mert a 36–70-es trackokról kell olvasni. A Tornado Copyt nem lehet beállítani a második oldalra, mert az a 1571-es módban fut. Más 1541-es másolók esetében pedig ha megfordítjuk a lemezt és úgy akarjuk a B oldalt lemásolni, szintén nem jutunk eredményre, mert ekkor nemcsak a lemez, de a forgás iránya is megváltozik! A filecopy programok csak az első oldalon álló betöltőt tudják lemásolni. A másolás tehát csak beolvasással és ismételt tárolással lehetséges, ami viszont különböző állománytípusoknál (REL, USR) nem is olyan egyszerű!

Hiba a CHAR parancsban!!!

Nemrégén a C128-asom operációs rendszerében, illetve a BASIC ROM-ban egy olyan hibát fedeztem fel, amely ugyan ritkán jelentkezik, ugyanakkor azonban igen kellemetlen. Bár a végrehajtás zavartalan, de nézzük csak meg mi történik, ha a CHAR utasítást a 80 karakteres képernyőre alkalmazzuk. Nos, ekkor fölülírjuk a BASIC RAM-ot! Méghozzá a d600 és d601-es címeket a nullás RAM bankban. Olyan BASIC programok esetében, amelyeknél a FRE(0)>10485, nem kell a következményektől tartani. Nagy programok esetében — és emiatt jelentkezik a hiba oly ritkán — azonban megváltoztatjuk azt!!! Nincs nagy baj, ha mondjuk éppen egy PRINT parancs szövegébe kutyulunk bele. Ekkor a futtatás során legtöbbször elkezd valami villogni. Nagyobb gond, ha épp egy BASIC tokent trafálunk el. Akkor vagy valami hibás programutasítás hajtódik végre, vagy a programfutás hibajelzéssel megszakad. A legborzasztóbb azonban, ha véletlenül egy BASIC sorszámot írunk felül. A programunk maradék része ekkor azonnal elérhetlenné deklarálódik számunkra, illetve igencsak nehézkesnek bizonyul az ismételt rekonstrukció.

Adjuk be bizonyítékként a 80 karakteres képernyőn az alábbi utasítást:

```
CHAR,9,9,"TESZT"
```

Most nézzük meg a monitorral az M 0d5f0 utasítással a nullás bank tárolóját. A 00 és FF zűrzavarban a d600/d601-es cím 0f 09-es tartalma feltűnést kelt.

Persze ez nem hagyott engem nyugodni. A d600/d601 címpárnak m- köze van a 80 karakteres képernyőhöz? Itt valami nem stimmel. Hiszen ott nekünk a VDC chippel van dolgunk, nem a bank 0-val! Valahol a VDC chip egy címzésének a nevezett tárolóba kell „csúsznia”. Ugyanis a 40 karakteres képernyőn ez a hiba sem a grafikus, sem a szöveges módban nem jelentkezik.

Nos, az egészet egy, a BASIC ROM-ban található program hibája okozza. Ez hívja fel a CHAR parancsot az f6838-as címen. Ebben, az f928d-nél álló programban be kell állítani a kurzort. A dolog pikantériája az, hogy itt először a sta \$ff03 parancssóval a 14-es bankot kapcsoljuk be (mert itt van a Load-konfigurációs-regiszter C, és a prekonfigurációs regiszter C-ben az fd503-nál rendes körülmények között a \$01=bank 14 van beállítva). Később a kurzor pozícióját a tulajdonképpeni PLOT rutinban átadjuk a VDC-nek. Ennek a d600/d601-es címen, a 15-ös bankban kell megtörténnie, azaz a bekapcsolt I/O tartomány mellett. Mi azonban a 14-es bankba írunk, azaz a karakter ROM-ba. Ez azonban csak olvasásra szolgál, amit az MMU annyira rossznéven vesz, hogy balkézről a nullás bankba ír. Egyébként később a CHAR rutin lefutásakor a sztring kiadásánál rendben átkapcsolunk az I/O tartományra:

```
pha
lda #$00
sta $FF00
pla
```

Legkésőbb most megérthetjük, miféle kód is kerül a nullás bankba. A kurzor címe a VDC chipben a cím=hex\$ (sor*80+oszlop) képlet szerint számítható. A CHAR,9,9 esetében — fenti példa — a \$02d9-es cím adódik. Először ennek a címnek a high byte-ját írjuk a \$0e=14-es VDC regiszterbe. Azután a low byte-ot a \$0f=15-ösbe. Emiatt áll a CHAR,9,9 után a d600-as címen egy \$0f, azaz az a regiszter, amelyikre utoljára hivatkozunk. A \$d601-ben pedig a kurzorcím low byte-ja áll, azaz itt a \$d9. Próbálkozzunk a CHAR,0,3-mal, így a monitorunk segítségével megláthatjuk, hogy a nevezett címekre a „0f 03” került.

Most két kérdés adódik: 1. Hogyan lehet a hibát elkerülni? 2. Más parancsokat is érint-e az a hiba?

Az első kérdéshez. Mégegyszer hangsúlyozzuk, hogy csakis az igen hosszú BASIC programok esetében kell törni a fejünket. Ezt a FRE parancssal ellenőrizhetjük is. Ha FRE (0)>10495, akkor minden rendben van. Ugyanakkor ha grafikákkal dolgozunk, akkor a HiRes grafikát legalább egyszer be kellett kapcsolni. — A továbbiakban a CHAR parancsot a 40 karakteres képernyőn mindenféle megfontolás és korlátozás nélkül alkalmazhatjuk.

Hogy a hosszabb programok esetében is megmaradjon a kurzor kényelmes pozicionálásának lehetősége, három módszert javasolhatunk. Elsőként megváltoztathatjuk a C prekonfigurációs regiszter tartalmát úgy, hogy az ff03 írásakor mindig a 15-ös bank legyen bekapcsolva:

```
BANK 15:POKE DEC ("d503"),0
```

Ez azonban nem igazán egyértelmű megoldás, mert nem lehet tudni, mely BASIC parancsoknál jelentkezik ismét valamilyen hiba. (Egy első kísérletnél ez csodálatra méltóan jól működött.)

A második módszert akkor alkalmazhatjuk, ha nem akarunk lemondani a CHAR parancsról. Levédhetjük a BASIC RAM-ot egy PEEK utasítással:

```
Program elején: D=DEC(„D600”): BANK0:D1=PEEK(D):
D2=PEEK(D+1)
```

```
Minden CHAR felhívásnál: CHAR...:BANK0:POKED, D1:
POKED+1,D2
```

Ez mindig jó, kivéve, ha véletlenül nem egy CHAR parancs farkincája nyúlik bele az érintett tárolókba. Így hát a harmadik megoldás látszik a legtisztábbnak. Minden, a 80 karakteres képernyőre vonatkozó CHAR parancsot egy SYS-szel helyettesítünk:

```
BANK15:SYS DEC(„FFF0”),,x,y: PRINT... vagy
SYS DEC(„C018”),,x,y,: PRINT
```

X és Y CHAR parancs ismert oszlop/sor koordinátái, és a kívánt sztringet a PRINT parancs adja ki.

A legutolsó a gépi kódú programokban az ismert kurzorpozícionáló módszer:

```
lda = $00
sta $FF00; a 15-ös bank bekapcsolva
```

```
...
ldx = oszlop
ldy = sor
clc
jsr $fff0; vagy jsr $c018
```

Az X és az Y regiszterben adjuk át az oszlopot és a sort, és ezt követően felhívjuk a tulajdonképpeni kurzor-PLOT rutint. Előtte a carry flaget feltétlenül törölni kell, aminek oka mindjárt világos lesz.

Mert ott a második kérdés, érint-e más parancsot is ez a dolog? Ténylegesen a \$928D alprogramot ötször hívjuk fel (a monitorunk H parancsával megnézni). Hogy a felhívásokat megértsük, röviden bemutatjuk a kurzorPLOT rutint. Az, ahogy a C128-as esetében megszokhattuk, egy univerzális zseni. Először a \$928D-nél csupán a 14-es bankra kapcsolás történik. Ezt követi a tulajdonképpeni PLOT rutin. Az azonban mindjárt két feladatnak is eleget tesz. Ha a carry flag törölve van, akkor kiolvassa a kurzor aktuális pozícióját, és átadja azt az X és az Y regiszternek. A továbbiakban minden esetben eseti döntés születik attól függően, mely képernyő mód az aktív. Most a következményeket egyenként ellenőrizhetjük.

Ez alapján a BASIC programok fölüírása valóban csak a CHAR utasítás esetében várható.

fölvívás	parancs	hatás
F55AA és	SPC, TAB	Nincs! Itt a kurzorpozíciót csak olvassuk, s eközben csak a nulláslapra hivatkozunk, a VDC-re nem.
F6838	CHAR	Hiba! Amint leírtuk, hivatkozunk a VDC-re.
F6ABF	SCNCLR	Nincs! Itt ugyan beállítjuk a kurzort, de a JSR \$928D-vel csak a 40 karakteres képernyőn.
F84D1	POS	NINCS! Itt a kurzorpozíciót csak olvassuk, (lásd SPC,TAB).

POKE-ok, PEEK-ek et SYS

Először is nézzük meg a restore és a reset billentyűkhöz kapcsolódó vektorok low és high byte-jait. Egy kis elmélet ehhez (haladók ezt átugorhatják): Vektoroknak nevezzük azokat a mutatókat, amelyek meghatározott gépi kódú rutinokra irányulnak. Egy szám high és low byte-ját pedig az alábbi módon számítjuk. Vegyük a decimális 4864-es számot. Ezt átalakítjuk a PRINT HEX\$(4864)-gyel a hexadecimális formára. Ekkor \$1300-at kapunk. Ezt most középen elválasztjuk 13 és 00-ra. Ezeket most visszaalakítjuk decimális számmá a PRINT DEC („13”),DEC(„00”) utasítással. Az eredmény a 19 és a 0 decimális számpáros. A jobb oldali szám mindig a low (alacsonyabb értékű), a bal oldali pedig a high (a magasabb értékű) byte. Azaz: $4864 = 19 * 16 + 0$. Más számítással $high = INT(4864/256)$; $low = 4864 - high * 256$.

1. Gyakorlati alkalmazás:

A 15-ös bankban a 792-es és 793-as decimális cím alatt áll a restore billentyű NMI vektora. Most ezt a vektort tetszőlegesen elhajlíthatjuk. Például:

```
POKE792,0:POKE793,64 A restore hatására bekapcsolási kép.
POKE792,75:POKE793,266 A restore hatására C64-es mód.
POKE792,0:POKE793,224 A restore hatására reset.
```

2. Gyakorlati alkalmazás:

Az 1-es bankban a 65528-as és 65529-es decimális cím alatt áll a reset vektor. Ebbe is belenyúlhatunk. Például:

```
BANK 1:POKE65528,75POKE65529,226 Resetre C64-s mód.
BANK 1:POKE65528,33 Resetre képernyős káosz.
```

Gondoljunk arra, hogy saját gépi kódú rutinok fölvívása éppúgy lehetséges a reset-tel vagy a restore-ral, mint a ROM rutinoké.

Most pedig különböző trükköket adunk, amellyel a saját programjainkat feljavíthatjuk:

1. POKE 820,189 Az ESC az idézőjel módban látható lesz.
2. POKE 808,225 A számítógépünk teljesen lemerevedik.
3. POKE 0,PEEK(0)OR64:POKE 1,PEEK(1)AND191
Átkapcsolunk a DIN billentyűzetre (aki eleddig ezt a POKE 0,PEEK(0)OR64:POKE 1-gyel próbálta volna, az meg kellett, hogy állapítsa, hogy ez a 40 karakteres módban csak egy színnel funkcionál).

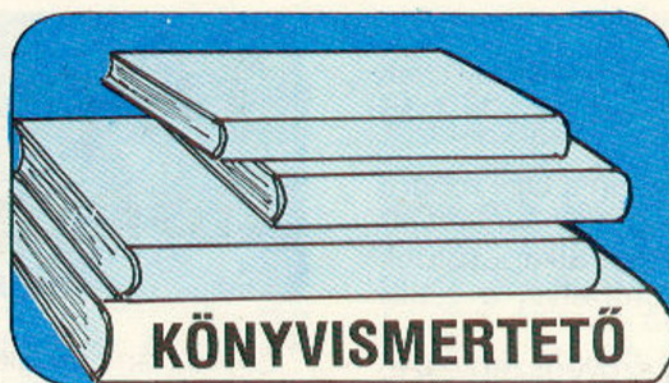
Óri István:

Fedezzük fel...

A könyv a C64-es gépen igen elterjedt EASY SCRIPT, illetve ennek magyar változatát a DELTEX szövegfeldolgozó rendszert kívánja minden alkalmazó számára közérthetően bemutatni. Az 1. fejezet a szövegfeldolgozás általános ismertetését adja, a 2. fejezet példaanyagon keresztül bemutatja a kiválasztott EPSON nyomtatócsalád funkcióinak működtetését BACIS-ből és EASY SCRIPT szövegszerkesztőből. A 3. fejezet az EASY SCRIPT igényesebb felhasználási lehetőségeit tárgyalja, a 4. fejezetben bőséges példaanyagot mutat be, az 5. fejezet pedig egy komplett irat-előállítási és -kezelési rendszert ismertet.

Ára: 190,— Ft

Megjelenés: 1988.



Lukács Sándor:

Túl a hetedhéten — Enterprise

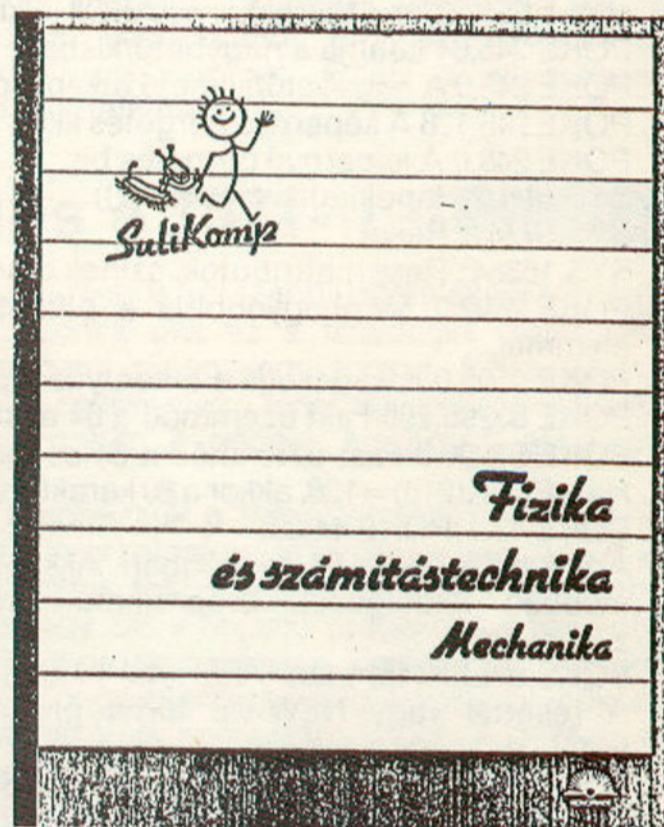
A Hetedhét sorozat szemlélete nem tette lehetővé, hogy egy kötetben bemutassa a sokrétű számítógépet. Ebben a könyvben tehát mindaz benne van, amíg a Hetedhétből kimaradt. A könyv szintén hetekre osztott, tanulásra alkalmas felépítésű. Sőt itt minden hét csak négy nappól áll, tehát jóval több idő marad játékokra, a tulajdonságok kipróbálására.



Dr. Kovács I. (főszerkesztő):

Fizika és számítástechnika — Mechanika

A könyv a gimnáziumok második osztályának fizika anyagát dolgozza fel. A feldolgozás és szemlélet újszerűsége abban mutatkozik, hogy mind az elméleti, mind a gyakorlati problémák tárgyalásánál a számítógépes feldolgozás lehetőségeit is bemutatják a szerzők. A problémák megértése az algoritmikus gondolkodás következetes végigvitelét jelenti az Olvasó számára. A programlisták BASIC nyelven, C64-es és PLUS/4-es gépekre készültek.



ELŐKÉSZÜLETBEN

A Commodore 64-es felhasználói kézikönyv

(Commodore)

A könyv a legújabb verziójú gépek felhasználói kézikönyve. Nélkülözhetetlen minden Commodore 64-es tulajdonos számára. Minden szükséges alapvető információ benne van: üzembe helyezés, BASIC nyelvű programozás. A Függelék a kódtáblázatokat, a hibaüzeneteket és még sok más fontos ismeretet gyűjt egybe.



Tippek—trükkök floppykhoz

Szeretnénk megkönnyíteni a Commodore lemezegekkel való munkát. Most a 1571-es és 1581-es floppyk hibáival kezdjük. Ezen kívül megnagyobbítjuk az 1700-as RAM floppy tárolóját.

A RAM floppy kibővítése

Az 1770-es (128 kbyte-os) RAM bővítés használatakor hamar észre fogjuk venni, hogy a GEOS ezt a plusz tárolót mint RAM floppyt nem tudja használni, mert a hely nem elég. Akkor pedig az ember egy nagyobb bővítés felé kacsingat. Ebben az esetben azonban olcsóbb a meglévő kibővítése mint egy új vétele. Tizenhat 41256-os (120 ns) RAM építőelemre lesz szükségünk.

Az átépítéshez szükségünk lesz egy kis tapasztalatra a páka-kezelést illetően, mivel az eredeti RAM-okat (U—2U17) ki kell forrasztani. Ezek helyére 16 pólusú IC foglalatokat kell beültetni. Ezekbe azután egyszerűen behelyezhetjük az új RAM elemeket (41256, 120 ns). Utoljára meg kell szakítani a J1-es hidat, hogy a számítógép felismerje a kibővített RAM-ot. Ekkor a GEOS képes lesz a tárolóbővítést RAM floppyként használni.

A 1581-es floppy villogása

Kit nem zavarna az a tény, hogy a 1581-es floppynál a piros LED más Commodore lemezegekkel ellentétben hiba esetén nem villog, csak úgy pislog? Ha a világítás épp olyan, elég nehéz felismerni azt. Nézzük hát meg, miért is pislog a dióda ahelyett, hogy ki- és bekapcsolással villogtatnánk.

Ha egy pillantást teszünk a 1581-es kapcsolási rajzra, hamar rájövünk, hogy ezt a LED-et egy ellenálláson keresztül kapcsoljuk a be/kiviteli építőelemhez. Egy másik ellenállás, amely a diódát a +5V-tal köti össze, a felelős azért, hogy a LED állandóan bekapcsolt állapotban van, a piros LED-et tehát nem lehet kioltani. El kell távolítani a fő panelről az R—14-es ellenállást. Mivel ekkor megszakítjuk a +5V-ra vezető ágat, a LED-et ezután szabályosan tudjuk villogtatni. A hibás kezelés jelzését tehát sokkal jobban fel lehet ismerni.

1571-es hiba

Miután a 1571-es meghajtóm hosszabb használat után állandóan hibajelzéseket produkált (pl. READ ERROR 27), szerettem volna alaposan utánajárni a dolognak. A tesztelt lemez nem volt hibás, ezt ki lehetett zárni, azaz csak a hader lehetett az ok. Ki kellett zárni a bepiszkolódott író/olvasó fejet is mint hibaforrást, mert a tisztítás után is jelentkeztek a hibák. A szakműhelyben történt fejbeállítás sem hozta meg a kívánt eredményt. Végezetül csak egy lehetőség maradt, mégpedig a fej magassági beállítása. Az a rugó, amelyik az író/olvasó fejet a lemezre nyomja elgörbülhet, ha a meghajtó hosszabb ideig használaton kívül van, illetve, ha hosszabb ideig lemez nélkül tartjuk lezárva. Ha a rugó elgörbül, akkor nincs meg a szükséges ellenerő a második író/olvasó fejhez. Emiatt a lemezek olvasása nehézségekbe ütközik. Természetesen semmi esetre sem szabad nekiállni kézzel hajlítgatni a felső fejet lefelé nyomó rugót, mivel ezzel akár tönkre is tehetjük a lemezt. Ha valaki ilyen hibával találkozik, a legjobb, ha a készüléket szakműhelybe viszi. Hogy a hiba föllépését megakadályozhassuk, egy egyszerű szabályt kell csak betartanunk:

A lemezeget mindig behelyezett lemezzel zárjuk csak be.

4. POKE 0,47 Ismét visszakapcsol az ASCII módra.
5. POKE 2757,129 A DIN módban a billentyűs bevitel csak nagybetűvel lehetségesek, de a számok beadására is mód van.
6. POKE 2757,128 Letiltja a 80 karakteres módban a DIN módra átkapcsolást. Ennek a POKE-nak van még egy funkciója: Ha a parancsmódban történő bevitel után átkapcsolunk ASCII-ról DIN-re, akkor az A—Z betűk shiftelve, a többi shift nélkül jelenik meg.
7. PEEK(235) Kiadja az aktuális sort (esetleg szövegszerkesztő programokhoz).
9. PEEK(236) Kiadja az aktuális oszlopot.
10. POKE 902,128 Minden parancsra SYNTAX ERROR a válasz.
11. POKE 900,20 Semmilyen parancsnak nincs hatása.
12. SYS 52684,255 Hatására a 80 karakteres képernyőn igen nagyok lesznek a karakterek. A kép az interlace mód miatt az nagyon remeg.
13. SYS 65366 Felhívja a BOOT CALL rutint.
14. POKE 244,0 Kikapcsolja program-/RUN módban az idézőjel üzemmódot (quote módot).
15. Ha a PEEK(204)=128, akkor az ASCII jelkészlet aktív.
16. POKE 245,64 Letiltja a nagybetű/kisbetű átkapcsolást.
17. POKE 245,0 A nagybetű/kisbetű átkapcsolás lehetséges.
18. POKE 248,128 A képernyő görgetés ki.
19. POKE 248,0 A képernyő görgetés be.
20. SYS 51602 Hangkiadás (=chr\$(7)).
21. SYS 57344: Reset.
22. SYS 16384: Reset; attributok, színek maradnak.
23. POKE 2592,2 Megnagyobbítja a billentyűzet puffert húsz elemmel.
24. POKE 2595,0 Kikapcsolja a billentyűzet puffert.
25. POKE 53296,255 Fast üzemmód a 64-es módban.
26. POKE 53296,0 Fast üzemmód a 64-es módban ki.
27. Ha a PEEK(215)=128, akkor a 80 karakteres mód aktív.
28. POKE 43,1:POKE 44,28:
Adjuk be ezeket a 64-es módban. Akkor a korábban a 128-as módban feldolgozott programmal tovább dolgozhatunk a 64-esben!
29. POKE (PEEK(45)+256*PEEK(46)+1),28:SYS DEC „4F4F”
A resettel vagy NEW-val törölt programot visszahozzuk (=OLD).
30. POKE (PEEK(45)+256*PEEK(46)),1:SYS 20303:SYS 20354
Ugyanaz, mint a 29-nél, de biztonságosabb.
31. POKE 2593,1 Pótolja a GETKEY-t.
32. POKE 243,1 Inverz be.
33. POKE 243,0 Inverz ki.
34. A PEEK(6212)-ben áll az éppen a GETKEY-jel lekérdezett billentyű kódja. Ezzel például a Q fölötti 1-est és a tizes blokk egyesét külön kérdezhethetjük le.
35. POKE 208,0:WAIT 208,1 Mint a 31-es pontban.
36. POKE 808,112 A RUN/STOP ki.
37. POKE 808,110 A RUN/STOP be.
38. POKE 208,0 Törli a billentyűzet puffert.
39. SYS 65520,oszlop,sor Helyettesíti a CHAR-t.
40. POKE 53265,27 FAST a 40 karakteres módban.
41. POKE 53265,11 SLOW a 40 karakteres módban.
42. POKE 241,PEEK(241)OR 214: PRINT „szöveg”:
A „szöveg”-et villogtatva adjuk ki.
43. POKE 241,PEEK(241)OR 215: PRINT „szöveg”:
A „szöveg”-et aláhúzva adjuk ki.
44. POKE 241,PEEK(241)OR 216: PRINT „szöveg”:
A „szöveg”-et invertálva adjuk ki.
45. POKE 241,PEEK(241)OR 217: PRINT „szöveg”:
A „szöveg”-et DIN jelkészlettel adjuk ki.
46. A 42—45-ös pontokat kombinálni is lehet. Például:
POKE 241,PEEK(241)OR 214+215:PRINT „szöveg”:
A „szöveg”-et villogtatva és aláhúzva adjuk ki.
47. SYS 57931 GO64 kontroll visszakerdezés nélkül.
48. POKE 981,128 Minden ezt követő SYS parancsot hibásan címezünk.
49. POKE 249,128 Letiltja a <Control G> hangot.
50. SYS 19910 Megfelel egy END-nek, de READY jelzés nélkül.

Biztonságos munka a 1581-gyel

Esetenként a 1581-es lemez egy-egy sávját tönkretesz. A hiba nem mindig jelentkezik, amiért is nehezen sikerült rábukkanni. Egy hibássá tett lemez alapos tanulmányozása után azonban ez sikerült: A sáv (track) egy részét rendben „megírtuk”, egy másik részét azonban egyáltalán nem. Ebből az következik, hogy az íráskor a megfelelő rendszerrutint valami megszakítja. Emiatt a megszakítás miatt azután az ellenőrző összeg számítás nem stimmel. Ha azután megkísérelünk hivatkozni erre a sávra, a lemezegység különböző hibajelzéseket produkál.

Ezt a hibát legalább részben az alábbi lista segít helyrehozni:

```
0 INPUT "a hibás sáv és oldal"; T,S:S+S*20+19
1 OPEN2,8,2 "#":OPEN1,8,15:PRINT#1,"U1 2 0"; T;S
2 PRINT#1,CHR$(T-1)+CHR$(181)+CHR$(S):CLOSE1
```

Egy sáv megjavítása azonban csak tüneti kezelés, mivel ezzel csak az ellenőrző összeget korrigáljuk, a maradék adatok helyreállítása nem történik meg. Az eljárás azt a célt szolgálja, hogy az adott trackre ismét lehessen írással és olvasással hivatkozni. A komplett javításhoz a dolgok gyökeréig kell eljutni.

A 1581-es ROM listájának alaposabb tanulmányozása után sikerült is fellelni a hibahelyet. A hiba oka az, hogy a lemezre írás előtt az interruptot nem tiltjuk le, ami miatt a lemezegység munkája az írás során megszakadhat. A hiba megszüntetésekor azonban egy új probléma merül fel: Az ellenőrző összeg teszt a lemezegység bekapcsolásakor. Ezt a problémát egyszerűen ennek az ellenőrző összeg képző rutinnak az átugrásával kerüjük meg. Most meg lehet változtatni az 1581-es ROM-ban a kívánt részeket anélkül, hogy a lemezegység „fennakadna” ebben a rutinban. Beszúrhatjuk a lemezre írás előtt a interrupt tiltási parancsot. Persze mivel egy ROM-ban nehezen tudnánk átírni bármit is, kell egy új EPROM (27256) és egy EPROM programozó. 8 byte-ot kell megváltoztatni a 1581-es ROM-jában (lásd lentebb).

Egyébként a 1581-esnek van még egy hibája. Ez abban nyilvánul meg, hogy alkalmanként a szektorok megírása nem történik meg] Ezt a hibát még nem találtuk meg. Ki tudna ebben segíteni nekünk?

CÍM	RÉGI	ÚJ	Megjegyzés
\$afc8	\$3a	\$00	Az ellenőrző képzést átugrani.
\$c160	\$6c	\$4c	
\$c161	\$48	\$af	Ugrás a bővítéshez.
\$c162	\$00	\$c5	
\$c5af	\$ff	\$78	Az interrupt letiltása.
\$c5b0	\$ff	\$6c	
\$c5b1	\$ff	\$48	Ugrás az IRQ vektorhoz.
\$c5b2	\$ff	\$00	

CP/M a 1581-gyel

A Commodore Braunschweig kiadott egy hivatalos CP/M-et a 3½ collos lemezegységhez. A hátrány csak az, hogy az nem száz százalékgig kompatibilis a CP/M-hez, mivel egy meglévő RAM bővítő modult nem használhatunk. Most egy egyszerűbb trükkel lehetővé tesszük, hogy kb. 380 kbyte területet használjunk a lemezről a CP/M-mel. A teljes 880 kbyte használata sajnos nem lehetséges, mert ehhez meg kellene változtatni a BIOS-t is. Ezt azonban megnehezíti a dokumentáció hiánya és a létező számos CP/M változatot is. Hogyan legyen mégis CP/M formatált lemezre szert tenni, ha a meghajtót a CP/M alatt nem lehet szóra bírni? A legegyszerűbb módszert most bemutatjuk:

Először kapcsoljuk ki a lemezegységet, és állítsuk a készülékszámot nyolcasra. Ezután a rendes lemezről elindítjuk a CP/M-et. Innen formatáljuk a 3½ collos lemezt. Ha a program a választási lehetőségeivel bejelentkezik, át kell kapcsolni a meghajtót (az indítót ki, a 1581-est be). A behelyezett lemez legyen C64-essel már formatált. Amikor a formatáló program az utolsó tracket (80-as) megírta, az író/olvasó fej hallhatóan messzire megy. Ebben a pillanatban ki kell kapcsolni a lemezegységet. Ezután állítsuk át az egységyszámot 9-re, majd boot-oljuk a CP/M-et. Miután az bejelentkezik, adjuk be az alábbi utasítás: b:dir. Ha mindent rendben csináltunk, a képernyőn megjelenik a meghajtójelölés négy „tabulált”, azaz egymástól adott távolságra elhelyezett kettősponttal. Ha valami BDOS hiba jelentkezik, akkor az azt jelenti, hogy a meghajtót túl későn kapcsoltuk ki. Ebben az esetben meg kell ismételni a formatálást. Ha nincs hibajelzés, akkor törölni kell a lemezt az ERASE*. * utasítással. Ha most behívjuk a directoryt, a NO FILE jelentést kapjuk. Most már majdnem kész is a lemez. Probléma jelentkezhet azonban az első PIP-es másolásnál, mivel a szektorok részben foglalt jelzéssel rendelkeznek. Ebben az esetben a programot újra kell másolni. Hogy ne kelljen mindig ezt a bonyolult procedurát végrehajtani, a lemezt másolhatjuk a „1581 Disk-Backup” programmal, amely a lemezegység teszt) demolemezén megtalálható.

Hőfejlődés a 1571-nél

Ismeretes, hogy az 1571-es a hosszabb üzemelésnél meglehetősen felmelegszik. Ennek oka az a felesleges hőtermelés, amelyet a floppy hálózati tápegysége termel. Ezt a veszteségi teljesítményt azonban csökkenteni lehet egy egyszerű trükkel. Az átépítés során azonban a gépnek kikapcsolt állapotban kell lennie!

Ha alaposabban megnézzük a transzformátort, észrevehetjük, hogy a bemenő (primer) oldalon két kivezetés van. A 1571-est ugyanis 220 és 240 Voltos feszültséggel is üzemeltetni lehet. Nálunk a kék vezeték egy 220 V feliratú csatlakozóra van kötve. Így a tápegység 220 V-os feszültségen üzemel. A kísérletek során azt tapasztaltuk, hogy a lemezegység akkoris jól működik, ha ezt a vezetékét a 240 V-os bemenetre kötjük. A 1571-es így nem melegszik úgy fel mint korábban. Ez meghosszabbítja az alkatrészek élettartamát, amelyek nehezen viselik el a meleget.

Megjegyzés:

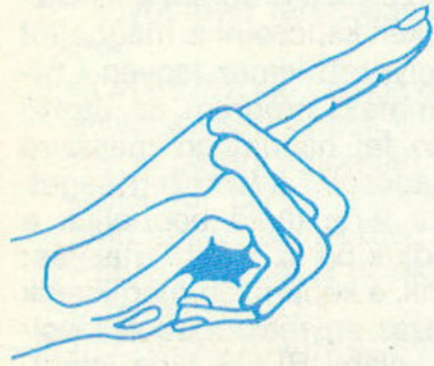
Ez a trükk az NSZK-ban bizonyosan jól funkcionál, ott ugyanis az egyesített nyugateurópai hálózat frekvencia- és feszültség-hálózatán mindenhol pont 220 V 50 Hz-es váltóárammal számolhatunk. Ezt úgy is tapasztaltam, hogy egy kint vásárolt olcsó, stabilizálatlan hálózati kvarcóra ott egyetlen percet sem késett vagy sietett, míg idehaza a napi 8 perces késés/sietés sem ritka. Az egzakt 220 V 50 Hz nálunk még vágyálom. Márpedig, ha a tápegységet 240 V-ra „kötjük”, s a feszültség mondjuk csak 210 V, a lemezegységünk meg fog (rendesen) dolgozni, mert kívül kerülnék a tolerancia határon!

Joker a 1581-nél

Amint arra véletlenül rájöttem, a 3½ collos meghajtónál a joker (*) funkciót más Commodore floppykhoz képest kibővítették. Ha egy régebbi floppynál (1541/70/71) a LOAD után állománynév mintának pl. a „\$:a*.pic” sztringet adjuk meg, a gép minden file-t megtalál, amely „a”-val kezdődik. Nincs értelme tehát ezeknél a csillag után más karaktereket írni. Másképp van ez a 1581-es esetében. Ha ott ugyanis pl. a „\$:*.t”-t adjuk meg, akkor azokat a bejegyzéseket kapjuk vissza, amelyek „a”-val kezdődnek, és „t”-re végződnek. Ezzel a módszerrel könnyű megtalálni a lemezen az azonos végződésű állományokat. Ez számos esetben hasznos, ahol a file végződésének szerepe van (szövegszerkesztők, grafikus programok, stb.).



kedvezmények



A NOVOTRADE RT. 2C
Áruházában az Egyesület
PLUSZ-, DEÁK- és
SZUPER PÁHOLYÁNAK
tagjai kedvezményel
vásárolhatják meg a
következő programokat:

KEDVEZMÉNYEK a PC-Szalontól a Szuperpáholy tagok részére!

IMPOSSIBLE MISSION

II. 15% 781,— ÁFÁVAL
játékprogram

AMBASSADOR PC szövegszerkesztő
10% 5950,— ÁFÁVAL



áprilisi
60 forintos
vásárlási utalvány

Beváltható készpénzes
vásárlás esetén az
APISZ szaküzleteiben
XI., Budafoki út 7.
VIII., Szigony u. 15.
Érvényes: 1989. július 31-ig



áprilisi
60 forintos
vásárlási utalvány

Beváltható készpénzes
vásárlás esetén a 2C
áruházban XIII., Balzac u. 35.
és a Művelt Nép
vidéki boltjaiban működő 2C sarokban.
Érvényes: 1989. május 31-ig

Az alábbi Data—Becker kiadványokat 20%-kal:

	Jelenlegi ár	—20%
Tudomány és technika	215,—	172,—
Robotok és a Commodore 64	249,—	199,20
Számítógép és sakk	390,—	312,—
Gépikódú programozás haladóknak	319,—	255,20
Az EPSON nyomtatók könyve	290,—	232,—
COMPILER	298,—	238,40
A 68000 mikroprocesszor	349,—	279,20
Sulikomp matematika	149,—	119,20
Sulikomp történelem	99,—	79,20
Videoton operációs rendszer	190,—	152,—

Az alábbi termékeket 10%-kal:

		—10%
Basic Puli	625,—	562,50
Szint billentyű	7618,—	6856,20

Az alábbi szoftvereket 20%-kal:

DIÉTMASTER:

A program a diabetes kezelést segíti elő. Oktató és diéta összeállító részei hasznosak lehetnek mind a cukorbetegnek, mind a velük foglalkozó egészségügyi dolgozók számára.

régi ár: 9808,— új ár: 7846,40

ENERGIAMÉRLEG:

Energetikusok napi munkáját segítő program. régi ár: 11 648,— új ár: 9318,40

FORMA 42:

A program lehetőséget nyújt C-64 kompatibilis mágnes lemezegységekhez használt 5,25"-os lemezek, adattárolók kapacitásának növelésére.

A lemezkezelés 42 sávon valósul meg, szemben az eddig ismert 35 sáv használatával.

régi ár: 1500,— új ár: 1200,—

FORTH:

Feltételezve a Forth nyelv lényegi ismeretét, ez a program egy láncolt kódú interpreter nyelv.

régi ár: 11 648,— új ár: 9318,40

A Newline számítástechnikai vállalkozás 10% kedvezményt ad az egyesület tagjainak:

C 16 beépíthető 64 KByte memóriabővítő

16-64-es átkapcsoló	1990,— Ft
beépítés munkadíja	150,— Ft
ROMTURBO 16	490,— Ft
együttes megrendelés esetén	770,— Ft
árengedménnyel:	3400,— Ft
	3060,— Ft

Jogosultak: a Plusz- és a Szuperpáholy tagjai
Igazolás: ennek a tikknek postai elküldésével
Cím: Newline, 1014 Budapest, Tárnok u. 26. 1/5.

NEWLINE

HARDWARE SOFTWARE

A Fotoelektronik-Novotrade-Alfa KFT az alább felsorolt szervizeiben mindenféle szervizszolgáltatás munkadíjából és az 1989. évben megkötött átalánydíjas szerződés összegéből 10% kedvezményt ad az egyesületi tagoknak.

Határidő: nincs, illetve átalánydíjas szerződésnél 1989. 12. 01.

A kedvezményt nyújtó szervizeink:

1053 Budapest, Magyar u. 12-14.	Telefon: 173-551
1083 Budapest, Szigony u. 9.	Telefon: 343-153
1191 Budapest, Gábor Á. sétány 3.	Telefon: 274-763
3100 Salgótarján, Arany J. u. 3.	Telefon: 32-14-007
3525 Miskolc, Fazekas u. 1-3.	Telefon: 46-17-011
4034 Debrecen, Holló L. u. 14.	Telefon: 52-32-863
5600 Békéscsaba, Bartók B. u. 37.	Telefon: 66-27-195
6726 Szeged, Csongrádi sugárút 76.	Telefon: 62-13-377
7400 Kaposvár, Füredi u. 24.	Telefon: 82-16-307
7624 Pécs, Jurisics M. u. 17.	Telefon: 72-11-812
8000 Székesfehérvár, Széchenyi u. 15/a.	Telefon: 22-12-711
9024 Győr, Bem J. tér 1.	Telefon: 96-12-802
9700 Szombathely, Szalonok u. 31.	Telefon: 94-13-419

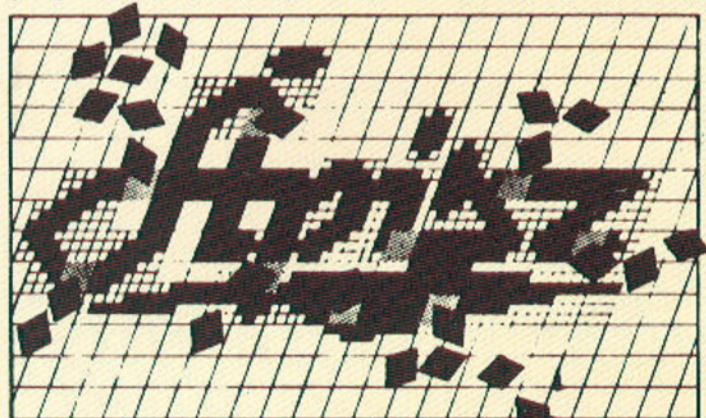
Igazolás: a javítandó berendezés leadásakor egyesületi igazolvánnyal.
A kedvezmény többször is igénybe vehető.



SPECIÁLIS IGÉNY – SPECIÁLIS SZAKÜZLET



S Z Á M I T Á S T E C H N I K A



S Z Á M I T Á S T E C H N I K A

**Számítástechnikai
szaküzleteink várják
a lakosság és a vállalatok
vásárlásait. Az igényeket
azonnal és megrendelés
alapján is teljesítik.**

Budapest XI. Budafoki út 7.

Telefon: 665-503

Budapest VIII. Szigony u. 15.

Telefon: 143-446



MONORADE 2C

DISK-OS C-64

A DISK-OS rendszert programfejlesztőknek ajánljuk, numerikus feladatok megoldására. Kész matematikai programok segítségével számolhatunk, valamint a függvényeket grafikusán is ábrázolhatjuk. A DISK-OS rendszer a felhasználó igényeinek megfelelő módon építhető fel a rendszert tartalmazó négy lemezoldalról. Ezeken a lemezeken a rendszer működéséhez szükséges file-ok könnyen áttekinthető könyvtárakba vannak rendezve.

A program ára: 2112 Ft.

LEMEZDOKTOR C+/4

A program lehetővé teszi a lemezműveletek közvetlen használatát, a katalógus kiírását, a foglaltsági térkép felrajzolását, valamint a lemez tetszőleges blokkjában történő felrajzolást, ilyen módon a katalógusba bejegyzett FILE-okat védettnek (törölhetetlennek) nyilvánítja, ezen FILE-ok törlése, a katalógusból törölt, de felül nem írt FILE-ok visszajegyzése. Jól felhasználható ezáltal a „véletlen” törlések korrigálására, valamint minden olyan módosításra a lemezen, amit a számítógép alap helyzetben nem tud.

A program ára: 1165 Ft.

RAKTÁRKEZELŐ PROGRAM: C-64

A számlakezelő programokhoz kapcsolódó raktárkezelő rendszer lehetővé teszi a raktáron lévő termékek adatbázisszerű kezelését, a változások folyamatos nyilvántartását, a számlázásoknak a készlettel való összekapcsolását. A program a készletállományról és a forgalomról nyomtatott listát készít. Alapkiépítésben a raktári állomány 700 termékből állhat.

A program ára: 7500 Ft.