



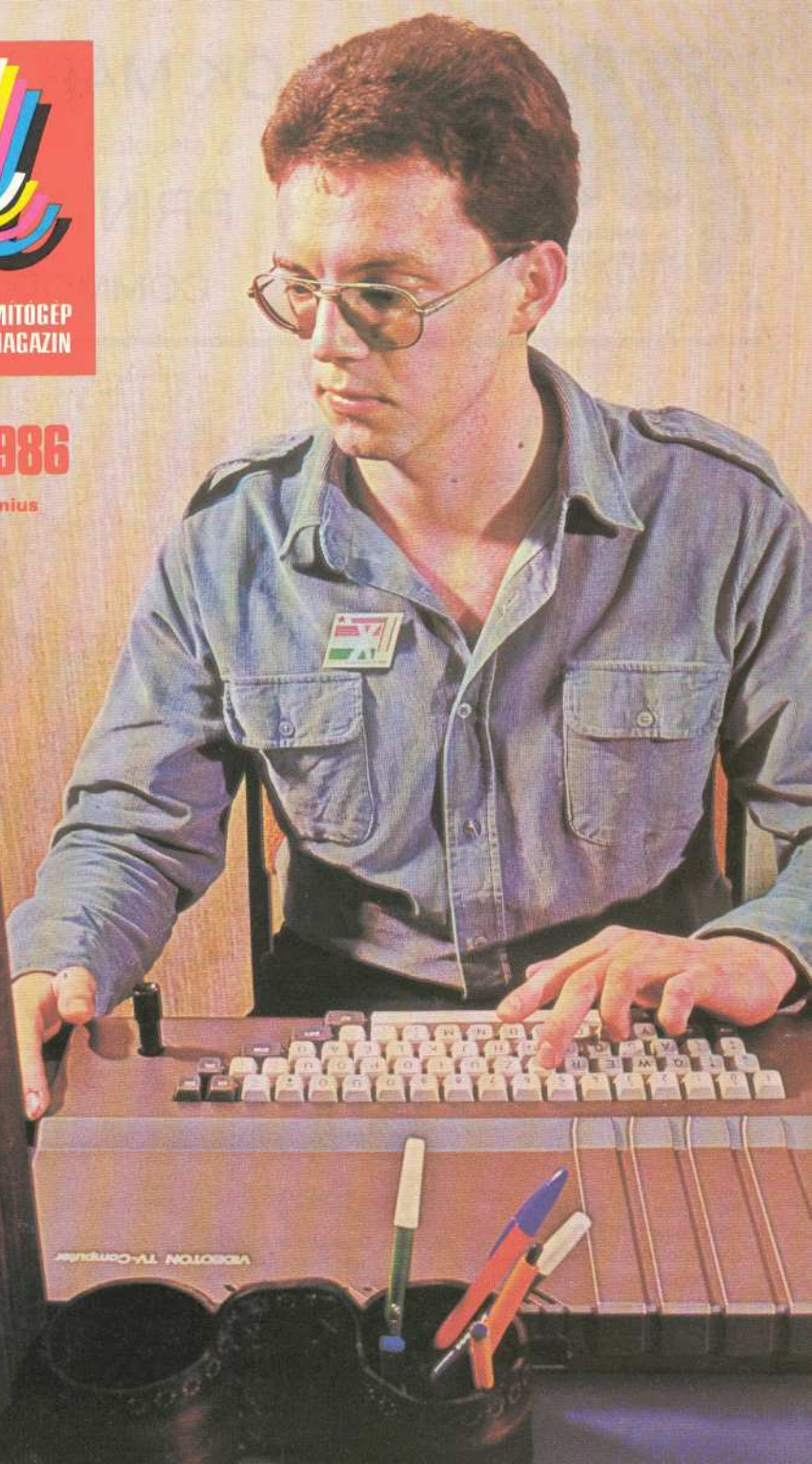
MIKROSZÁMÍTÓGÉP
MAGAZIN

1986

június

Ára: 30 Ft

A NEUMANN JÁNOS SZÁMÍTÓGÉP TUDOMÁNYI TÁRSASÁG LAPJA





A MIKROSZÁMÍTÓGÉP
MAGAZIN AJÁNLATA!

ŐK MÁR TUDJÁK!

a megoldás:

PRINTER BASIC

COMMODORE-64-re



A PRINTER BASIC a következő
nyomtatók kezelését támogatja:

- Seikhosa GP 100/A
- Seikhosa GP 100 VC
- Commodore VC-1515
- Commodore VC-1525
- Commodore MPS-801
- Commodore MPS-803

Forgalmazza:
NOVOTRADE Rt. 2C Számítástechnikai
Áruház

Bp. XIII., Balzac u. 35.
Pécs, Zrínyi Miklós Könyvesbolt,

Jókai u. 25.
FOTOELEKTRONIK Ipari Szövetkezet
szakboltjai

Bp. V., Múzeum krt. 19
Bp. XIII., Rajk László u. 46/b.
Miskolc, Korvin Ottó u. 5.
Debrecen, Szabó István alb. tér 6.
Győr, Bem tér 1.

HARDSOFT Számítástechnikai Gazda-
sági Munkaközösség

Százhalombatta, Rózsa Ferenc u. 32.
- Iskolák itt 50%-os árkedvezmény-
nyel rendelhetik meg a programot.
A program ára 7000,- Ft és 8000,- Ft
között mozoghat (az egyes forgalma-
zók árai között).



**MIKROSZÁMÍTÓGÉP
MAGAZIN**

A NEUMANN JÁNOS SZÁMÍTÓGÉP- TUDOMÁNYI TÁRSASÁG LAPJA

**A kiadvány
a Tudományos-vezetési
és Informatikai
Intézettel
együttműködve készül**

**A szerkesztőbizottság
vezetője:
Kovács Győző**

**E számunkat
szerkesztették:
Bakos Tamás
(programozástechnika)**

**Broczkó Péter
(hírek)**

**Kovács Győző
(levelezés)**

**Lindner László
(sakkprogramozás)**

**Petróczy Judit
(könyvek)**

**Simonyi Endre
(klub)**

**Vadkerti János
(µprogramok)**

**Varga András
(iskola — számítógép)**

**A szerkesztőség
munkatársa:
Kardos Zsuzsa**

**Felelős szerkesztő:
Könyves Tóth Pál
Szerkesztőség:
1027 Budapest II., Fő u. 68.
Telefon: 154-250**

**Kiadja a Delta Szaklapkiadó
és Műszaki Szolgáltató
Leányvállalat
Felelős kiadó:
Faklen Pál igazgató
1442 Budapest VII., Garay u. 5.
Telefon: 415-583, 215-440**

**Terjeszti a Magyar Posta
Előfizethető
bármely postahivatalban,
a kézbesítőknél,
a Posta hírlapüzleteiben
és a Hírlapelőfizetési
és Lapellátási Irodánál
(Budapest V., József nádor tér 1.
Postacím: 1900 Budapest)
közvetlenül
vagy postautalványon,
valamint átutalással
a HELIR 215—96162
pénzforgalmi jelzőszámmal.
Megjelenik havonta
Példányonkénti ára 30,— Ft
Előfizetési díj:
egy évre 360,— Ft
fél évre 180,— Ft**



**Szikra Lapnyomda
Budapest (86-3203)
Felelős vezető:
Csördes Zoltán vezérigazgató**

**INDEX: 25629
ISSN 0236-6088**

**Címképünk:
A VIDEOTON legújabb
számítástechnikai
terméke
a TV-COMPUTER**



Tartalom

Az ifjúság kongresszusa	2
Hogyan tovább?	3
A minőségügy közügy	12
Assemblerek, cross-assemblerek	14
Adok — veszek — cserélek	16
Mikroszámítógépek a tavaszi fesztiválon	24
Programozás profi szinten	26
Mikroklub → mikrovallokozás	27
A klub-vitáról	27

ISKOLA — SZÁMÍTÓGÉP

Gép a gépen IV.	4
A tanárok „nyílt végű” programokat keresnek	5
Hardcopy C16-hoz	6
Orosz igeragozás	7

PROGRAMOZÁSTECHNIKA

Hattyúdál	8
BASIC és gépi kód	13

PIAC

	16
--	----

JÁTÉKPROGRAMOK

	17
--	----

TERMÉKISMERTETŐ

A Videoton TV-Computer	20
------------------------	----

DIÁKROVAT

Ékezetes ábécé C64-re	28
PTA-4000 programok	30
A számítógép anatómiája	31

µKLUB

Méréstechnikai alkalmazások	32
PTA-4000. A hardver további részei I.	35
VC20-használóknak	36
Ki ad magyarázatot?	37
Külső tápegység ZX81-hez	38
Hallható programok a Datasette-en	38

AZ OLVASÓ IRJA

	40
--	----

SAKKPROGRAMOZÁS

Bitek és figurák	42
------------------	----

KÖNYVEK

	44
--	----

HÍREK, ÉRDEKESSÉGEK

	47
--	----

Gép a gépen IV.

Befejező részünkben egy kissé hosszabb program közülnk, kevesebb magyarázó szöveggel. Itt már valóban csak leartjuk annak természetét, amit a sok szubrutin szervezésénél kidolgoztunk. A program irásokor elé a részt már szinte soronként teszteljük. Még csak egy dolgot jegyünk meg: az 1580-as utasítás miatt az F változó értékét állítsuk be. Például: F = -1.

```

1600 REM MUKODES OSSZEALLITASA FUNKCIONA
LIS EGYBEGEKBOL
1610 REM *****
*****
1620 REM A PROC DEKODER ES MEMORIA A HAR
OM DOBOZ SZERELME
1630 A(8)=15:A(9)=15:A(10)=15
1640 GOSUB930:X%= "A SZAMITOGEP LEGFONTOS
ABB EGYSEGEI":Y%= "A PROCESSZOR, A
CIM DEKODOLO ES A MEMORIA":Z%= "GOSUB1
550:GOSUB1500
1650 X%= "A PROCESSZORBOL KIVEZETO FONTOS
ABB VONALAKAT MUTATJUK MEG.":Y%= "Z%= "
GOSUB1550
1660 Y%= "AZ ADDRESS BUS 16 BITES VONAL.
65536 MEMORIA BYTE-OT TUD MEGCI- MEZNI."
GOSUB1550:GOSUB990:GOSUB1500
1670 Z%= "VAGY":Y%= "A DATA BUS 8 BITES VO
NAL. AZ ADATOKAT SZALLITJA A MEMORIAKBA.
":T=0:GOSUB1550:GOSUB1060:T=0:GOSUB1120:
GOSUB1500
1680 T=-1:GOSUB1060:GOSUB1120:Z%= "A MEMO
RIAKBOL A PROCESSZORBA.":GOSUB1560:T=0:G
OSUB1180:T=0:GOSUB1250:GOSUB1560:GOSUB15
80
1690 REM A VEZERLO VONALAK
1700 X%= "OT LENYEGES VEZERLO VONALAT M
UTATUNK BE":Y%= "Z%= "GOSUB1550:T=0:GOS
UB1300:GOSUB1310:GOSUB1320:GOSUB1330:GOSU
B1340
1710 T=-1:GOSUB1300:GOSUB1310:GOSUB1320:
GOSUB1330:GOSUB1340:Y%= "M1 VONAL JELZI,
HOGY A PROCESSZOR UTASITAS LEHIVAS CKLU
SBAN VAN.":T=0:GOSUB1550:GOSUB1300:GD
SUB1500
1720 Y%= "MREG VONAL JELZI, HOGY AZ ADRE
SS BUS-ON KIADOTT CIM ERVENYES.":GOSUB15
50:GOSUB1310:GOSUB1500
1730 Y%= "RD VONAL JELZI, HOGY A PROCESSZ
OR AZ ADAT BELOVASASARA KESZ.":GOSUB1550
:GOSUB1320:GOSUB1500
1740 Y%= "WR VONAL JELZI, HOGY AZ ADAT VO
NALAKON ERVENYES ADAT VAN.":GOSUB1550:GD
SUB1330:GOSUB1500
1750 Y%= "RFSH A DINAMIKUS MEMORIAK FRISS
ITESET VEGZI.":GOSUB1550:GOSUB1340:GOSUB
1500
1760 REM *****
*****
1770 A(8)=0:A(9)=0:A(10)=0:GOSUB970
1780 X%= "VIZSGALJUK MEG EGY MEMORIA BYTE
FELEPITESETI":Y%= "Z%= "GOSUB1550:G
OSUB1500:GOSUB1300
1790 X%= "EGY BYTE EGY MEMORIA EGYSEG, AM
ELY 8 BIT INFORMACIOT TARTALMAZ.":Y%= "A
BIT AZ INFORMACIO LEKISEBB EGYSEGE."
1800 Z%= "KET LEHETOSEG KOZUL AZ EGYIKET
KIVALASZTJA.":GOSUB1550:GOSUB1500
1810 X%= "IGY EGY BYTE 2-NEK B. HATVANYA
AZAZ 256 LEHETOSEG KOZUL VALASZT":Y%= "EG
YET, TEHAT EKKORA SZAMOT TARTOLHAT.":Z%= "
GOSUB1550:GOSUB1500
1820 X%= "EGY MEMORIA BYTE B OLYAN ARAMK
RT TARTALMAZ AMELYNEK KET ALLA-":Y%= "POT
A VAN.":GOSUB1550:GOSUB1500
1830 X%= "MINDEN BYTE KAPCSOLATBAN VAN:":
Y%= "A DEKODOLOVAL A ----":VEZETEKE
N KERESZTUL.":Z%= "ENNEK AKTIV ALLAPOTA J
ELENTI REKESZ MEGCIZESET."
1840 GOSUB1550:GOSUB1500
1850 X%= "AZ RD ES WR VONALAK SZEREPE MAR

```

```

ISMERT.":Y%= "A FUGOLEGESEN RAJZOLT VEZE
TEK A DATA BUS VONALAK.":Z%= "GOSUB15
50:GOSUB1500
1860 X%= "MEGVIZSGALUNK EGY MEMORIA IRASI
S EGY OLVASASI VALAMINT EGY":Y%= "UTASITA
S BEHAVASI /FETCH/ CKLUST.":GOSUB1550:G
OSUB1500:CLS
1870 REM CKLUSOK VIZSGALATA
1880 REM *****
*****
1890 A(8)=15:A(9)=15:A(10)=15:GOSUB970
1900 CLS:REM IRAS MEMORIABA
1910 T=0:GOSUB930:X%= "IRAS A MEMORIABA":
Y%= "I":Z%= "GOSUB1550:GOSUB990:GOSUB15
60:Y%=Y+CHR$(181):GOSUB1560:T=0:GOSUB13
10
1920 Y%=I+I2*:GOSUB1060:T=0:GOSUB1120:
GOSUB1560:Y%=Y+I1*:GOSUB1560:C=Y
1930 Y%=Y+CHR$(181):T=0:GOSUB1560:GOSUB
1330
1940 Y%=C+I2*:GOSUB1560:Y%=Y+I1:C=Y
GOSUB1560
1950 Y%=Y+CHR$(181):GOSUB1560:T=-1:GOSU
B1310:GOSUB1330:Y%=C+I2*:GOSUB1560
1960 Y%=Y+CHR$(151):GOSUB1560:T=-1:GOSU
B1860:GOSUB1120:PRINT944+19," ";
":GOSUB990:PRINT64+19," ";
":GOSUB1500
1970 REM *****
*****
1980 REM *****
*****
1990 REM OLVASAS MEMORIABOL
2000 X%= "OLVASAS MEMORIABOL":Y%= "I":Z%= "
":CLS:GOSUB930:GOSUB1550:GOSUB990:C=Y:
Y%=Y+CHR$(181):GOSUB1560:T=0:GOSUB1310:
GOSUB1320
2010 Y%=C+I2*:GOSUB1560:Y%=Y+I1:GOSUB
1560:T=0:GOSUB1180:T=0:GOSUB1250
2020 Y%=Y+I2*:GOSUB1560:Y%=Y+I1:Z%= "M
INTAVETEL TORKENIK A DATA VONALAKROL.":G
=-1:GOSUB1560:T=0:GOSUB1250:G=0
2030 C=Y:Y%=Y+CHR$(181):Z%= "GOSUB15
50:T=-1:GOSUB1310:GOSUB1320:Y%=C+I2:I=
-1:GOSUB1180:GOSUB1250:PRINT64+17,"
";
":GOSUB1500
2040 Y%=Y+CHR$(151):GOSUB1560:T=-1:GOSU
B990:PRINT64+19," ";
":GOSUB1500
2050 GOSUB1500
2060 REM *****
*****
2070 REM UTASITAS ELERES
2080 CLS:X%= "UTASITAS ELERES /FETCH/ CIK
LUS:":Y%=CHR$(151):GOSUB930:Z%= "GOSUB1
560:T=0:GOSUB1300:GOSUB990
2090 Z%= "A CIMBUSZON A PROGRAMSZAMLAO T
ARTALMA.":Y%= "I":GOSUB1560
2100 Y%=Y+CHR$(181):GOSUB1550:T=0:GOSUB
1310:GOSUB1320:Y%=I+I2*:GOSUB1560:Y%=Y
+I1*:GOSUB1560:T=0:GOSUB1180:T=0:GOSUB1
250
2110 Y%=Y+I2*:T=0:Z%= "MINTAVETEL A DATA
BUSZ TARTALMABOL.":G=-1:GOSUB1550:GOSUB
1250:C=Y:Y%=Y+CHR$(151):Z%= "G=0
2120 GOSUB1550:T=-1:GOSUB1180:GOSUB1250:
PRINT944+19," ";
":GOSUB1300:GOS
UB1310:GOSUB1320:GOSUB990:T=0:GOSUB990:Z
%= "A CIMBUSZON FRISSITESI CIM VAN.":GOSU
B1560
2130 T=0:GOSUB1340:Y%=C+I1:GOSUB1560:C
=Y:Y%=Y+CHR$(181):GOSUB1560:T=0:GOSUB
1310
2140 Y%=C+I2*:GOSUB1560:Y%=Y+I1:GOSUB
1560:C=Y:Y%=Y+CHR$(181):T=-1:GOSUB156
0:GOSUB1310:Y%=C+I2*:GOSUB1560
2150 Y%=Y+CHR$(151):Z%= "GOSUB1550:T=-
1:GOSUB990:PRINT64+19," ";
":GOS
UB1340
2160 GOSUB1500
2170 GOTO1600

```

Reméljük, ezzel a programmal hozzájárultunk ahhoz, hogy az iskolákban egyre több diák értsen meg a mikroszámítógép működését, és egyre többen iranjak jól működő, ügyes programokat. Talán ahhoz is kedvet kap valaki, hogy tovább építse a programot, bemutatva a Z80-as processzor megszakitási rendszerét, amely szemléltetné, hogy a számítógép hogyan vezérel külső berendezéseket.

NYIRATI LÁSZLÓ

A tanárok „nyílt végű” programokat keresnek

Ez derült ki a Londonban rendezett High-Tech and Computers in Education (Fejlett technika és számítógépek az oktatásban) című kiállításán. A hagyományos oktatási segédeszközöket, a diavetítőtől a kísérleti mérőműszerekig ugyanúgy számítógép vezérli, mint a legfejlettebb technikai csodákat: a mindentudó iskolarobotot, a rajzdigitalizálót, a lézerlemezeket.

Az iskolai felhasználásban egyre inkább az „összekapcsolt gépek elve” érvényesül: sokszorozzuk meg az amúgy is ügyes technikai segédeszközök teljesítőképességét — azáltal, hogy hálózatba kapcsolva működtetjük őket vagy úgy, hogy különböző célú eszközöket kapcsolunk egyetlen, hatékony rendszerre.

A legtöbb érdeklődőt egyértelműen a számítógép-vezérlésű, mozgó, igazi gépeket tökéletesen utánozó LEGO-játékok vonzották. Ezek a mozgatható játékszerek tökéletesen megfelelnek a játszva tanítás egyre népszerűbb elvének: miközben a gyerek kedvenc játékból, a LEGO-ból dömpert, közlekedési lámpát, darut vagy éppen sárga fűnyírót épít, megismerkedik az adott eszköz felépítésével, működési mechanizmusával, irányítási lehetőségeivel, s — minthogy a vezérlést maga programozhatja be —, a programozás néhány alapjával is. Boldog, mert a felnőtt világ eszközeinek kicsinyített mása nemcsak testet ölt a kezében, de mozog, működik, pontosan úgy, ahogy az igazi. S miközben a gyerek belefeledkezik a



játék gyönyörűségébe, észrevétlenül olyan ismereteket szerez a technikai világról, amelyek ma már nélkülözhetetlenek.

Szinte korlátlan lehetőségeket rejt magában a képlemezen rögzített oktatófilmtrá: a képernyőn megjelentetett katalógusból kijelöljük a kívánt demonstrációs filmet, a számítógép megkeresi a képlemezen a kiválasztott címet, s máris indul a film. Ha például az ornitológiai oktatócsomag búsbanka címszavára visszkurizt, néhány másodperc múlva perog a képernyőn a madár képét, hangját, lakhelyét bemutató, szövegmagyarázattal kiegészített, tökéletes kép- és hangminőségű oktatófilm.

A műszakirajz-tanítást és a mérnöki tervezőmunkát könnyítik meg azok a számítógép-vezérlésű rajzolóeszközök, amelyek néhány megfelelő adat és egyszerű utasítás bebillentyűzése után a legbonyolultabb mérnöki rajzot is képesek megjeleníteni, majd kinyomtatni. A másik gyakori megoldás, amikor a számítógéphez csatlakoztatott eszközön a számítógép által megtervezett ábra a szemünk láttára, fázisonként rajzolódik ki a papíron. A számítógéphez csatolt digitalizáló letapogatás vagy képi rögzítés útján szereznek információt az ábrázolandó tárgyról, a felvett jelet digitális jelle átalakítva, bármilyen ábrát képesek megjeleníteni képernyőn vagy kinyomtatott formában.

Bár a kiállítás — mint erre a címe is utal — elsősorban hardvercentrikus volt, tehát a legfejlettebb technikai eszközök oktatási alkalmazását mutatta be, említést érdemelnek a legújabb oktatászoftverek is.

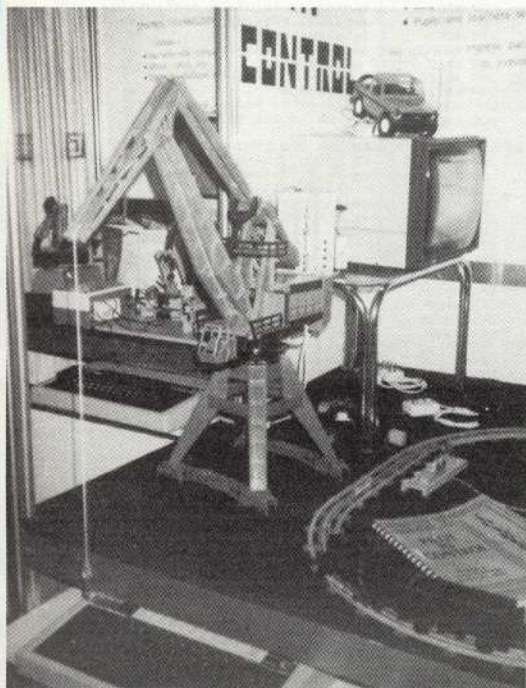
Csak az a program tűnt életképesnek ezen a piacon, amely interaktív, vagyis beavatkozási

lehetőséget nyújt — és nemcsak a diák, hanem a tanár számára is. A tanárok a „nyílt végű” programokat keresik, amelyek csak keretet adnak, s a végleges programot ők tölthetik fel az adott témakör, illetve saját ízlésük, igényeik szerint. A legfőbb alapelv a játszva tanítás, vagyis szemléletes, egzak, mégis szórakoztató módon közölni ismereteket a tanulókkal, s ily módon nevelni őket kreatív gondolkodásra.

Az egyik legjobb példa e fel fogás illusztrálására Dienes Zoltán Pál, a világhírű, magyar származású matematikus egész életműve. Ő már a számítógépek korát jóval megelőzve azt tűzte ki céljává, hogy játékos eszközökkel kedveltesse meg a matematikát. Alapelve: az elvont tételek tényszerű közlése helyett kísérleteztetni a gyereket. Helyezük valamilyen szituációba, csupán a legszükségesebb rendező elveket közöljük vele, s hagyjuk játszani. Rövidesen ő maga jön rá a törvényszerűségekre, s a saját felfedezés öröme miatt ezek mélyen, életre szólóan rögződnek benne.

Dienes professzor — aki nyolcvanadik életévéhez közeledve tanult meg programozni — személyes jelenlétével tisztelte meg a kiállítást. Bemutatta saját, BBC-n, az angol iskola-számítógépen készült programjait, s felhatalmazta a Novotrade Rt-t, hogy magyarországi gépekre írja át és forgalmazza. Hozzájárult ahhoz is, hogy az ő ötletei nyomán szegedi programozók által készített logikai játékprogramok forgalomba kerüljenek. Rövidesen megjelennek a piacon a Dienes-játékok; aki többet szeretne tudni róluk, figyelje az Iskolatelevízió Dienes-sorozatát.

MIKLÓS KATALIN



PRIMO

OROSZ IGERAGOZÁS

A gép a képernyőre kiírja az orosz szónak azt a részét, amelyet raggal kell kiegészíteni. A rag beírása után a gép ellenőrzi a választ, az elkövetett hibákat számolja és hanggal jelzi. A képernyőn minden esetben a helyesen ragozott szó jelenik meg, természetesen cirill betűkkel.

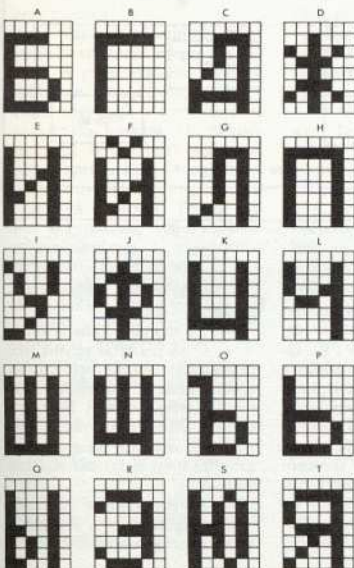
A cirill betűket a következő módon kell beírni. A „jo” betű helyére „E” betű, a „3” betű helyére a „3” szám kerül. Azokat a betűket, amelyeket a gép billentyűről nem lehet beírni, a képernyőn látható táblázat segítségével, kódolva kell megadni.

A program működéséhez a számítógép bemutató kazettáján levő „Karakter” programra is szükség van. A programot a következők sorrendben kell elkészíteni. Először az *ábrán* látható karaktereket készítjük el a „Karakter” program segítségével. Ezután az új karaktereket tartalmazó programot egészítjük ki az itt látható programmal, majd rögzítjük szalagra a kész programot a „Karakter” programban leírt utasítás szerint. A program a LIST 2 paranccsal listázható.

Új szavakat az 5000-es sortól a 6000-es sorig DATA utasítással, kódolt formában adhatunk meg. A szó kódja után levő szám a ragok kódját jelzi. Ezek a következők:

1. -ju, -es, -et, -em, -ete, -jut
2. -u, -es, -et, -em, -ete, -ut
3. -ju, -is, -it, -im, -ite, -ját
4. -u, -is, -it, -im, -ite, -at

SOMOGYI GYÖRGY



```

5 REM OROSZ IGE
6 REM EZZEL A PROGRAMMAL KELL KIEGESZITENI A CIRILLETUKKEL FELTOLTOTT
7 REM "KARAKTER" NEVU PROGRAMOT.
8 CLEAR 500
9 DIM K$(20),R$(4,6),B$(20)
10 FOR I=1 TO 20 : B$(I)=CHR$(I+127) : READ K$(I) : NEXT I
14 DATA 1,G,D,2,1,J,L,4,V,F,5,6,9,7,8,9,0,W,U,0
15 CLS : U$="" : E$="" : PRINT CHR$(6)
20 FOR I=1 TO 6 : READ O$(I),OH(I) : A$=O$(I) : GOSUB 2000 : O$(I)=A$ : NEXT I
21 DATA 0,1,TQ,2,OH DNA,OH0,10,MO,2,ED,2,ONI,3
30 FOR I=1 TO 4
32 FOR J=1 TO 6
36 READ R$(I,J)
50 NEXT J
52 NEXT I
53 DATA U,ES9,ET,EM,ETE,UT
54 DATA Y,ES9,ET,EM,ETE,YT
55 DATA U,IS9,IT,IM,ITE,OT
56 DATA Y,IS9,IT,IM,ITE,AT
105 PRINT "A 'JO' BETU HELYETT AZ 'E' BETUT, A '3'
106 PRINT "BETU HELYETT A '3' SZAMOT KELL IRNI."
110 PRINT "AZOKAT A BETUKET, AMELYEKET A GEP BILLEN--"
111 PRINT "TYUIROL NEM LEHET BEIRNI, A KOVETKEZO"
112 PRINT "TABLAZAT SEGITSEGEVEL, KODOLVA KELL"
113 PRINT "MEGADNI!"
114 PRINT " I R D B E A R A G O T T "
150 FOR I=1 TO 20 STEP 4
151 PRINT B$(I);E#;K$(I);U#;B$(I+1);E#;K$(I+1);U#;B$(I+2);E#;K$(I+2);U#;
    B$(I+3);E#;K$(I+3)
156 NEXT I
158 PRINT : PRINT : PRINT
200 READ A$
202 IF A$="" THEN 2500
203 GOSUB 2000
204 READ Z
206 D$=A$
300 FOR I=1 TO 6
310 A$=R$(Z,I)
315 GOSUB 2000
320 PRINT# 13,0,D$(I) : PRINT#13,OH(I)+1,D$
330 INPUT W$
331 PRINT# 12,0,"
333 PRINT# 12,0,D$(I) : PRINT# 12,OH(I)+1,D$;A$
336 IF W$=R$(Z,I) THEN BEEP 50,100 ELSE BEEP 500,200 : HIBA=HIBA+1
340 PRINT# 13,0,"
350 PRINT# 14,0,"
400 NEXT I
1999 GOTO 200
2000 REM DEKODOLO
2005 H=LEN(A$)
2010 FOR Q=1 TO H
2020 AQ$=MID$(A$,Q,1)
2030 FOR W=1 TO 20
2040 IF AQ$=K$(W) THEN AQ$=B$(W) : W=20
2050 NEXT W
2060 A1$=A1$+AQ$
2100 NEXT Q
2109 A$=""
2110 A$=A1$
2111 A1$=""
2130 RETURN
2500 CLS : PRINT HIBA;"ALKALOMMAL ADTAL HIBAS VALASZT!"
5000 DATA BYLO,1,418,2,GOBOP,3,CLQS,4,611A,1,2D,2,17,2,MA2,2,4AS,2,4LA6,2
5002 DATA FE2,2,CKA2,2,CTAH,2,B039M,2,HA6H,2,40JM,2,4PIM,2,
    CHIM,2,1EP,2,30B,2
5004 DATA K$16,4,CTD,3,4G,1,CMQ:F,3,19,1,L9,1,49,1,21B,2,CTPD,2,MC,1
5006 DATA 4LOB,2,KP1KH,2,TOH,2,TPE1Y,1,PICY,1,DA,1,HEC,2,KLAD,2,FACT,2
5008 DATA ID,2,ED,2
6001 DATA *
    
```

Hattyúdal

— RUN! — dörrentett rá az operációs rendszer Lötifutira, a zöldfülű BASIC programcskára, aki ijedtében hanyatt-homlok futásnak eredt. Már túl volt két DIM-en és három READ-en. Szerencsésen kikerült egy jól álcázott STOP-ot, mikor némi egerutat nyerve, a magaslemezre menekült. Minden zugot és jó néhány fájlnévet végigböngészve még mindig nem bukkant a keresett adatállományra. Am ekkor váratlanul, egy félreeső szektorban belebotlott a lustán ott heverésző FORTRAN-fordító egyik blokkjába.

— Mit keresel erre őcsém, ahol még a madár se jár? Lötifuti hetykén visszakerdeztet.

— Ki vagy? A híredet sem hallottam eddig.

— Ez nem csoda. FORTRAN apó vagyok. A világon se voltál, amikor évekkal ezelőtt utójára értem jött a LOADER. A már akkor is porosodó testem miatt elszállt a rendszer, azóta a kutyá sem törődik velem. Valamikor elképzelhetetlen volt nélkülem egy valamirevaló programfejlesztés. Kódom gyakran plántálák gépi kód sáncára, s nyögte FORTRAN hős hadat gépnek büszke tára.

— FORTRAN-t mondtál? Úgy tudom, az egy kökorszaki programozási nyelv. Eljárt föltötte az idő. Múzeumban lenne a helyed. Manapság a BASIC-vel dolgozik boldog-boldogtan. Már a televízióban is szerepeltem — dicsekedett Lötifuti.

FORTRAN apó gondolatai a távoli múltban bolyongtak.

— Bizony, öreg vagyok, mint az országot — sóhajtott. — A számítógépek hőskorában születtem, az első magas szintű programozási nyelvek egyike voltam. Korán szabványosították. Nyelvezetem egyszerű, de általános célú. Könnyen megtanultak, hatékony fordítóprogramok készültek. A gépi reprezentációk közti hordozhatóságom is hozzájárult egykori népszerűségemhez. Az idő előrehaladtával a szabvány nyelv egyre korszerűtlenebbé vált. Egyes gépcsaládokban és némelyik feldolgozási területen háttértárbá szorultam. A '70-es évek végéig még így is jó helyen álltam.

A szakemberek jelentős hányada rajtam nevelkedett, és ők nem feledtek. Próbáltak vértömlesztéssel megfitalítani. Új — strukturált nyelvi elemeket is tartalmazó — szabvány készült, de a FORTRAN-77 már nem válhatta meg a világot. Nyelvi kincsemet előfordítókkal is bővítették, egyiküket sem ismerte el a szakma általánosan. A strukturált programozás hívei átgyergeltek más nyelvekre. A személyi számítógépek megjelenése adta meg nekem a kegyelemdőfést.

— És új nap született az égen. BASIC — az ASZ — húzta ki magát Lötifuti, s már éppen menni készült, de FORTRAN apó intésre emelte ujját.

— Igen. Végigpörtetek a világon, mint a spanyolnátha. Letettél a trónról, mint Uránoszt fia, Kronosz, de fogsz még te is követ nyelni!

— En nem vagyok a fiad! — feleselt a tájekozatlan Lötifuti. — Úkapámnak is öreg lennél.

— Tévédz! — mosolyodott el deresedő bajusza alatt a vén fordító. — A BASIC sem mai csirke, negyedszázados múlttra tekint vissza. Nyelvi szerkezete a FORTRAN-ra épült, és — szerintem — ingatagabb nála. Lötifuti, aki eddig mit sem tudott családfájáról, most megtorpant.

— Ha ez igaz, akkor én sem vagyok korszerű. Ennek ellentmond, hogy elterjedtem a legújabb korban.

FORTRAN apó folytatta a tanítást:

— A programozási nyelvek gépre vitelének két módszere terjedt el (1. ábra). Az értelmezési eljárás a programozó szempontjából nézve sokkal egyszerűbbnek látszik, mint a fordítási programfejlesztés. A fordításnál gyakori eljárás, hogy a különböző forrásmódlokra szabdaltnak program különböző tárgyprogramokhoz vezet. Hogy program lehessen belőlük, meg kell ismerkedniük egymással az összekapcsolt program közreműködésével. A programozó feladata: a PROGRAM létrehozása. A munka két jól elkülönülő részre bomlik. A FORRÁS papírja való elkészítését nevezzük programtervezésnek. A második lépés a programfejlesztés, ekkor alakul ki a

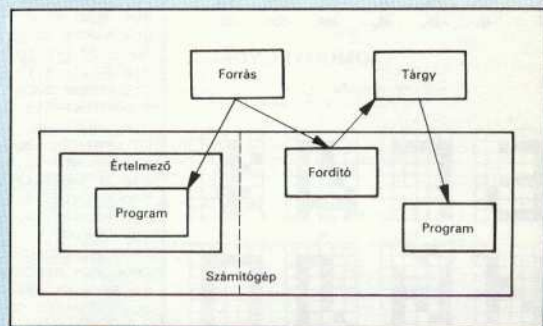
PROGRAM. Programot tervezni annál könnyebb, minél jobb lehetőségeket kínál a nyelv a feladat modellezésére. A programfejlesztés akkor könnyelmű, ha ehhez egyszerű módszert nyújt a megvalósítási környezet. A BASIC megalkotói a második szempont szerinti könnyítést helyezték előtérbe. A felhasználót elválasztották az operációs rendszertől, és közvetlenül a BASIC rendszerrel támogatják minden tevékenységét. Tehát a BASIC nem csupán a programozási nyelvet, de az operációs rendszer jó néhány funkcióját is magába foglalja.

Az értelmező környezet könnyelműsége teszi a programfejlesztést. A FORRÁS és a PROGRAM formátuma egymástól alig tér el, ugyanis alapelve, hogy a PROGRAM belő-

hanem a BASIC rendszer számára! A nyelv olyan, a programozó oldaláról nézve kényelmetlen korlátokat állít, amelyek az értelmező dolgát könnyítik, adott esetben gyorsítják. Csak példaként említem a túlságosan rövid, változó neveket.

— Hogyha nem a programozónak készült a BASIC nyelv, akkor hogy lehet gyorsan megtanulni? — jutott szóhoz Lötifuti és levegőhöz FORTRAN apó. — Pedig ezt a tulajdonságát úton-útfélen hirdetik.

— Megtanulni könnyű — bólintott FORTRAN apó. — Kurta farkú feladatok megoldásakor nem is érezzük hiányosságait. A vakolókanalet sem nehéz kézbe venni, de próbáljunk meg vele felhőkarcolót építeni!



1. ábra

vése során azonnal listázható, és szükség esetén egyszerűen lehet javítani. Ezek után — természeténél fogva — a PROGRAM üzemszerű felhasználása nem lehet hatékony. Te is, kedves Lötifuti, akármilyen büszke is vagy ifjúságodra, mégiscsak ölomlábakon mozogsz. Ezt tudták a BASIC tervezői is, és ott csökkentették az értelmezőprogram dolgát, ahol lehetett. A nyelvet leegyszerűsítették, de nem a programozó,

Lötifuti végleg elbizonytalanodott.

— Te nagyon okos lehetsz! — Csak a korral együtt járó tapasztalat beszél belőlem — szerénykedett FORTRAN apó.

— Kérlek, mesélj még magadról! Most már nagyon érdekel a múltad, hiszen mint mondtad, én is leszármazottad vagyok.

— A programok információfeldolgozást végeznek. Ezért a programozási nyelvek-

GOTO utasításokat? Az IF...THEN forma már előre jelez valamit a program továbbfolyásáról. Honnan származik a probléma? Onnan, hogy néhány általános célú utasítást kell használnunk az egyébként sokszor előforduló konkrét részletek leírására. Arról már nem is beszélünk, hogy a programozót e lehetőség nem készíti a strukturált gondolkodásra és programkészítésre. Az utóbbi évek tankönyvei bőségesen elemzik ezt a gondot. Még csak annyit, hogy az IF-beli feltételnegálás egészen elrettentő, de a FORTRAN-ban e ponton kikerülhetetlen fogás. Olyan, mintha valaki a jobb kezével a bal fülét vakarná meg. Persze a félkarú ember kénytelen ezt cselekedni, márpedig mi csak az egyik kéz használatát engedélyezzük.

— És a strukturált FORTRAN-előfordítók? És a különböző, megfelelően bővített BASIC rendszerek?

— Nem ellenzem az előfordítók használatát. A baj csupán az, hogy egyikük sem kapott olyan rangot, hogy általánossá válhatott volna. Ami pedig a különböző BASIC rendszereket illeti, már a jelzőjük miatt sem tetszenek. KÜLÖNBÖZŐEK! A FORTRAN szabványtervezői azt mondták: ez meg ez meg ez legyen a nyelvben! A BASIC megalkotói: ez lehetőleg legyen a nyelvben! Ez pedig benne lehet, meg ez is, meg ez is. A BASIC „szabvány” kialakításában nem teljesen érthetetlen szempontok vezérelték a tervezőket. A nyelv megvalósítói aztán éltek és visszaéltek a modularitás elvével. Bizonyos modulokat bevettek, másokat kihagytak. Az egyes modulok nyelvi készletét is igen rugalmasan kezelték mind a forma, mind a tartalom tekintetében.

— Ez nem jó?

— Bábél építői sem érték el az eget — mondta ki a summát FORTRAN apó.

Lótífuti fészkelődni kezdett, megdörzsölte zsiabdát szubrutinját, majd büszkén lóbálta meg a fordító orra előtt.

— Látod? Azt csinálók velem, amit akarok.

— Dicséretre méltó dolog — évődött FORTRAN apó. — Bezzeg az én alprogramjaim büjösckáznak velem. Nem győzöm hívni őket. Jöszerve

csak paraméterezünk egymással.

— Én minden gondolatát ismerem a szubrutinnak, nem lehet előttem titka — mondta az öntelt Lótífuti.

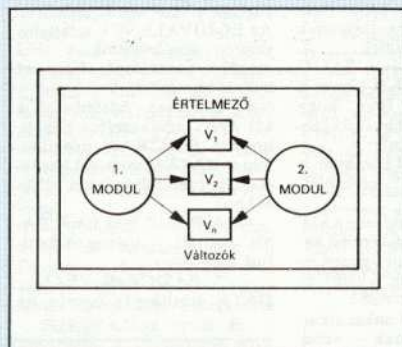
— Kisebbségem is nagyobb annál, sem hogy minden szubrutint kívülről-belülről ismerjek — dohogott FORTRAN apó. — Tegye meg, amire kérem. A többi az ő magánügye. Nem is beszélve arról, hogy a BASIC rendszer túlnyúl demokratikus, az anyagi javak köztulajdonban vannak. Míg hátat fordítasz, egy álnok alprogram elronthatja a változóidat.

40 GOSUB 1000
50 LET Z1 = X1
60 LET Z2 = X2

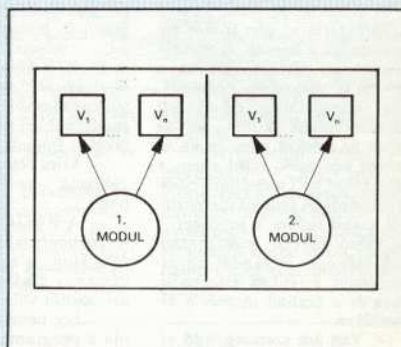
A BASIC program változói valójában a BASIC rendszer felügyelete alatt állnak, a FORTRAN-beli változó egy rutin birtoka (4. ábra). Tehát nálam két modul azonos nevű változói is különböznek. Pontosabban: a hívott modul a hívó változóinak némelyikére rálat (5. és 6. ábra). Továbbá, hogy teljes legyen a kép, a FORTRAN programmodulok is osztozhatnak egyenlő joggal adatokon. Ezek az adatok közösek, tehát egyik programzsemsen sem birtokolja őket (7. ábra).

dolgozásra, ha ezekről a főprogram — indokoltan — nem hajlandó tudomást venni. Tehát a FORTRAN elég jól támogatja a moduláris programozási technikát. Bonyolultabb feladat megoldásakor enélkül szinte lehetetlen boldogulni. A BASIC alprogram csupán a programvezérlés tekintetéből szubrutin. Igaz, szerencsésen megválasztott konvenciók betartásával a BASIC-ben is lehet modulszerűen dolgozni, de ebbe most nem mélyedünk bele!

— Érdekeségeket mesélsz, apó. Én még sohasem gondolkodtam ezeken a dolgokon.



4. ábra



5. ábra

— Igaz. A múltkor nehezen derült fény egy ilyen esetre. Azután másik változót kaptam a programozómtól.

— Én csak azokat a változókat nyomom a szubrutin kezébe, amelyekkel éppen dolgozni szeretném. Ez az eljárás azt is lehetővé teszi, hogy az általános célú szubrutin különböző hívásainál más és más paramétereket adhassunk neki. A te általános célú alprogram hívása előtt értékadó utasításokkal kell feltüntetned a bemenő paramétereket hordó változókat, majd visszatérés után nagy valószínűséggel mented a kimenő paramétereket. Mondjuk, egy másodfokú függvényt megoldó szubrutin hívása így néz ki FORTRAN-ban:

CALL MASODF (1, P, Q, Z1, Z2)

és így festhet a BASIC programon belül:

10 LET A = 1
20 LET B = P
30 LET C = Q

— Nem fölösleges ennyi eszközt készenlétben tartani? — kérdezte Lótífuti. — A programozó nehezen fogja eldönteni, vajon paraméterezzen-e, vagy inkább COMMON-területre helyezze az átadandó információkat.

— Van, amikor az egyik, van, hogy a másik eszköz érdemesebb választani — felelte FORTRAN apó. A tapasztalat alapján mondhatom: a hívó és a hívott programrés közti információcserét célszerűbb paraméterezni, hiszen a paraméterlistához csak ők ketten férnek hozzá. Ha azonban egymás mellé rendelt rutinok akarnak kommunikálni, akkor ebbe az őket vezérlő főprogram ne üsse bele az orrát! Tekintsünk egy vezérlőprogramot, amely fölvaltva hívogatója a bemenő rekordokat beolvassa, valamint a beolvasott adatokat feldolgozó modulokat. A bevitelt végző rutin csak a COMMON-területen keresztül továbbíthatja a rekordokat fel-

Valami nagyon fúrja az oldalamat. Kérlek, mondd meg nekem: miért olyan hanyagok a FORTRAN-programozók?

— Honnan veszed ezt? — hökkent meg az öreg.

— Azt hallottam, hogy sokszor elfelejtene sorszámat írni az utasítások elé.

— A FORTRAN-ban nincs sorszám.

— De igenis! Láttam olyan utasítást, amely előtt volt — erősödött a BASIC program.

Hogy Lótífuti meg ne sértődjék, FORTRAN apó csak csendesen kuncogva válaszolt:

— Az nem sorszám lehetett, hanem címke.

— Nem mindegy, minek nevezüdk?

— Egyáltalán nem! — csattant föl FORTRAN apó. — A téves nézetek döggeselelyű módjára keringenek a pontatlan szóhasználat felett. Majdnem minden programozási nyelvben előfordul, hogy egy-egy utasítás hivatkozhat a másikkra. A FORTRAN-ban ilye-

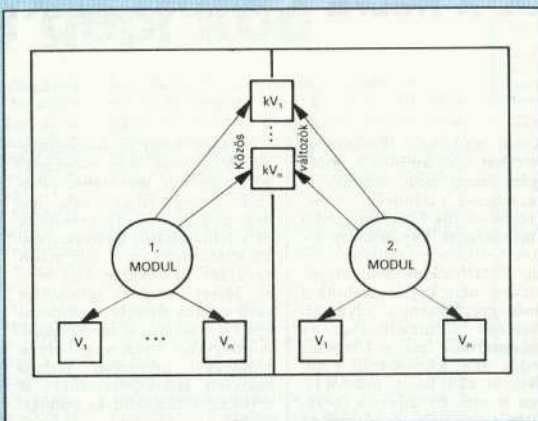
nek a GOTO, a DO és a formátumos bemeneti-kimeneti műveletek. A hivatkozás eszköze a címke. A FORTRAN-címke — mindössze a hivatkozott utasítást látjuk el, azonosítjuk vele — írásának szabályra korlátozott. Csak számjegyeket tartalmazhat, ezért a sorszám benyomását keltheti bennünk. Más programozási nyelvekben címkézéshöz felhasználhatunk betűket is, tehát szemléletesebb címkét konstruálhatunk.

— A BASIC GOTO és GOSUB utasításai is címkékre hivatkoznak? Akkor miért nevezdük sorszámmal?

— A BASIC-beli sorszám címként is használható, de tényleges sorszámként is szolgál. A rendszer növekvő sorszám alapján hajtja végre az utasításokat.

— Igaz is! — tapogatóztak félhomályban Lótífuti gondolatai. — A FORTRAN programban mi jelzi az utasítások sorrendjét?

— Mint a legtöbb programozási nyelvben, a felírásunk sorrendje. Nem annyira a FORTRAN, mint inkább a BASIC különbözik azzal, hogy minden utasítás sorszámával a programozó veszködik. Érdemes a forrást sorszámok nélkül papírra vetni, átmenetileg — akár szimbolikusan is — címkézni néhány utasítást, és csak a rögzítéskor látni el minden sorát tényleges sorszámmal. A BASIC rendszer nemcsak a program futásánál használja a sorszámot, hanem rögzítések is.



7. ábra

— Tényleg! A FORTRAN-forrásból hogyan lehet sort törölni? Vagy újat beszúrni? — lendült utolsó ellenrohamra Lótífuti.

FORTRAN apó vállalt vont. — Ez nem tartozik rám. A forrásszöveg karbantartását általános célú szövegszerkesztők támogatják, őket kell megkérdezned. Én a rögzített forrást ajándékba kapom.

— Hű, akkor a FORTRAN-programozóknak sok mindent kell ismernie a programfejlesztéshez! Szövegszerkesztőt, fordítót, esetleg előfordítót és összekapcsoló programot. És ki tudja, még mit?

— Bizony, a jó szakembernek szakmája szerszámai és technológiái is illik ismernie.

Csak így végezhet hatékony munkát. De még nem merítettük ki a sorszámtemát. Nagyon fontos felhasználási területe a programvezérlés nyomon követése. Ha valamilyen gubanc miatt elszáll a BASIC program, a rendszer könnyen megjelölheti a hibás sort, nem kell sokáig keresgélnie a programozónak. Én e problémát úgy oldom meg, hogy külön kéresem hajlandó vagyok automatikusan sorszámozott forráslistát készíteni a fordítás melléktermékeként. Egy másik kérelemre olyan tárgykódot generálok, amely ezekre a sorszámokra futási hiba felleptekor utalni fog. A hibátlan programból végül ki lehet hagyni ezt a többletkódot. E szolgáltatásom már nem közvetlenül a nyelvvel kapcsolatos, hanem inkább a programfejlesztő eljárás része. Ezért nem biztos, hogy minden fordító rendelkezik vele. Különböző futási hibák figyelésére — mint a tömbök indexhátairainak átlépése — készítik vagy elhagyó kódot, a programozó igénye szerint.

Elhallgattak. FORTRAN apó azon tanakodott, mit felejtett ki előadásából. Lótífuti fejé zsongott a tömördek, számára új összefüggéstől. Kusza gondolatait fésülgette, majd szedelőzködni kezdett.

— Lassan mennem kell. Felhasználom nem szereti, ha sokáig elmaradok. Barátom a minap végtelen ciklusba keveredett, s lelőtték, mint a veszett kutyat. Nem szeretnék a sorsá-

ra jutni — mentegőzött. — Ennyi idő alatt még egy 1/0 művelet is befejeződik.

— 1/0! — csapott homlokára a fordítóprogram. — Ezt még el kell mondanom! Már említettem, csebul állunk az adatszerkezetekkel. Ezzel párhuzamosan az adatátvitel területén sem jeleskedünk. A szekvenciális input/output-on túlrá az én szabvány nyelvem nem jutott. A tiéd valahányadik modulja megemlíti a közvetlen fájlkezelést, de az implementációk zöme efeléldi megvalósítását. A FORTRAN reprezentációk között viszont elterjedtek — a szabványt kisebb-nagyobb eltérésekkel meghaladva — a direkt állományok kezelését segítő nyelvi elemek.

— Úgy tudom, hogy a FORTRAN-programozók sokat bajlódik az átvitt adatok konverziójának megadásával. Ismeretlen dolog az automatikus konverzió, mint amelyet az én INPUT és PRINT utasításaim végrehajtanak.

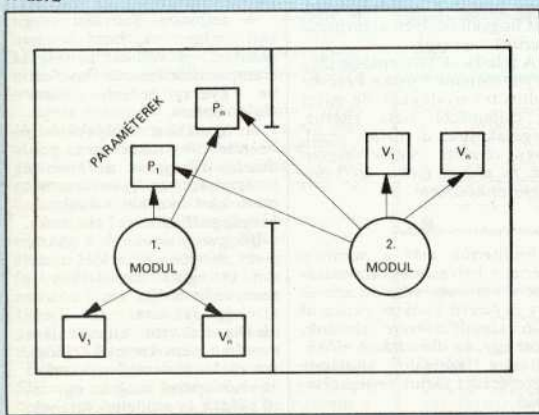
— Bár jó néhány fordító be-tölti ezt az irt megfelelő elemekkel, a szabvány FORTRAN valóban nem gondoskodik róla. A READ és WRITE utasításokhoz formátumleírást társítunk. Mondok még egy különbséget. A FORTRAN tudja, hogy nem minden periféria szolgálja az ember és a számítógép párbeszédét. Hát-tértáron történő adattárolás esetén általában nem szükséges az adatkonverzió. A FORTRAN-programozó a számokat belső ábrázolási formájukban is átviheti, valószínűleg helyet és egész biztosan időt megtakarítva. A BASIC-programozó a PRINT és az INPUT utasításokkal csak a szöveges adatokat kezelheti konverzió mellözésével.

FORTRAN apó és Lótífuti nyiladozó barátsága drámai véget ért. Az öreg fordító utolsó szavait már elhaló hangon, lehunyt szempillákkal rebegte. Hosszú életét megtörte a végelgyengülés.

— „Ne ily halált adj, Istenem!” — suttogta részvétell Lótífuti, és óhaja betelt. A túrelmetlen programozó elküldte számára a selyemzsinórt, de ennek Lótífuti sem örvendezett jobban, mint az egykor vesztés ozmán hadvezérek.

SZABÓ ANDRÁS

6. ábra



A MINŐSÉGÜGY KÖZÜGY

HOL A HELYE A MINŐSÉGNEK?

Sokban tudatosul mostanában, hogy az egészséges fejlődést valami zavarja. Ezek a felismerések azonban, sajnos, nem jutnak túl a jó szándékú ostorozáson. És szinte mindenki abban látja a bajok gyökerét, hogy „alacsony a minőség”. Ez azonban csak megtevesztő látszat, az igazi okok, noha kapcsolatosak a minőséggel, mélyebben vannak. *A minőséggel ugyanis sohasem az a fő baj, hogy alacsony, hanem, hogy nem tudják kielégítően irányítani.* Ennek pedig szinte minden esetben az a fő oka, hogy az embereknek helytelen elképzelések vannak arról, hogy hol is a helye a minőségnek tulajdonképpen.

Ha pedig nem tudjuk valaminek a helyét, akkor azt sem tudhatjuk, hogy miktől függ. Így helyes irányítása is lehetetlen.

A hibák felderítéséhez és a javítási módok kutatásához először ne azt nézzük, hogy hol kellene lennie a minőség helyének, hanem azt, hogy hová helyezik, hol juttatják szerephez, illetve hol foglalkoznak vele az emberek.

Azt fogjuk tapasztalni, hogy a minőség nem állandóan és tudatosan jelenlévő, hanem csak egyszer-egyszer, külső kényszer hatására felbukkanó jelenség a fejekben.

Azán azt is észre vesszük, hogy a felbukkanások nemcsak időben, hanem térben is korlátozottak: szinte kizárólag csak az ún. *átvételi helyeken* kapnak szerepet. Ott, ahol például anyagátvételt vagy gyártás belüli átvételek zajlanak, vagy a kereskedelemben, ahol a vevő átveszi az árut.

Természetes a kérdés, hogy helyese-e ez. Ha pedig nem, mi volna a helyes?

Gondolkodjunk folyamatokban! Így rögtön észre vesszük, hogy a minőség (pontosabban a különböző minőségjellemzők) folyamatjellemző, és a folyamat más jellemzőivel van szoros kapcsolatban. (Ez abban az — egyébként vitatható — esetben is igaz, amikor a minőséget végtermékjellemzőnek fogja fel valaki. Ilyenkor is igaz az, hogy a minőség más folyamatjellemzők függvénye.)

A folyamatok egy része kívülről is hatáskörünkön. Hatáskörünkön belüli folyamatokat

viszont irányítani igyekszünk. Így tehát van, aminek a minőségére hatni nem tudunk, és van, aminek a minősége többé-kevésbé tőlünk függ, mégpedig a minőséggel kapcsolatos folyamatjellemzők irányítása révén. E felfogásomban azonban már nem korlátozódunk a minőségre, hanem a folyamatjellemzők szemeltartása és kézbe tartása lesz a központi kérdés, akár közvetlenül a minőségről, akár bármilyen másról legyen is szó. És mivel a folyamatjellemzők a rendszerben mindenhol, minden pillanatban szakadatlanul kapcsolatban vannak, rögtön kiderül, hogy a minőség minden helyről, minden pillanatban hatásos lehet kitevé.

A minőség alakítása tehát lényegében mindenütt és minden pillanatban folyik kisebb vagy nagyobb mértékben. És mivel a minőség kézbe tartása csak a minőségre ható összes jellemző állandó kézbe tartása révén valósítható meg, a minőségről sehol egy pillanatra sem feledkezhetünk meg.

Ha alaposan végiggondoljuk, rájövünk, hogy nem is kell mással foglalkoznunk, nem is kell mátra figyelniünk mint a minőségnek. Csak annak nehez a megállapítása, hogy mikor minek a minőségét kell fontosabbnak tartani, és hogy a minőségek alakításának mikor melyek a legelőnyösebb — a legjobb minőségű — módjai.

Miután belátuk a minőségnek folyamatokhoz kötöttségét, kimondhatjuk azt is, hogy amikor minőségről és minőségirányításról (minőség-kézbe tartásról) van szó, akkor *mindig* folyamatok alakulása és folyamatirányítása kell gondolnunk.

A folyamatirányítás azonban mindig közelítése valamiknek valamikhez. Tevékenység, ami valamit minél pontosabban utánozni, követni, approximálni akar.

Ezek az approximációk azonban általánosabbak a megszokott matematikaiknál, ugyanis itt nem (csak) az a fontos, hogy mivel — milyen formulával — approximálunk, hanem az approximálónak az egész előállítási folyamat és annak összes jellemzői is lényeges szerepük és számításba veendőik.

Világos, hogy az approximáció jószágának (minőségének) jellemzéséhez, méréséhez nemcsak a végeredménynek, hanem az előállítási folyamatnak és a felhasználói igények megközelítésének a jellemzőit egyaránt figyelembe kell venni. Mivel pedig az igényektől való eltérés sokféle szempontból vizsgálható, e szempontok mindegyikét ilyen vagy olyan erősséggel szerephez juttató összetett eltérésjellemzőket is célszerű kialakítani és felhasználni.

Rátérve az elmondottak gyakorlati hasznosítására, az első és legfontosabb lépés a termék vagy szolgáltatás minőségjellemzőinek összegyűjtése.

A második lépés a terméket előállító folyamat, illetve szolgáltatást eredményező folyamat összes olyan jellemzőjének számbavétele, amely lényegesen befolyásolja a minőséget.

A harmadik feladat az összegyűjtött jellemzők közötti kapcsolatok feltárása és matematikai rendszermodellbe foglalása.

A negyedik feladat azoknak a jellemzőknek a kiválasztása, amelyeknek értékét alkalmas módon alakítva, a kiválasztott minőségjellemzők értékeinek alakulását tudjuk a kívánások szerint biztosítani.

Ez utóbbival együtt kell elvégeznünk azt a feladatot is, hogy magát a minőségirányítást hogyan (milyen algoritmus szerint) végezzük.

A feladatok felsorolását még folytathatnánk többé-kevésbé pontos tennivalókkal, de mivel ez céljainktól kissé eltérne, megszakítjuk a listát azzal, hogy célszerű *tevékenységtervet*, és ahhoz előbb *tevékenységlistát* készíteni.

Említettük már a minőség mértéke helyes megválasztásának a kérdését. Mielőtt ennek egy egyszerű konkrét példával való szemléltetésére térnénk, most egy, az illusztráció előkészítésére szolgáló általános észrevétellel zárjuk fejtegetéseinket.

Képzelnék el, hogy valaki szép lassan, folytonosan csök-

kenti az odafigyelést, a gondosságot, amit munkájára fordít. E folyamatot biztosra vehetően a minőség monoton romlása követi. Igen, de meddig? A végtérmeék értékét ezen a módon általában nem lehet a végletekig csökkenteni, még ha szándékosan rosszra akarnánk is csinálni.

Vizsgálódjunk most a másik véglet közelében! A gondosság és az odafigyelés növelésével a minőség biztosan javul egy darabig. A tökéletes gyártmány vagy szolgáltatás így ún. eljutni azonban hiú ábránd.

A „dolgozz hibátlanul!” elv ösztönösen sejtí, hogy nem elég a minőséggel csak bizonyos helyeken és bizonyos időkben foglalkozni. Célja azonban a gyakorlatban megvalósíthatatlan. A hibátlanosságnak, a legjobb minőségnek az egyenletes tartása ugyanis lehetetlen. Hibátlanul dolgozó ember éppúgy nincs, mint hibátlanul dolgozó gép.

Alkalmazzuk azonban egy pillanatra az operációkutatást! Ez felveti a kérdést, hogy mit nyerünk a minőség javításával, és mibe kerül ez a javítás nekünk. Csak akkor érdemes a minőséget javítani, ha az rövidebb vagy hosszabb időszakot figyelembe véve, előnyös a gyártáson. A hibátlanul dolgozó alkalmazott (ha ez egyáltalán sikerülne is neki) oly mértékben lassú kibocsátást és esetleg további költségeket eredményezne, hogy ez bizonyos veszteségessegghez vezetne.

A teljesség kedvéért meg kell említenünk, hogy — ha pusztán egy vállalat pénzügyi szempontjait vesszük figyelembe — gyakran érdemes a minőséget rontani.

A termékbe, szolgáltatásba beépítendő szakértelem, gondosság kihagyása azonban a „megtakarításnak” sokszorosát elérő kárt okozhat másoknak. Megengedhető-e ez? Ha az ún. célfüggetven nemcsak a gyártó javát, hanem a vásárlóét is, nézünk az egész társadalmát is szem előtt tartja, nincs okunk az aggodalomra. Az ilyen, ideális eszközök kimunkálása azonban nem könnyű feladat. Az ezekhez vezető gyakorlati út első lépései azok az egyszerű példák és modellek lesznek, amelyekkel majd legközelebb foglalkozunk.

BASIC és gépi kód

Sorozatunk a Commodore számítógépek gépi kódú programozásába vezeti be az érdeklődőket. Az alapvető ismereteken kívül bemutatja a különféle alkalmazásokat, programozási fogásokat, programo-
soraimat, a különféle alkalmazásokat, programozási fogásokat. Nem tanfolyam-szerűen, inkább kirándulások formájában, újra meg újra visszatérve egyes területekre, más-más nézőpontból szemlélve a témákat.

A három legerjedjebb géptípus, a VC20, a CBM64 és a C16 belső felépítése lényegesen eltér egymástól. A beépített mikroprocesszorok utasításkészlete, működése azonos, de a memória felosztása mindegyik gépénél más, a különleges szolgáltatásokról nem is szólva. (Ilyen például a sprite-kezelés a CBM64-esen vagy a beépített monitor a C16-nál.) Ez a későbbiekben nem kevés problémát okoz, de remélhetően sikerül ezeket valamilyen módon át-
hidálni.

A leírtak megértéséhez és kipróbálásához különböző előismeretek nem szükségesek. A BASIC nyelv ismeretén kívül csak a hexadecimális (tizenhatos alapú) számrendszer használatában való jártasságot feltételezem. A hexadecimális számokat — a konvencióknak megfelelően — a szám előtti \$ jel jelöli.

Célszerű még kéznél tartani a Mikroszámítógép Magazin 1986. áprilisi számában megjelent, A 65xx mikroprocesszor gépi és assembler szintű utasításkészlete című összeállítását, amely összefoglalva, áttekinthető táblázatban tartalmazza a szükséges tudnivalókat.

Megjegyzem, hogy a korábbiakban szándékosan nem írtam le a mikroprocesszor típusjelt. Ugyanakkor legalább négy típpussal találkoztam eddig az említett gépeken. Elég annyi, hogy a 6502-esről és utódairól van szó.

Mikor célszerű gépi kódú programozni?

A cím megtévesztő. Nem gépi kódúban célszerű programozni, hanem gépkódú (assembly) nyelven. Mi is assembly nyelven írjuk majd programjainkat, de egyelőre mi magunk fordítjuk le gépi kódúvá. Tehát a kérdést pontosabban így fogalmazhatjuk: mikor célszerű gépkódú nyelven programozni?

Egyik szempont a nagyobb futási sebesség. A BASIC-ben írt programokhoz képest akár kétszázszoros gyorsulást is elérhetünk gépi kódúban. Nem mindegy, hogy két másodpercet vagy hat percelt kell várni egy összetett művelet sor elvégzésére.

Nem lényegtelen szempont a tömörség sem. Az 1. listán látható program által előállított gépi kódú rutin összesen 12 bajtot foglal el a memóriából. Aki teheti, hasonlítsa össze a Mikroszámítógép Magazin 1984/2. számának 19. oldalán található 02 programmal, amely hasonló feladatot old meg BASIC nyelven.

Vannak olyan feladatok, melyeket BASIC-ben nagyon nehezen, vagy egyáltalán nem lehet megoldani. Ezeknél is hasznos a gépi kódú programozás ismerete.

Találkozunk majd olyan esetekkel, ami-

kor BASIC programmal újabb BASIC programokat hoznak létre. Ilyenkor ügyelni kell arra, hogy a létrehozott programok ne keveredjenek össze az őket létrehozó program soraival. A gépi kódú programozás ehhez is segítséget nyújthat.

Ne feledkezünk meg a hátrányokról sem. A magas szintű programnyelvnek nagyon megkönnyíti a programozói munkát a gépkódú nyelvekhez képest. Az assembly nyelvet a programozóknak kell a feladatot elemi lépésekre lebontania, a program megírása, belövése sokkal tovább tart, több fáradsággal jár. A gépi kódú programozásnál ehhez még hozzájön az assembler (az assembly fordítóprogram) munkájának manuális elvégzése: a címformációk adminisztrálása, a mnemonikok (emlékeztető kódszavak) gépi kódú megfelelőinek kikeresése a táblázatból és behelyettesítésük a programba, a kódok elhelyezése a memóriában stb. Érdemes tehát minden feladtnál megfontolni, hogy megéri-e a többtel-fáradságot.

A betöltőprogram

Aki csak BASIC-ben szokott programozni, az is használhatja gépi kódú rutinokat. Különféle könyvekben, folyóiratokban publikált programok részleteit találkoztunk úgynevezett betöltőkkel, de nem ritkák az önálló betöltőprogramok sem. Ilyen betöltőprogramot láthatunk az 1., 2. és 3. listán, ugyanannak a programnak a különböző géptípusokra írt változatait. Az 5. listán lévő programok csak része a betöltő, a 10...50-es sorok nem tartoznak hozzá.

1. lista

```
0 REM KISZAMITOTT GOTO (CBM-64)
1 FORI=828T0939
2 READA:POKEI,A:C=C+A
3 NEXTI
4 IF(CO=1671)THENPRINT"ADATHIBA":END
5 PRINT"RENDENB"
828 DATA32,253,174,32,138,173,32,247
836 DATA183,76,163,168
```

2. lista

```
0 REM KISZAMITOTT GOTO (VC-20)
1 FORI=828T0939
2 READA:POKEI,A:C=C+A
3 NEXTI
4 IF(CO=1799)THENPRINT"ADATHIBA":END
5 PRINT"RENDENB"
828 DATA32,253,206,32,138,205,32,247
836 DATA15,76,163,200
```

3. lista

```
0 REM KISZAMITOTT GOTO (C-16)
1 FORI=828T0939
2 READA:POKEI,A:C=C+A
3 NEXTI
4 IF(CO=1238)THENPRINT"ADATHIBA":END
5 PRINT"RENDENB"
828 DATA32,145,148,32,20,147,32,228
836 DATA15,76,80,141
```

```
10 TT=TT
20 PRINT"TT":FORI=1T0999:PRINT"##"
: NEXT
30 TT=TT
40 GETA:IF(A#="" THENA40
50 PRINT"TT"TT
```

4. lista

```
2 FORI=828T0948
4 READA:POKEI,A:C=C+A
6 NEXT
8 IF(CO=3425)THENPRINT"ADATHIBA":STOP
10 TT=TT
20 SYS828
30 TT=TT-TT
40 GETA:IF(A#="" THENA40
50 PRINT"TT"TT
828 DATA169,147,32,210,255,160,231,162
836 DATA163,42,32,210,255,136,208
844 DATA250,202,208,247,96
```

5. lista

A betöltők működése egyszerű. A gépi kódú utasítások értékeinek decimális megfelelői a DATA sorokban vannak elhelyezve. A program elején lévő ciklusban történik az adatok olvasása a tár megjelölt helyre. A C nevű változóba az ún. ellenőrző összeg kerül. Ha valamelyik DATA-hoz tartozó adatot a programba való beíráskor eltévesztettük, a ciklus lefutása után a program hibajelzést ad és leáll, hogy a hibát kijavíthatjuk.

A betöltőknek nagyon sok változata van. Néhánnyal e sorozat keretében találkozhatunk, de gyakran előfordulnak a lap más cikkeiben is.

Időmérés

Az egyes programrészek futási idejének mérésére egy egyszerű segédeszközt fogunk használni. Ez a 4. és 5. listákon lévő programokba van beépítve. A 10-es sorban a TT nevű változóba tesszük a rendszeróra pillanatnyi állását. Ezt követi a vizsgálandó algoritmus, majd a 30-as sorban tároljuk az előző feljegyzés óta eltelt időt. Ezt PRINT utasítással írathatjuk a képernyőre (50-es sor). A különböző algoritmusok futási idejének összehasonlításához ez elég, de ha valaki másodpercben akarja megkapni az időt, TT/60-at írassa ki. A módszer természetesen alkalmas csak BASIC-ben írt programrészek összehasonlítására is.

A mintaprogramokról

A kiszámított GOTO programjára (1., 2., 3. lista) a későbbiekben még többször visszatérünk. Most részint a különböző gépekhez való különböző gépi rutinokra próbáltunk példát adni, részint pedig a betöltőprogramot mutattam be.

A 4. és 5. listán látható programok az időméréshez adnak illusztrációt. Az előbbi BASIC, az utóbbi gépi kódú utasításokkal végzi el ugyanazt: letérli a képernyőt, majd 999 csillagot ír rá. (VC20-on ez nem mind fér el egyszerre.) Az időmérés nem jelez lényeges eltérést. A gépi kódú rutin sebessége még a hétszeresét sem éri el a BASIC-ben írtak, viszont sokkal bonyolultabb. Példa arra is, hogy néha célszerűbb a BASIC-nél megmaradni.

Assemblerek, cross-assemblerek

4. A rendszer működése. I. A fordítás két menete

A (cross-)assembler fordítóprogramok általában kétmenetesek. Ez annyit jelent, hogy a fordítónak kétszer kell sorról sorra végigmennie a forrásprogramon ahhoz, hogy az ún. tárgyprogramot előállítsa. A tárgyprogram olyan adatállomány, amely a program gépi kódú megfelelőjét, és utasításoként jelző információkat (ún. flageket) tartalmaz. A betöltő- vagy a szerkesztőprogram a jelző információk segítségével állítja össze a végleges, futásra kész programot, amelyet esetleg azonnal be is tölt a mikroszámítógép központi tárolójába. A tárgyprogram háttéráron vagy egyéb, a fordítás indításakor megadott kimeneti periférián áll elő.

A két menet alkalmazása nem szükségszerű előírás. A fordítóprogram egyszerű meneten is fel tudná dolgozni a forrásprogramot, ha előírnánk, hogy a címke- és azonosítóneveket definiáló utasításoknak és direktíváknak mindig meg kell előznie a programban azokat az utasításokat, amelyeknek operandusmezőjében ezek szimbolikus hivatkozásokként megjelennek. Ez azonban olyan szigorú programozástechnikai megszorítást jelent, hogy általában eltekintenek tőle, s ezért tervezik a fordítóprogramokat többnyire kétmenetesre.

Tekintsük át röviden, hogy mi a fordítás két menetének feladata!

Az első menet. Címke tábla

Az előzőekből kitűnik, hogy az első fordítási menetnek az az egyik feladata, hogy a programban használt szimbolikus címke- és azonosítónevek definícióját feldolgozza. Ez egy ún. címke tábla felépítésével történik. Az első menet végére valamennyi használt szimbolikus névnek szerepelnie kell a címke táblában, az értékével együtt. Mint a 2. részben már láttuk, az értékekés kétféleképpen lehetséges: a címke értékét a beültetési számláló pillanatnyi tartalma adja meg, az azonosítót pedig egy direktíva (EQU vagy SET) operandusmezőjének értéke.

A makrók kezelése

Az első menet végzi el a makrókkal kapcsolatos műveleteket is. Sok esetben a makrók kezelése a fordítást megelőző (az ún. preassembler) időben működő külön makróprocesszor feladata. Ha azonban néhány, korántsem szigorú megkövetelt feltétellel élünk, akkor az elvégzendő műveletek beépíthetők a (cross-)assembler első menetébe. Így kiköjtjük, hogy makródefiníciókat nem lehet egymásba ágyazni, és makróhivatkozások csak predefiniált makróra lehetségesek (azaz minden makró definícióját meg kell, hogy előzze a makróra történő hivatkozásokot).

Ha a fordítóprogram új makródefinícióval találkozunk, akkor bejegyzést készít a

Nyolchrészes sorozatunk a mikroszámítógépek assemblereiről, cross-assemblereiről szól. Célja, hogy a cross-assemblerek példáján keresztül megismeresse az olvasót az assembler programok működésével. A bemutatáshoz a rendszermodellezési eszközöket használjuk fel, s így készítjük el az assemblerek működésének egy szabványosított algoritmusát, modelljét. E cikkünk a sorozat negyedik része.

1. (Cross-)assemblerek és a rendszermodellezés
2. Az Intel 8080 assembly nyelv
3. A (cross-)assembler program mint rendszer
4. A rendszer működése I. — A fordítás két menete
5. A rendszer működése II. — Táblák és adatterületek
6. Az operátorkészlet I.
7. Az operátorkészlet II. — A rendszer kapcsolatábrája
8. Példák a rendszer működésére

makródefiníciós táblába. Ez a tábla tartalmazza az összes, a makrókönyvtárban tárolt makródefiníciót, valamint a feldolgozás alatt álló programban definiált és nem katalógizált makró(k) eléréséhez szükséges információkat: a makrók nevét és a háttéráron való elhelyezkedésük kezdőcímét.

A fordítóprogram a műveleti kód mező feldolgozásakor előbb a mezőt a makróhivatkozás szempontjából vizsgálja meg, és csak miután kizárta a makróhivatkozást, akkor nézi tovább, hogy a mezőben érvényes gépi utasítás vagy direktíva mnemonikus kódja szerepel-e. Ha a mezőben a makródefiníciós táblában előforduló névű makróra talál hivatkozást, akkor elindítja a makrókifejtést. Minthogy a belső makróhívások megengedettek, célszerű az egymásba skatulyázás szintjének jellemzésére egy változót (neve MIND) bevezetni.

A változót a 3. részben leírt értelmezés szerint rendszerváltozóznak tekintjük. Értéke \emptyset , ha nem makró fejünk ki, és az értéke minden makrókifejtés indulásakor eggyel nő, illetve a kifejtés befejezésekor eggyel csökken. Ahhoz, hogy egy magasabb szintű makrókifejtésből vissza tudjunk térni az előzőbe, meg kell jegyezni az előző szint megszakitási helyének, vagyis soron következő sorának helyét.

Szükség van az egyes szinteken érvényes formális és aktuális paraméterekre is.

A makrócímkek kezeléséről elmondottak értelmében pedig minden makrókifejtéskor önálló, lokális címke táblát kell felépíteni, így az egyes szinteken a hozzájuk tartozó címke tábla kezdőcímét is meg kell jegyezni. Vannak tehát olyan logikailag összefüggő adatok, amelyek egy-egy makrókifejtés szintre jellemzőek, illetve az adott szinten a fordítóprogram működéséhez szükségesek. Mivel a hívás szintjének növekedésekor ezeket az adatokat elő kell állítani, a szint csökkenésekor pedig az előző szinthez tartozó adatokat vissza kell állítani, célszerű ezeket egy verememóriában (stackben) tárolni. A makrókifejtési veremtár egy szinthez tartozó elemi tehát:

- az előző szint megszakitási helye,
- az adott szinten kifejett makró formális paramétere(i),
- az adott szinten kifejett makró aktuális paramétere(i),
- az adott szinthez tartozó lokális címke tábla kezdőcíme.

A verem minden új makróhívás felismerésekor mélyül, a hozzá tartozó lezáró MEND direktíva feldolgozásakor pedig egy szinttel visszaállítódik.

Utasítás- és direktívafeldolgozás az első menetben

Függetlenül a hívás szintjétől — a MIND rendszerváltozó értékétől —, már az első menetben is foglalkozni kell a műveleti kód mezőben talált gépi utasítás- és direktívakódokkal is. Ha a mező nem makrónevet tartalmaz, akkor az ún. utasítástáblában ellenőrizni kell, hogy a mezőben megengedett mnemonikus kód szerepel-e. Ha a mező egy gépi utasítás kódját tartalmazza, akkor az első menetben elegendő az utasítástáblából csak az adott utasítás tárgyködjének a hosszát kiolvasni. Ezzel a hosszértékkel kell ugyanis a beültetési számláló értékét növelni a sorfeldolgozási ciklus végén.

Direktíva esetén más a helyzet. A direktívákat — minthogy azok a fordítóprogramnak szóló utasítások és a fordítási folyamat alakulását, azaz a rendszer állapotváltozásait is irányítják — már az első menetben is fel kell dolgozni. Kivételt képeznek a DB és DW direktívák, ahol elegendő csak a tárgfoglalás hosszát megállapítani. Az IF direktíva két rendszerváltozó, az IF1 és IF2 változó, az értékek lehet hatással. Az IF1 változóban számláljuk a megnyitott egymásba skatulyázott IF—ENDIF párokat akkor, ha a köztük lévő programrészlet lefordításra kerül, az IF2-ben pedig egy fordításra nem kerülő programrészleten belül számláljuk az egymásba ágyazási szintet.

Valamennyi eredeti forrásnyelvi és a makrókifejtés során a programba bemásolt utasítással a második menet számára átme-

etileg háttértra kerül. Az első menet az END direktíva feldolgozásával fejeződik be. Az első menet feldolgozási folyamatának eredményeképpen tehát előáll:

- a teljes, valamennyi makróhívás kifejtését tartalmazó és a kifejtésekben a formális-aktuális paraméterscseré után létrejövő (másodlagos) forrásprogram, a háttértraon elhelyezve (megjegyezzük, hogy a makrókifejtések tartalmazzzák a MEND direktíva sorát, de a MACRO direktíva sora, a makródefinició első sora, nem másolódik be a hívás helyére);

- a 0-ás szinthez tartozó ún. modulcímke-tábla, amely a program minden sorából elérhető szimbolikus nevek definícióit tartalmazza;

- valamennyi makróhívási szint önálló lokális címkeablája, amely csak az adott,

kifejtett makróörzsnél belül elérhető szimbolikus nevek definícióit tartalmazza.

A címkeablák is a háttértra kerülnek. Ezek az adatállományok képezik a fordítás második menetének bemeneti adatait.

A második menet

A második menet szintén sorról sorra haladva végzi a forrásprogram feldolgozását. A makrók kezelésével kapcsolatos műveletek itt már jelentősen egyszerűsödnek az első menethez képest. A makródefiniciókkal például a fordítónak már nem kell törőnie, ugyanúgy kezelheti őket, mintha IF direktíva után következő és a fordításból kihagyandó programrészeket lennének, csak az ENDM direktíva beérkezését kell figyelnie.

A makróhívásokat tartalmazó sorok felismerések pedig az egyetlen feladat, hogy a fordítóprogram számára az első menetben már bemásolt és aktualizált makróörzshöz tartozó lokális címke-tábla hozzáférhetővé váljék. Mivel a makrókifejtési verem-tár egy szinthez tartozó elemei közül ezek szerint a második menetben csak a lokális címke-táblák kezdőcímeire van szükség, célszerű ezt a többi elemről fizikailag elkülönítve, önálló verem-tárból kezelni. A makróhívások helyén a záró MEND direktíva sorát is éppen azért kellett az első menetben bemásolni a másodlagos forrásprogramba, hogy a fordítóprogram el tudja végezni az ezzel a verem-tárral kapcsolatos műveleteket.

Direktívák

A direktívák feldolgozása lényegében azonos a két menetben. Eltérés a makrókezelésről elmondottak értelmében a MACRO és ENDM, valamint DB és DW direktívák esetében van. Az utóbbiaknál a különbség annyi, hogy a második menetben megtörténik a direktívák operandusainak kiértékelése és a hozzájuk tartozó tárgykód meghatározása. Tárgykód készül egyébként az ORG, DS, END direktívákról is.

Gépi utasítások

A gépi utasításokat tartalmazó sorok esetében más a helyzet. Az utasítástábla alapján a fordító megállapítja az utasítás tárgykódjának hosszát, az utasítás bináris műveleti kódját, valamint hogy szükséges-e az utasításhoz operandus(ok) megadása, és ha igen, akkor annak milyen típusúnak kell lennie. Az utóbbi információk alapján történik az operandusmező kiértékelése, ami után a fordító már összeállíthatja az adott sorban kijelölt gépi művelet tárgykódját.

Fordítási lista

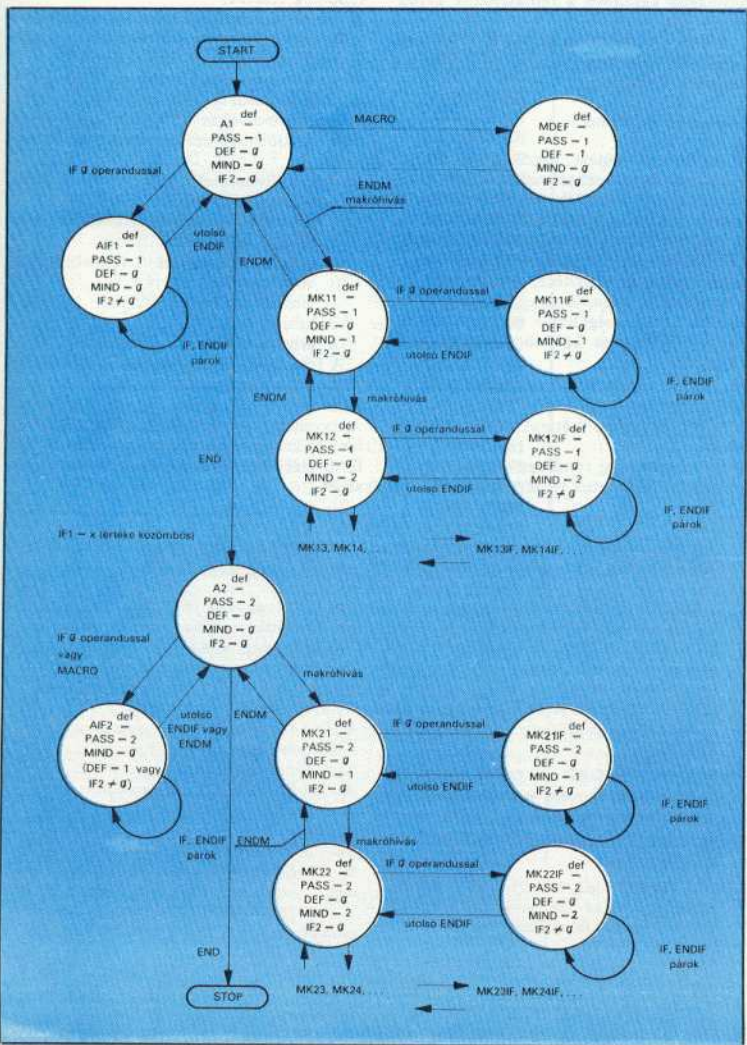
A forrásprogram sorainak feldolgozási ciklusa szükség esetén listanyomatással fejeződik be. A lista nemcsak az eredeti forrásnyelvi sorokat tartalmazza — beleértve a makrókifejtések sorait is —, hanem soronként egyéb hasznos adatokat is szolgáltat.

Ilyenek lehetnek:

- a forrássorok sorszáma,
- a beültetési számláló értéke (az adott tárgykód első bajtjának fordítási tárcíme),
- a tárgykód értéke,
- esetleg külön az operandus(ok) és azoknak a szimbolikus neveknek az értéke, amelyekre a sor operandusmezőjében hivatkozás történik,
- hibábaüzenetek.

Hibafelderítés

Ez az utolsó információ igen fontos a programozó számára. Eddig azzal a feltételezéssel vizsgáltuk a rendszer működését, hogy minden sor formailag és tartalmilag eleget tesz a nyelv szabályainak, és így egyértelműen kiértékelhető. A gyakorlat-



ban azonban ez a feltétel nem mindig teljesül. Ezért a fordítóprogramban számos helyen — operátorban — meghatározott ellenőrzések beiktatására van szükség. A programozó nem létező mnemonikus kódot használhat, szintaktikailag helytelen kifejezést adhat meg operandusként, szimbolikus nevet kétszer definiálhat vagy elmulaszthat definiálni, megsérthet egy sor, az alábbiakban részletezett korlátozó feltételt. Ezeket a hibákat a rendszernek fel kell tudnia fedni, s megtalálásukról értesítenie kell a programozót.

A fordítás befejeződése

A sorfeldolgozási ciklusok az END direktíva beérkezéséig ismétlődnek. A fordítás a tárgyprogram lezárásával fejeződik be. A tárgyprogram — a fordítás eredménye — a kívánt kimeneti periférián (háttértár, lyukszalaglyukasztó stb.) keletkezik, s bemeneti adatként szolgál a következő, a fordítástól már független lépésnek, a betöltő- vagy a szerkesztőprogram működésének.

Az előzőekben nagy vonalakban ismertett működésű rendszernek felrajzolhatjuk az állapotgráfját (lásd az ábrát). Minden állapotot négy rendszerváltó értékével jellemezzük: a PASS nevű a fordítás első vagy második menetét azonosítja, a DEF nevű a makródefiniáció feldolgozására utal, a MIND, (IFI) és IF2 rendszerváltók szerepe már ismert. Az állapotok elnevezése az alábbiak szerint történik:

- Ai — az i-edik (i=1,2) menet alapállapota, amelyben semmilyen makróval kapcsolatos művelet nincs, és a feldolgozásból nem maradnak ki a sorok;
- AIFi — az i-edik menet azon állapota, amelyben a nulla operandusú IF direktíva után következő programrészlet a fordításból kimarad;
- MDEF — az első menet makródefiniációs állapota;
- MKij — az i-edik menet j-edik szintű makrókifejtésével kapcsolatos rendszerállapot;
- MKijIF — az i-edik menet j-edik szintű makrókifejtésében nulla operandusú IF direktíva után következő állapot, amelyben a záró ENDIF direktíváig terjedő sorok a fordításból kimaradnak.

Az egyes állapotok közötti átmenetekben bejelöltük, hogy milyen esemény — a műveleti kód mező milyen tartalmának — bekövetkezése váltja ki őket.

VÁRGEDŐ TAMÁS

Kedves Olvasóink!

Nem rendelésre készült kéziratokat nem örvényel meg és nem küldünk vissza. Levelek, cikkek stb. közlése esetén szerkesztőségünk fenntartja a jogot az írások rövidítésére.

Piac

Az SG2-MKB - önmagáról

Cégünk a Magyar Külkereskedelmi Bank (MKB) és a párizsi SOCIÉTÉ GÉNÉRALE DE SERVICE ET DE GESTION (SG2) közös vállalata, az SG2-MKB Pénzügyi Informatikai Kft., amely 1983 októberében alakult. A közös vállalat tevékenységi körei: a pénzügyi információk szervezése, az informatika és az üzemszervezés körébe tartozó tanácsadás, pénzügyi alkalmazási programcsomagok készítése, a felhasználók képzése.

A közös vállalatban az MKB 51%-os többségi részesedéssel van jelen, az SG2 pedig a maradék 49%-kal.

Tevékenységi köreink:

- ügyfélkiszolgálást végző kisszámítógépes rendszer takarékszoftverek és takarékpénztárak részére (betétkezelés, hitel- és tagnyilvántartás, pénztár, főkönyv);
- nagykereskedelmi vállalat raktárbázisának készlet- és göngyöleg-nyilvántartása, a kiszállítások előkészítése, elszámolási folyamatainak számítógépes feldolgozása, valamint adatszolgáltatás a forgalmi adó elszámolásához és különböző statisztikák készítése;
- komplex könyvviteli szolgáltatás kisszámítógépes bázison kisszövetkezetek részére (tétel könyvelés, mérlegkészítés, elemzés stb.);
- bérelszámolási rendszer kis- és nagykereskedelmi vállalatok részére, amely a havi-

béres elszámoláson túlmenően a jutalékos bérelszámolási formát is tartalmazza;

- pénzügyi és számviteli információs rendszerek szervezése bankok, ipari és kereskedelmi vállalatok, illetve intézmények részére;
- magyar szakemberek foglalkoztatása külföldön;
- tanácsadás (szakvélemények, rendszerek, tanulmányok készítése) külföldi megrendelők részére;
- az aktív memóriakártya kibocsátásához szükséges hardver és szoftver magyarországi meghonosításában való részvétel;
- az általános célú hálótervezési programcsomag külföldi forgalmazása olyan naprakész információkat szolgáltat, amely manuális hálózattervezéssel nem valósítható meg;
- interaktív termelés-előkészítési és tervezési rendszer álmányai:
 - tételtranz (anyagok, félkész-, késztermékek adatai),
 - gyártmányössztétel (darabjegyzék, anyagnorma),
 - termelési program,
 - készlet (raktár),
 - anyagrendelés.

A fent említett területeken Kedves Megrendelőink rendelkezésére állunk és ezen kívül vállalkozunk egyéb egyedi rendszerek szervezését, kivitelezését és adaptálását, valamint kész programcsomagok külföldi értékesítésével kapcsolatos tevékenységet.

ADOK - VESZEK - CSERÉLEK

Ebben a rovatunkban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hírdetéseket közlünk: A díjszabás: közlekednek gépektől soronként (60 karakter) 100 Ft, magán-személyeknek az első sor 50 Ft, minden további sor 20 Ft. Az NJSZT tagjainak az első három sor ingyenes. Hirdetéseiket a szerkesztőség címére várjuk.

● **COMMODORE 16-OS** programokat cserélek kazettán vagy lemezen. Szabó Tamás, Pécel, Bartoshegy u. 13/a 2119.

● **ZX81 16 K-S**, játékkazettákkal eladó. Telefon: 32-54-66.

● **C116-OS** eladó, MK 29-esből kialakított adatmagnóval és sok programmal együtt, 15 000 forintért. Béres Antal, Bp. Labanc út 19. 1021.

● **ELADÓ: ZX-SPECTRUM 48 K-S** 14 000 forint, interface 1. 6000 forint, microdrive + 2 cartridge 7000 forint, interface II. 3000 fo-

rint, botkormány 1500 forint, mintegy 100-féle játék és felhasználói program, kb. 1000 oldal magyar és német nyelvű irodalom 2500 forint. Ha együtt vásárolja meg, akkor a programok és a könyvek ingyen. Pető József, Győr, Ságvári Endre u. 4. 9026.

● **C64-ES** játékok és másolóprogramokat cserélnék, elsősorban magnót használókkal. Lehöcz Rudolf, Kaposvár, Kinizsi lt. 32.

● **C64-ES GÉPHEZ** programokat, programleírásokat cserélek kazettán és lemezen egyaránt. Mosolygó Csaba, Nyiregyháza, Smidth M. u. 7. Telefon: (42) 13-072.

● **C64-ES PROGRAMOKAT** cserélek. Németh Zsolt, Zalaegerszeg, Atalzséget u. 69.

● **COMMODORE 116-OS** számítógép olcsón eladó. Nyéki Csaba, Hernád, Deák Ferenc u. 2. 2376.

● **16 K-S ZX 81 KLAVIATÚRA** kiegészítéssel, programokkal 7000 forintért eladó. Tel.: 588-731.

C16-ra

FAL

A program a közismert Fal-játék leegyszerűsített változata. Monitor üzemmódban kell beírni az M utasítás segítségével a S1700 címtől kezdve. A játék vagy a G19C0 monitorutasítással, vagy annak BASIC megfelelőjével (SYS 6592) indul.

A játék célja a fal lebontása. Az ütőt a "." és a "," gombokkal irányíthatjuk jobbra és bal-

ra. Ha eközben a SHIFT billentyűt is nyomva tartjuk, az ütő kétszer olyan gyorsan fog mozogni. Az F3 gombbal a fal színét változtatathatjuk, az F2-vel az ütő sebességét állíthatjuk. A játék az F1 gomb le nyomására indul.

TASS ZOLTÁN
Berzsenyi Dániel
Gimnázium

>1700 CE 72 1A D0 09 AD 73 1A
>1708 8D 72 1A 20 11 17 4C 0E
>1710 CE A5 EC C9 26 F0 03 20
>1718 1A 17 A5 06 C9 2F D0 0C
>1720 CE 6B 1A 10 18 A9 00 3D
>1728 6B 1A F0 11 C9 2C D0 0D
>1730 EE 6B 1A A9 24 CD 6B 1A
>1738 B0 03 3D 6B 1A A9 20 AE
>1740 6B 1A 9D 97 0F 9D 9C 0F
>1748 A9 77 9D 98 0F 9D 99 0F
>1750 9D 9A 0F 9D 98 0F 0E
>1758 78 1A AE 78 1A 10 95 A2
>1760 04 8E 78 1A 8E 70 1A A2
>1768 01 8E 71 1A A9 09 8D 6E
>1770 1A A9 28 8D 6C 1A 20 7F
>1778 17 20 A9 83 17 4C 50 18 A9
>1780 D8 A0 0B 85 03 84 04 AD
>1788 6C 1A 4A 18 65 03 85 03
>1790 AE 6E 1A E8 A9 28 18 65
>1798 03 85 03 90 02 E6 04 CA
>17A0 D0 F2 60 A0 00 A9 51 91
>17A8 03 60 A0 00 B1 03 60 AD
>17B0 6C 1A 8D 6D 1A AD 6E 1A
>17B8 8D 6F 1A 60 AE 6D 1A AC
>17C0 6F 1A AD 6E 1A 8D 6F 1A
>17C8 AD 6C 1A 8D 6D 1A 8C 6E
>17D0 1A 8E 6C 1A 60 A9 FC 25
>17D8 03 85 03 A0 03 A9 20 91
>17E0 03 88 10 FB A5 04 18 69
>17E8 FC 85 04 A0 03 A9 9E 91
>17F0 03 88 10 FB A2 02 FE 74
>17F8 1A BD 74 1A C9 0A D0 8E
>1800 A9 00 9D 74 1A CA 10 EE
>1808 60 AD 71 1A 49 01 8D 71
>1810 1A 60 A9 04 38 ED 70 1A
>1818 8D 70 1A 60 AC 6F 1A AD
>1820 71 1A D0 09 88 10 07 20
>1828 09 18 4C 10 18 C8 8C 6E
>1830 1A 60 AE 6D 1A CA CA
>1838 AC 70 1A E8 88 10 FC 8A
>1840 10 06 20 12 18 4C 32 18
>1848 4A C9 28 90 08 A9 4F 8D
>1850 6D 1A 20 12 18 4C 32 18
>1858 8E 6C 1A 60 20 AF 17 20
>1860 10 18 20 7F 17 20 AA 17
>1868 C9 51 F0 0D C9 20 F0 09
>1870 20 D5 17 20 09 18 4C 5F

>1878 18 20 32 18 20 7F 17 20
>1880 FA 17 C9 51 F0 0D C9 20
>1888 FD 09 20 D5 17 20 12 18
>1890 4C 79 18 AC 6E 1A C0 16
>1898 D0 29 20 09 18 AD 6C 1A
>18A0 4A 38 69 00 38 ED 6B 1A
>18A8 90 72 C9 06 B0 6E AE 70
>18B0 1A 18 69 06 CA 10 FB E9
>18B8 05 AA BD 3E 19 8D 70 1A
>18C0 20 32 18 20 BC 17 20 7F
>18C8 17 A0 00 A9 20 91 03 20
>18D0 BC 17 20 7F 17 20 A3 17
>18D8 AD 28 00 00 88 D0 FD CA
>18E0 D0 FA A2 C8 A9 A0 D0 9F
>18E8 0C F0 2E CA D0 F8 A2 01
>18F0 20 F6 17 A2 00 A9 20 9D
>18F8 00 0C 9D 00 0D CA D0 F7
>1900 20 59 19 AD D9 18 38 E9
>1908 85 8D D9 18 A0 00 A2 00
>1910 88 D0 FD CA D0 FA 4C 57
>1918 17 4C 5C 18 A2 77 A9 20
>1920 9D 48 0F CA 10 F8 CE 77
>1928 1A F0 0D A2 00 A0 00 88
>1930 D0 FD CA D0 FA 4C 57 17
>1938 04 D9 19 00 00 00 00 01
>1940 04 00 01 01 01 02 04 00
>1948 01 02 02 03 04 00 02 03
>1950 03 03 04 00 03 04 04 04
>1958 04 A2 C8 A9 A0 9D 9F 8C
>1960 CA D0 FA A9 05 8D 6B 1A
>1968 A0 C7 AE BE 19 AD BF 19
>1970 8D BE 19 8E BF 19 A2 0A
>1978 8E 69 1A A2 04 AD BE 19
>1980 99 A0 08 88 CA D0 F9 AE
>1988 BE 19 AD BF 19 8D BE 19
>1990 8E BF 19 CE 69 1A D0 E3
>1998 CE 68 1A D0 CD 60 06 32
>19A0 20 2D 20 15 14 0F 13 05
>19A8 02 05 13 13 05 07 20 3A
>19B0 10 0F 0E 14 06 31 20 2D
>19B8 20 13 14 01 12 14 40 46
>19C0 78 A9 00 A0 17 8D 14 03
>19C8 8C 15 03 A9 11 8D 6B 1A
>19D0 A9 01 8D 72 1A 8D 73 1A
>19D8 58 A9 93 20 D2 FF 20 59
>19E0 19 A2 12 BD 9D 19 9D 97
>19E8 0D CA D0 F7 A2 0A BD B3
>19F0 19 9D E7 0D CA D0 F7 A2
>19F8 04 BD AF 19 9D 3B 0E CA
>1A00 D0 F7 A9 3A 38 ED 73 1A
>1A08 8D AB 0D A2 02 BD 74 1A
>1A10 18 69 30 9D 38 0E CA 10
>1A18 F4 A2 28 A0 00 88 D0 FD
>1A20 CA D0 FA A5 C6 C9 06 D0
>1A28 06 EE BE 19 4C DE 19 C9
>1A30 04 D0 03 4C 47 1A C9 05
>1A38 D0 DF CE 73 1A D0 A2 A2
>1A40 09 8E 73 1A 4C E1 19 A2
>1A48 C8 A9 20 9D 90 0D CA D0
>1A50 FA A9 05 8D 77 1A A2 02
>1A58 A9 00 9D 74 1A CA 10 FA
>1A60 A9 2D 8D 79 18 4C 57 17
>1A68 00 00 00 00 00 00 00 00
>1A70 00 00 00 00 00 00 00 00

A játékprogramozás technikája

HELIKOPTER

Első cikkünkben a játékprogram egy-egy rövid részletet keresztül megkezdjük a játékprogram fő részeinek és azok fő funkcióinak ismertetését. Az esetleges előzetes, a játékprogram tényleges indítása előtti tevékenységekkel, a címlap és a rajzelemek előállításával foglalkozunk. Most ez utóbbit folytatjuk.

Az eddig megismert rajzelemek egy-egy mozgatandó ábrarészletet jelentettek, úgy mint

- a balra fordított helikopter képe (CL-változó),
- a jobbra fordított helikopter képe (CR-változó),
- a zuhanó emberke képe (M-változó),
- a szembefordított emberke képe (CM-változó),
- az álló emberke képe (M1-változó),
- a fekvő emberke képe (M2-változó),
- a balra haladó emberke képe (M4-változó),
- „Veszély” felirat (DA-változó).

Ezeket a rajzelemeket előállítottuk és a memóriában eltároltuk. Megjelenítésükre eddig még nem került sor.

Ahogy az 1. listában láttuk, a program eddigi menetében egy fővonal és annak mellékvonalai voltak megállapíthatók. Ez utóbbiak egy-egy részfeladat megoldására

```
230 GOSUB1620
```

4. lista

```
1670 PMODE4,1:PCLS
1630 DRAW*BM55,20;L15E15R15;BR6;G15E7R15G7E15;
1640 DRAW*BR6;R15G15L15E15R15;BR6;G15E8R15E7L15R15;
1650 DRAW*BR6;G15E8R15E7L15R15;BR6;G15R15L15E8R8L8E7R15;
1660 DRAW*BR6;G15E15R15G7L15R7D8
1670 DRAW*BM55,38;G15R15L15E15R15;BR6;R15G15L15E15R15;
1680 FORT=1T02
1690 DRAW*BR6;G15E15R7G7E7R7G15E15;
1700 NEXTT
1710 DRAW*BR6;G15E15R15G7L15R15G8E15;BR6;G15E15D15E15;
1720 DRAW*BR6;R15G15L15R3E15;
```

5. lista

szolgáló szubrutint jelentettek. A fővonal a 4. listában folytatódik. Az itt szereplő mellékvonalat az 5. lista mutatja. Ahogy a 11. ábrán látható, itt most egy címlaprészletet állítottunk elő. Ez látszólag visszatérés egy már korábbi feladathoz, amikor is szintén egy címlapot állítottunk elő, de az csak a program használatához szükséges rövid ismertető bevezető címlapja volt. Az igazi, a főcímlap most következik.

A főcímlap

Ahogy említettük, az igazi címlap elkészítésének egy részletét láthatjuk az 5. listában. E címlaprészlet előállítása is úgy történik, hogy nem jelenítjük meg az előállított rajzot. A 11. ábra előállításánál is az első részben bemutatott módszerrel jártunk el, vagyis a program kismérvű módosításával

a nem látható rajzelemeket láthatóvá tettük. Itt ez a következő sorok beírásával

```
1625SCREEN1,1
```

```
1725RETURN
```

```
és a
```

```
RUN230
```

beadásával történik.

Ez a részlet eddig ismeretlen utasítást nem tartalmazott. Most egy újserű motívum következik: a már rendelkezésünkre álló rajzelemek mozgatása.

A rajzelemek mozgatása

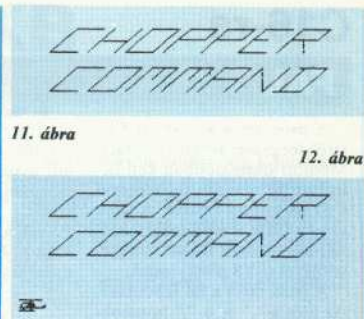
Elsőként a balra fordított helikoptert vesszük elő, és megfelelő sebességgel a képernyő adott magasságú helyétől vízszintes irányban balra, különböző helyekre viszük. A megfelelő sebesség itt azt jelenti, hogy ezt olyan gyorsan kell végrehajtani, hogy ne külön-külön megjelenjen egyes helikoptereket lássunk, hanem úgy tűnjön, mintha a helikopter jobbról balra haladna. A konkrét megoldás a 6. listában látható, melynek első részlete (1730—1760-as sor) valószínűleg ezt meg. Természetesen ennek a „mozgásnak” csak egy kiragadott részletét tudjuk nyomtatón megjeleníteni. Ezt az 1760-as sor következő módosításával érhetjük el:

```
1760NEXTT:RETURN
majd
RUN
```

Ennél a részletnél egy eddig nem ismer-

6. lista

```
1730 SCREEN1,1
1740 FORT=215T020STEP-2
1750 PUT(T,70)-(T+23,85),CL,PSET
1760 NEXTT:PUT(20,70)-(43,85),CH,PSET
1770 FORT=1T05;FORTH=230T0 T*9+180STEP-2
1780 PUT(TN,70)-(TN+5,83),M4,PSET
1790 PLAY*T3002C
1800 LINE(TN,70)-(TN+6,83),PRESE T,BF
1810 PUT(TN,70)-(TN+5,83),M1,PSET
1820 PLAY*T3001C
1830 LINE(TN,70)-(TN+7,83),PRESE T,BF
1840 NEXTTN
1850 PUT(TN,70)-(TN+5,83),M1,PSET
1860 NEXTT
```



11. ábra

12. ábra

tett utasítást is használunk, ez a PUT. Ez az utasítás egy átlójának végpontjaival megadott négyszöget vesz elő a memóriából, mégpedig oly módon, hogy ez a négy- szög az utasításban megadott változó névvel a memóriába eltárolt négy- szög vagy annak részlete, melyet a képernyő valamely tetsző szerinti helyére helyezünk. A négy- szög egyszerű elővételén kívül az egyes pontokat logikai kapcsolatba is hozhatjuk a képernyőn a már eddig meglévő pontokkal, ugyanebben az utasításban. Így lehetősé- günk van arra is, hogy meglévő képeket módosítsunk.

A mozgás egy újabb elemét tudjuk elő- állítani, ha a megfelelő helyre eljuttatott heli- kopter mintegy szembefordított ázaltal, hogy ezen a helyen a szembefordított heli- kopter jelenítjük végül meg. Ez az 1760-as sor utolsó utasítása.

Ennek megjelenítése az 1760-as sor víz- számosításával,

```
1765RETURN
```

beadásával és a program indításával törté- nik. Mindez a 12. és 13. ábrán látható.

Ugyancsak a már rendelkezésünkre álló rajzelemek látszólagos mozgatásával állit- tuk elő az emberek felsorakoztatását. Az 5 emberke bejön jobbról balra, majd egymás

```
1870 PLAY*T403U3L4L1;FORT=1T02;
:PLAY*L4B-AB-AB-AL2GL4AGAGAGL4.F
L8AL4GFGFGL4.E8L6L4FEFEFGL2AL1A
L2AL4CDDCDL2L4CDDCDL2L4CACDF
GFDL1DL2.L4C:NEXTT:PLAY*L1DL2.
L4CADFL2DDDL2.D
1880 FORT=1T02000:NEXTT:RETURN
```

7. lista

```
240 SC=0;LI=3;LV=1;MS=1:NF=10
250 GOT02160
```

8. lista

```
2160 CLS:PRINT"SCORE=";SC;:LIVES
=";LI
2170 PRINT:PRINT:PRINT:PRINT*HIG
H SCORE=";HS
2180 PRINT*236;"LEVEL=";LV
2190 PRINT*324;"PRESS ENTER TO C
ONTINUE"
2200 INPUTZX$
2210 IF MS=6THEN MS=1:NF=Nf-1:SC
=SC+300
2220 IF NF<=2THEN NF=10:MS=1:SC=
SC+1000
2230 GOT0260
```

9. lista

CHOPPER
COMMAND

13. ábra

吳吳吳吳吳

14. ábra

CHOPPER
COMMAND

吳吳吳吳吳

15. ábra

16. ábra

SCORE= 1000 LIVES= 0

HIGH SCORE= 0

LEVEL= 0

PRESS ENTER TO CONTINUE

mellett felsorakozik. A megoldás részlete a 14. ábrán, a teljes főcímlap a 15. ábrán látható. Az emberkék mozgását már egy új elem, a hanghatás segítségével tesszük még érdekesebbé. A hanghatás nem dallam, hanem pusztán zörej. Mindez az 1770—1860-as sorig valósul meg. A rajzok előállításához az 1765-ös sort törölni kellett, és be kellett adni a következőket: 1625SCREEN1,1:GOTO1770 1865RETURN

majd a programot újra indítjuk.

Itt két új utasítást használunk. Az első, a rajzolóhoz használt másfajta utasítás a LINE. Ennél egy egyenes szakasz felrajzolása az egyik lehetőség, melynél az egyenes szakasz végpontjainak koordinátáit és azt kell megadnunk, hogy látható vagy láthatatlan vonalat rajzoljunk-e. A másik lehetőség egy téglalap rajzolása, melynél a téglalap belseje törölhető vagy sem (BF használata a törlést jelenti).

A másik használt utasítás a PLAY, amely dallamok lejátszására és előállítására szolgál. A dallamot egy karakter sorozat adja meg, mely a DRAW utasításhoz hasonlóan utasításokat tartalmaz.

Az alutasítások a következők lehetnek: — hangjegyek (a skálától eltérően nem h, hanem b szerepel, és a zenei kereszt helyett + jelet is használhatunk, míg a zenei b helyett csak a — jel a megengedett), — az oktávok jelölésére az O betű és utána egy 1 és 5 közti szám áll, — a hang hosszát L betű jelöli, utána egy 1 és 255 közötti szám áll, mely azt jelzi, hogy az egész hang hányad részét adtuk meg (a pontozás itt is használható),

- a tempó (T) és utána egy 1 és 155 közötti szám a dallam lejátszásának sebességét jelzi,
- a hangerő (V), utána egy 1 és 31 közötti szám,
- a szünet hossza (P) és egy 1 és 255 közötti szám,
- a szünet utasításnál használt X itt is használható.

Az oktávot, a hangerőt, a tempót, valamint a hanghosszt a következőképpen lehet módosítani: a + jel egygel növeli, a — egyvel csökkenti, a > megkétszerezi, a < felezi az eddigi értéket.

Hangeffektusok

Az előbb ismertetett utasítás segítségével nemcsak egyszerű zörejeket, hanem teljes dalokat is előállíthatunk és eljátszhatunk. Ez minden számítógépes játék fontos része.

Itt ezt a 7. listában található programrészletet valósítja meg. A dal lejátszhatjuk a következők programmódosítással:

140GOSUB 1870:STOP

és a program futtatásával. (Ismét felhívjuk és a figyelmet arra, hogy a játék használatá-

hoz minden módosítást előzetesen vissza kell változtatnunk!)

A bevezetés vége

A 8. listában beállítunk néhány alapparamétert (a pillanatnyilag elért pontszámot: SC, a még rendelkezésre álló helikopterek számát: LI, a játék bonyolultságai szintjét: LV stb.), mely a már beállítottal (az eddig elért maximális pontszám a 160-as sorban beállított HS) együtt lehetővé teszi a játék megkezdését.

Eredménykijelzés

A játék egy nélkülözhetetlen mozzanata az eredmények közlése a játékkal. A 9. lista és a 16. ábra alapján látható, hogy ez az elért pontszámunk, a még meglévő helikopterek számának és az eddigi legnagyobb pontszámunk a kijelzéséből áll ebben az esetben. Gyakran sikerlistákat is megadnak, felírják a résztvevők nevét stb. A programnak ebben a részletében történik meg a pontszámok változtatása is.

GALATINA FERENC

PROGRAMOZÓK MÉSÉLIK



A Novotrade számítógépjátékait eddig főként Nagy-Britanniában kedvelték. Nemrégiben sikerült néhány játékkal az Egyesült Államokban megjelenni a piacon, de az európai kontinens meg meghódításra váró terület.

Legutóbb egy francia cég megvette néhány, Angliában már lefutott játék forgalmazási jogát — elsősorban olyanokét, amelyeket Commodore 64-esen és Amstradon is elkészítettek a magyar programozók, mivel az Amstrad jóval elterjedtebb gép Franciaországban.

A játékok mesterlemezeit kikerítették a francia céghez, és hamar kiderült, hogy gondok vannak: saját erőből nem tudják a kiküldött magnelemezről kazettára írni a programot. Nem maradt más megoldás, mint kiutazni és segíteni. Majsza Katalin és Kiss Donát megérkezett a cég Párizs melletti telephelyére, és munkához láttak. Az Amstrad-játékokkal hamar végeztek, hiszen Majsza Kati hónapok óta intenzíven foglalkozik a géppel és ezekkel a játékokkal. Kiderült azonban, hogy — amiről addig szó sem esett — a Commodore-játékokkal sincs minden rendben. Ott a kazettaverzió ment simán, viszont az egyik játék sehogysem akart lemezzel futni.

Ekkor következett az a telefonbeszélgetés, amelynek hallatán Commodore-programozók talán felfigyeltek a nevesítésre, de mivel akkor komoly tétre ment a dolog, senkinek sem volt kedve mulatni.

A vonal egyik végén Kiss Donát Franciaországból, a másikon a sorok írója, aki véletlenül

(és egymagában) e késői órán, este 8 körül is még a stúdióban tartózkodott. Donát gyakorlati programozó, Commodore-on azonban egy 20 GOTO 10 nem sok, annyi programot sem írt. Jómagam a bölcsész végzettségemmel a BASIC-ben már úgy-ahogy eligazodom, de gépi kódú programba mentőöv és oxigénálarék nélkül eddig még sohasem merészkedtem.

Egyeszkor lett volna a megoldás, ha sikerül megtalálni a játék programozóját, de hát Murphy törvénye a számítógépszakmában fokozottan érvényesül: az illető nem volt otthon. Magunkra voltunk utalva, nekünk kellett megtalálni a tüt a szénakazalban. Feltévesünk szerint egyetlen 01 értékű bajt okozta a problémát; ezt kellett megtalálnunk egy 14-15 kb-ajit-os programban, 01 értékű bajtók tengerében, interubán telefonvonalon keresztül, egy mindkettőnk számára csak látásból ismerős diszsembler program segítségével.

Nem csigázom tovább a feszültséget: végül sikerült, meglett az a bizonyos 01-es bajt, elindult a játék Franciaországban is, itthon is. Közben persze kilókat fogytunk, és a hajunk öszbe csavarodott.

Utólag végiggondolva a történetet, a szerencse kétszeresen is a kegyeibe fogadott: egyszerűen valóban azon az egy bajton mulott a dolog (hiszen megeshetett volna, hogy a lemez szállítás közben megsérül, és bajtók százai válnak hibássá), másrészt a telefonszámlát a francia partner fogja kifizetni...

RÉVBÍRÓ TAMÁS

A VIDEOTON TV-COMPUTER

Az iskolaszámítógép-pályázaton a TVC (a gyártó által használt rövidítés) az általános iskolai kategóriában megszavazta első díjat nyert, a középiskolai kategóriában második lett. A két kategóriában indított változatot egyébként csak a RAM méretében különböztetik.

A gyártó elsősorban tömeges felhasználásra: oktatásra, játékra, egyéb házi használatra készítette gépét, de a bővítés lehetőségeit kihasználva, tervezte egy kisebb, professzionális feladatok elvégzésére is alkalmas rendszer kiépítését.

A készülék felépítését vizsgálva szokatlannak tűnik a jobb oldalon látható botkormány. A hazai gyártmányoknál ugyancsak szokatlannak, több buszcsatlakozóval ellátott kivétel és a botkormány miatt a gép az általában szokásos háziszámítógép-méretnél nagyobb. Az 1. és 2. képen bemutatott nézetek alapján ránézéssel is megállapítható, hogy a gép viszonylag sok periféria csatlakozását teszi lehetővé.

A billentyűzet

A billentyűzet 66 nyomógombot tartalmaz. Ez az első, hazai tömeggyártásba került mikroszámítógép, amelyen az összes magyar ékezetes karakter fellelhető. A billentyűzet részét képezi a fénypont nyolc irányú mozgatható szolgálat, de játékoknál is használható botkormány.

A billentyű lenyomásokor a billentyűzet alatt elhelyezett nyomtatott áramköri kártya aranyozott, fésűs érintkezőit vezetőgumi zárja. Így képezik azokat a jeleket, amelyeket a számítógép a billentyű azonosítására használ. A gyártó tájékoztatása szerint a billentyűk élettartama átlagos körülmények között több mint 5 millió leütés. Ellenben az eddig itthon gyártott billentyűkkel, a feliratok kettős fröccsöntéssel készültek, így nem veszítik el tartós használat után sem olvashatóságukat.

A beépített botkormány a billentyűzethez hasonlóan membránkapcsolós, várható élettartama nem túl nagy. A bot letörésének megakadályozására alkalmazott felütökötéses megoldás az üzletben gyakran kapható legolcsóbb botkormányokénál várhatóan nagyobb élettartamot biztosít.

A billentyűzeten található vezérlőgombok közül a SHIFT funkciója közzismert, a LOCK az írógépek egy részénél használt UPPER CASE rögzítő, a CAPS pedig az előző hatásához hasonló, de csak a kisbetűk váltódnak át. Teljesen újszerű az ALT gomb hatása: lehetővé teszi a karakterek egy részének átirását és további új karakterek készítését. 32 karakter írható át, és to-

vábbi 64 definiálható. Az átirás is és az új karakterek írása is oly módon történik, hogy az erre szolgáló utasítás segítségével közöljük a számítógéppel, hogy milyen kétszámjús karaktert adunk meg, és e karakter tíz sorának milyen értékei lesznek. A karakterek mindegyike egy nyolcszor tízes pontmátrixban helyezkedik el, mely mátrix egyes pontjainak ki-, illetve bekapcsolt voltát kell meghatározni. Az általunk „elkészített” karakterek ezután a megadott kétszámjús történő hivatkozással vagy az ALT gomb segítségével írathatók ki a képernyőre, sztring formában is.

A billentyűzet nyomógombjainak lenyomott helyzetét egy 8×10 -es mátrix 8×8 -as része érzékeli, a megmaradó 2×8 -as mezőt a botkormány használja.

A csatlakozók

A használt csatlakozók aranyozottak, jó minőségűek. Az 1. képen látható egy párhuzamos nyomtatóillesztő csatlakozója, a tápegység csatlakozója, két magnetofoncsatlakozó és egy összetett videojel-csatlakozó, az ún. RGB-kimenet (ez három fősín külön-külön történő kivitelére szolgál), valamint a hagyományos tv-csatlakozó. A 2. kép két, általánosan használt külső botkormány csatlakozására szolgáló csatlakozót és az ún. programmodul-csatlakozót mutatja. Ezek közül különösen előnyösek tartom az RGB-, a két független magnetofoncsatlakozó és a nyomtatóillesztő meglétét, és igen célszerűnek az 1. képen látható négy további buszcsatlakozót, melyek használatával különböző bővítőkártyák csatlakoztathatók további egységek kiépítése nélkül.

A csatlakozók minőségére vonatkozó általános megállapítás mellett meg kell jegyezni, hogy a tv-csatlakozó mechanikai felépítése (rögzítése) nem tűnik időtálló megoldásnak.

A készülék külsején helyezkedik el még egy, a színek ki-be kapcsolására szolgáló nyomógomb, mely megkönnyíti a fekete-fehér tévék használatát. A számítógép alsó felületén, rugalmas lemezként, a doboz anyagából van kialakítva a RESET gomb. Ennek elhelyezését és kialakítását nem tartom szerencsésnek; csak bizonyos gyakorlat után található meg könnyen és kezelhető biztonságosan. Az a megoldás, hogy „rövid idő”-n belül ismétlenül megnyomva, a memóriát is törli, veszélyesnek látszik, mert előfordulhat nem szándékos törlés is (a „rövid idő” a gyártó tájékoztatása szerint 0,5-0,6 másodperc). Programmegszakításra a CTRL + ESC gombokat használva, ez teljesen kizárható.

A színtévé-használatot bizonyos mértékben korlátozza, hogy csak a PAL-rendszerű, illetve a kétszámjús tévék adnak színes képet. A legolcsóbb és igen elterjedt szíves televíziók nem alkalmasak a gép színes képének előállítására, mivel csak SECAM-rendszerűek.

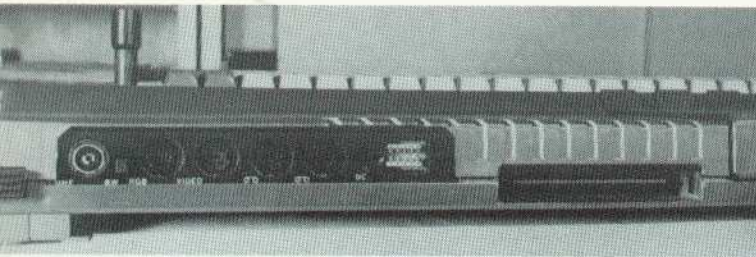
A lábazat gyártás kialakítása nem tűnik szerencsés és tartós megoldásnak. A doboz relatív nagy mérete miatt lehetségesnek tartom, hogy az időjárás változásai miatt bekövetkező összehúzódások, tágulások olyan vetemedéseket okozhatnak, amelyek következtében a doboz megsérülhet, és kárt okozó zárlat léphet fel.

A belső felépítés

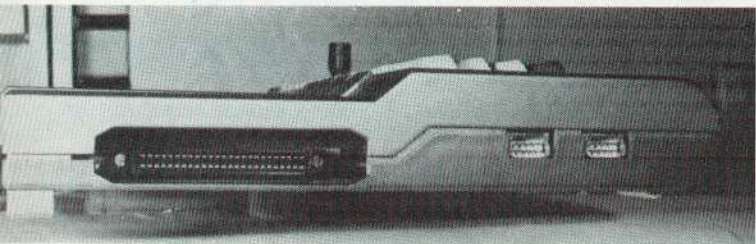
A számítógép belsejében egy nagy-, egy közép- és egy kisméretű kártya található. A közepeméretű (a 3. képen nem látható) a billentyűzet kiszolgálását végzi. Ez egy csatlakozóval kapcsolódik a nagy kártyához, és mechanikailag függetlenített attól. A kisméretű kártya a PAL kódolási funkciókat valósítja meg. Az RGB-kimeneteket használva erre a kártyára nincs szükség. Felépítésében egy szempontból lényegesen különbözik a hasonló célra kidolgozott megoldásoktól: az UHF-modulátor nem a világszerte szokásos Astec gyártmány, hanem egy speciális IC-2 tartalmazó megoldás. A színeződésvivő olyan zavarásait elmennek, amik például a C64 betűszélének, „színeződését” okozzák, olyan megoldást dolgoztak ki, amely — saját tapasztalataim szerint — eredményes volt. A már említett 8×10 -es méretű betűmátrix jobb, mint a Commodore és Sinclair gépek 8×8 -as mátrixa. A nagyobb méret teszi lehetővé a magyar karakterkészletnek a helyesírásához közelebb álló megvalósítását. Ez azt jelenti, hogy a teljes magyar betűkészlet megvan, de a betűk mérete, az ékezetek elhelyezése, a betűk alakja egyedi. A beállító képernyőkijelzési módok közül a 16 karakteres nem elegendés, a 64 karakteres csak közelről olvasható, és néhány betű, például a hosszú ú és hosszú ő összetéveszhető.

E megjegyzéstől eltekintve, a kép fekete-fehérben olvasható mind Junoszy 402D, mind Elektronika C—401m típusú televíziókon is, nemcsak Videoton gyártmányú tévéken. Az ékezetekkel kapcsolatban eltérés a magyar helyesírástól, hogy az Ú kivételével az összes nagybetű pont van (Á, E, Ó, Ű esetében is). Amennyiben Y, Ű, illetve X, Ű vagy Y, Ű kerül egymás alá, úgy összetéveszhető. Ugyanez érvényes az O, Ő, Ő és G betűkre is. A betűk itt is összenyomottak. A képernyőn a szöveges kijelzésen kívül háromféle grafikus kijelzés is megvalósítható. Ezek a címezhető képpontok és a használatához színek számában különbözőnek.

Az órajeleket egy 25 MHz-es kvarccsillátóral állítja elő. Ennek jeléből feleléssel kapják a képelőállításához szükséges alapórajeleket, melyekből további feleléssel lehet megkapni a különböző felbontású alapórajeleket. A megjelenítendő szöveg a kép frissítéséhez a memóriából 640 ns-os egységként olvas ki egy-egy bájtot. Az információt



1. kép



2. kép

ók képernyőre való kivételéhez szükséges adatok a 16 színű üzemmódban másképp tárolódnak, mint a két másikban. Az elsőnél egy képponthez négy bit tartozik, melyből az első három a három alapszínre vonatkozik, az utolsó a színintenzitást adja meg (teljes vagy fél); a két másik üzemmódban az egy, illetve két bit egy ún. palettaregiszterben történő tároláshoz szükséges sorszámot jelez. A kép keretszíne külön kerül tárolásra. Az említett négy bitből adódó három színjel kerül ki az RGB-kivezetésre. A PAL videojelhez színesgédvívó tartozik, melynek jelét fekete-fehér tévé használatakor kapcsolóval ki lehet iktatni, a kapcsoló mindenkor helyzetét a program érzékeli. A színesgédvívó a sorfrekvenciához szinkronizált.

A mikroprocesszor Z80A típusú, 3,125 MHz frekvenciájú járatott. Mivel a memória lehetséges maximális mérete meghaladja a mikroprocesszor által címezhető területet, ezért lapozós technikát használnak. A felhasználói RAM 32-64 kb-ot; a 3. képen látható, 4116 típusú integrált áramkörökkel valósították meg. A képmemória 16 kb-ot. A mikroprocesszor ehhez a memóriaterülethez közvetlenül is hozzáférhet, így azt módosítani is képes. A rendszer ROM 20 kb-ot, és a BASIC-interpretet, valamint az azt kiszolgáló operációs rendszert tartalmazza. Ezt maximálisan 24 kb-ot lehet kiterjeszteni. Programmodulok („dobozolt EPROM”) számára maximálisan 16 kb-ot áll rendelkezésre (ilyenek például a játékgépprogramok, az UPM – CP/M 2.2 – operációs rendszer). A perifériákhoz tartozó memóriák a külső kártyákon helyezhetők el, kártyánként maximálisan 8 kb-ot nagyságban. A BASIC használata esetén a ténylegesen felhasználható memóriaterület 25–41 kb-ot. A négy lapra osztott memóriacímterület első lapján a rendszer ROM,

a felhasználói RAM eleje, a programmodul ROM helyezkedik el. A második és a harmadik lapon a felhasználói RAM további részei vannak, ez utóbbin a képernyő RAM is. Az utolsó lapon a maradék található.

Az összes órajel egy alappfrekvenciából állítják elő, osztólánc segítségével. Az adatbusz pufferelt, a busz többi része nem. Az összes kártyát lakkozás védi a por és légszennyezés ellen.

A hangkeltő áramkör egy 12 bites programozható számlánc, mely a mikroprocesszor órajelét osztja le. A számláncot 16-os leosztás követi. A számlánc kimenete négy bites amplitúdószabályozó áramkörön keresztül kerül a kimeneti csatlakozóra, illetve az UHF-modulátor bemenetére. Az amplitúdószabályozó bemenetére folyamatosan „1”-es szint is adható, ekkor D/A konverterként működik.

A kazettás egység jelét billenőkör állítja elő. A jel frekvenciája adja meg információtartalmát. A kazettán a jel formátuma a következő: egy 4 s hosszúságú, 470 µs periódusidejű, négyszög alakú szinkronizáló jelsorozat, melyet egy 740 µs periódusidejű négyszögimpulzus zár. Az átviendő adatok 256 bájtanként 2 bájtos CRC-vel (Cyclic Redundancy Check) ellenőrzöttek. A „0” 555 µs periódusidejű négyszögimpulzus, az „1” pedig 395 µs. A szinkronizáló célja átlagssebesség-meghatározás, ehhez állítja be a számítógép a frekvenciát. A két magnetofoncsatlakoztatási hely egymástól függetlenül, programból kiválaszthatóan működtethető. Bekapcsoláskor vagy RESET után a tápcsatlakozóhoz közelebb levőt címezi a BASIC interpreter. A másik csatlakozó START/STOP áramköre POKE utasítással működtethető.

Három magnetofonnal vizsgáltam, hogy a gyártó által a géphez adott DEMO-kazettáról hogyan olvas, illetve ír. A gyártótól

kapott BRG MK—29 típusúval mindkettő hibátlan volt, egy Panasonic (Japán) R 5090 típusúval 6 olvasásból 4 sikerült (az írás kábelhiba miatt nem volt vizsgálható), egy Centrokord (NSZK) 250 típusúval teljesen eredménytelen volt. Nem maximális hangerőnél semmit sem olvasott be, maximálisan a fejlecezt általában igen, de néhány blokk beolvasása után az ellenőrzőszám hibája miatt leállt.

Bővítő kártyák

A gép bővítésére négy beépített kártyacsatlakozó szolgál. Ezekre vezették ki a buszt; ezek a pozíciók BASIC utasítással elérhetők el. Az egyes csatlakozók egy két bites számmal közvetlenül megcímezhetők, vagy a BASIC-ből is elérhetők. Eddig RAM-bővítés, lemezegység-csatoló, soros illesztő, általános célú párhuzamos illesztő készült el.

Rendszerprogramok

A BASIC-interpretet támogató operációs rendszer számos hozzáférhető szubrutint tartalmaz. Ezek egy 8 bites funkciókód használatával elérhetők. A funkciókódból az alsó hét bit kerül kihasználásra, melyek közül a felső három a szubrutin osztályát határozza meg (például a 0 a video, az 1 a billentyűzet stb.), az alsó négy pedig a konkrét funkciót adja meg (például a 0 a megszakítást, az 1 a karakteres ki-be menetet).

BASIC

Dr. Tóth

A BASIC számos tulajdonságában eltér a világszerte legerjedtebb és itthon is gyakran használt MICROSOFT típusútól. Bizonyos tulajdonságai hasonlítanak a Spectrum típusúhoz. Néhány előnyös tulajdonsága:

- a LIST parancsot tartományosozott is követheti,
- a DELETE parancs létezése, melynél tartományosozott is használható,
- a RESTORE sorszám létezése,
- a 250 karakter hosszúságú utasítássor és a 254 karakter hosszúságú karakterlánc-hossz,
- a sokféle paraméterrel (szín-, vonaltípus-, vonalkeresztelés-, időállandó-, holtidő-beállítás, karakterdefiníció) ellátható SET létezése,
- a PLOT utasításnál több vonalhúzás is összefogható (kár, hogy nem használhatóak relatív elmozdulások, koordinátatengely és mellékírny felé tett elmozdulásokra egykarakteres alutasítások, elforgatás, nagyítás),
- 14-féle vonalhúzás állítható be, és a vonalkeresztelés eredménye logikai függvénykapcsolattal meghatározható,
- a botkormányok az INKEY\$ utasítással olvashatók be.

Felsorolok néhány hátrányt is: — a hangelölítésnél a szünet hossza nem szabályozható, a zenei skálahangok helyett frekvenciákat kell megadni,

— a kompatibilitás hiánya más BASIC-interpreterekkel.

Az alábbiak előnyös vagy hátrányos volta vitatható:

— a kiíratásnál nem törli a sort, amelybe ír, de törli az utána következő karaktert,
— ha a TAB értéke kisebb, mint az előző PRINT által kiadott karakteresorozat vége, akkor ebbe a sorozatba beírja a következőt,

— a NEW és LOAD hatására a LO-MEM, amellyel a gépi kódú programnak helyet biztosítottunk, alapértékére áll vissza, de a VLOMEM tartalmának átírásával ennek hátránya megszüntethető. NEW és LOAD esetén a LOMEM a VLOMEM értékét veszi fel,

— a RESET hatására a négyszínű üzemmódba ugrik, de CTRL + ESC-nél nem!

— az INPUT PROMPT forma,

— az INPUT-nál az, hogy hibás adatformátum „0”, illetve „ ” beadását jelenti,

— az egymásba ágyazott IF... THEN hatástalanítja (az első kivételével) az ELSE-t,

— a FOR... NEXT ciklusból a kiugrás megengedett,

— az ASC-függvény nevének ORD-ra változtatása.

A dokumentáció

A géphez két könyvet kap a vásárló: a Kezelési útmutatót és a BASIC programozási segédletet. Ezek a felhasználó részére igen sok, jól használható adatot tartalmaznak, az eddig itthon megszokottaknál lényegesen jobbak. A Kezelési útmutató általában jól követhető, érthető, alapos. Megemlítek azonban néhány olyan hibát, amely a hibajegyzékben nem szerepel és zavaró.

A 15. oldalon leírtak ellenére a SHIFT gomb lenyomása nem váltja vissza a fénypontot alaplómódba, ezt csak a LOCK gomb lenyomása eredményezi.

A 22. oldal első példájában a LOG nem „10-es alapú logaritmus”-t jelent, hanem — ahogy ezt a példa eredménye is bizonyítja — ún. természetes alapú logaritmust.

Az Ismerkedés a BASIC-kel című fejezet valódi kezdőknek nehéz, a példák egy része egy átlag kezdő számára nem érthető. A már említett hibás példa is ilyen. Igen előnyösnek tartom a csatlakozókiosztás ismertetését (kár, hogy jelmagyarázat, terhelési adatok nélkül), az egyéb műszaki adatok közlését, a billentyűzetábrák és a karakterkód-táblázat megadását.

Megtévesztő lehet viszont a BASIC elitéréseit tartalmazó fejezet. Ezt olvasva ugyanis az olvasó azt hiszi, hogy itt megtalálja az összes eltérést, azonban nem szerepelnek olyan eltérések, mint például a perifériára írásnál a PRINT n: formátumban használatos „:”, az INPUT PROMPT forma létezésének említése és ebben a „:” használata, az INPUT-nál az, hogy hibás adatformátum „0”, illetve „ ” értéket okoz, az egymásba ágyazott IF... THEN hatástalanítja (az első kivételével) az ELSE-t, a FOR... NEXT ciklusból engedélyezett a kiugrás, nincs említve az ON utasításban

MIKROPROCESSZOR ÉS MEMÓRIA

Típus:	Z80A
Órajfrekvencia:	3,125 MHz
Rendszer ROM (OS+BASIC):	20 kbájt (max: 24 kbájt)
RAM memória:	332 kbájt (64 kbájtig bővíthető)
Video RAM:	16 kbájt
I/O ROM + RAM:	max. 8 kbájt/kártya (max. 4 kártya)
Programmodul:	max. 16 kbájt ROM

KÉPERNYŐSZERVEZÉS

240 × 128 pont, 16 szín (24 × 16 karakter)
240 × 256 pont, 4 szín (24 × 32 karakter)
240 × 512 pont, 2 szín (24 × 64 karakter)

A két- és négyszínű üzemmódban az aktuális színek tizenhatos színkészlétből választhatók. Külön állítható a képernyő színe (tizenhat szín).

KARAKTERKÉSZLET

10 × 8-as teljes pontmátrix
96 definiált, nem módosítható karakter
32 definiált, újradefiniálható karakter
64 nem definiált, a felhasználó által definiálható karakter

HANG

Frekvencia (Hz):	195312,5 (4096-P), ahol P=0...4094
Amplitúdó:	tizenhat fokozatban programozható

BILLENTYŰZET

57 alfanumerikus nyomógomb
kilenc szerkesztő-, funkció- és kódmódosító nyomógomb
beépített kis botkormány (négy irány)

KI- ÉS BEMENETEK

Antennacsatlakozó

Képvivő:	UHF 25...40 csatorna között
Hangvivő:	Képvivő + 6,5 MHz, FM
Szín:	PAL-rendszerű, a színsegédvívő kikapcsolható

RGB-kimenet

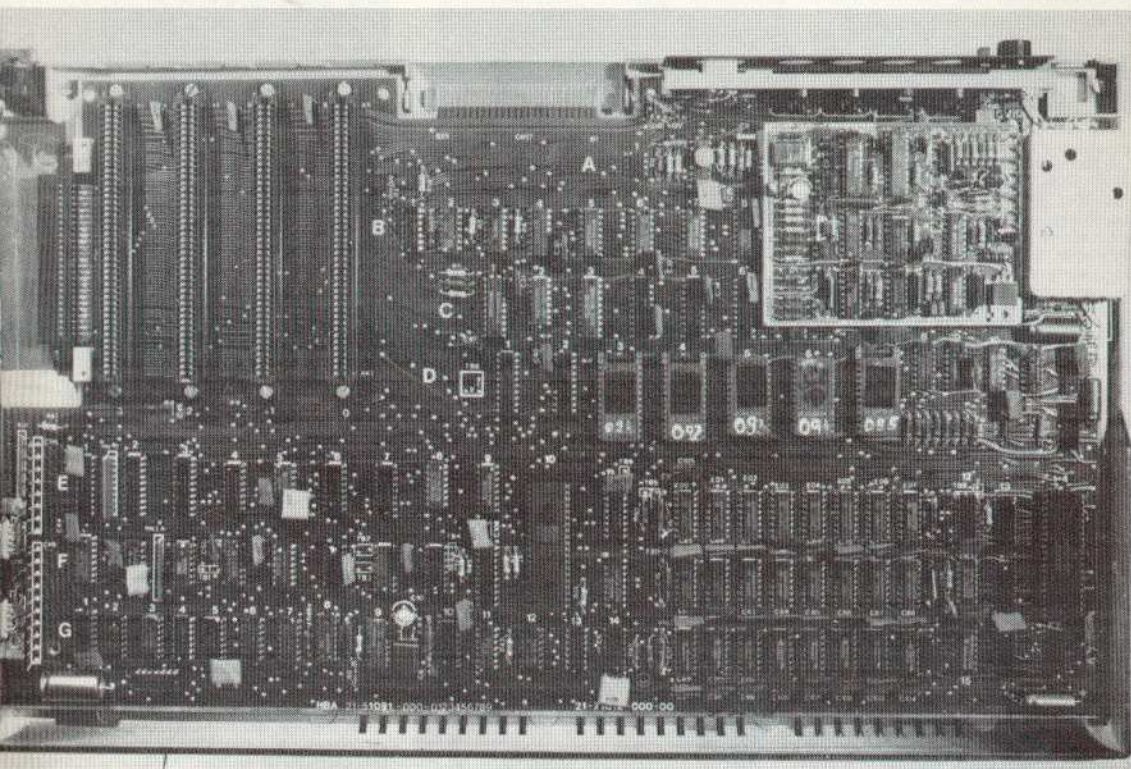
A G, R, B jelek szintje	75 ohmmal lezárva:
— a teljes fényű színekénél	1 V
— a sötét színekénél	0,5 V
— fekete szint:	0 V
Polaritásuk	pozitív
Szinkronjel:	1,4 Vpp 75 ohmmal lezárva, polaritása negatív

Összetett videojel-kimenet

Videojel:	— amplitúdó: 1 Vpp 75 ohmmal lezárva
	— polaritás: pozitív (szinkron negatív)
	— színkülönbségi jel: PAL-rendszerű, kikapcsolható
	— a nem leválasztott kimeneten a fekete szint 75 ohms lezárás esetén kb. +2 V
Hang:	— a maximális kimenőszint 1,1 Vpp

Magnócsatlakozók

Kimenet:	200 mVpp
Bemenet:	0,8...5 Vpp
Motorvezérlő jel:	földelt emitteres tranzisztor kollektora van kivézetve, a tranzisztor határadatai:
	Ucmax: 30 V
	Icmax: 300 mA



3. kép

az ELSE léte, az ASC-függvény nevének ORD-ra változtatása.

A BASIC programozási segédlet kezdőknek érthetetlen, mert számos fogalom ismeretét feltételezi. Más BASIC-változatokat ismerők számára azonban jól használható. A gyakorlottabb programozók számára előnyös a BASIC-változók és munkaterületek egy részének, az adatábrázolás módjának bemutatása. A gépi kódú programozást elősegíti a Z80 utasításainak ismertetése.

A BASIC egyik jellemző adata a futtatási sebesség. A közismert KILOBAUD-teszt eredményét a 2. táblázat tartalmazza, a korábban már más helyen közölt néhány gép-re vonatkozó adatokkal együtt.

Összefoglalás

A gép az itthon használt más gépekkel összehasonlítva egyes tulajdonságaiban, melyeket korábban ismertettem, lényeges előnyöket mutat, és különösebb hátrányát nem találtam. Meg kell azonban jegyezni, hogy tartós üzemi vizsgálat végzésére még nem volt lehetőségem. A dokumentációt az itthon megszokottaknál lényegesen jobbnak találtam. A gyártók eddig már jó néhány programot kifejlesztettek, például játékprogramokat, műszaki, irodai alkalmazásokra szánt programokat. A hardver-

GÉPTÍPUS	PROGRAMSORSZÁM							
	1	2	3	4	5	6	7	8
C64 ¹	1,6	9,7	18,3	20,3	21,8	31,5	49,5	115,9
Spectrum ¹	4,8	8,7	21,1	20,4	24,0	55,3	80,7	253,0
Atari 400/800 ¹	2,3	7,4	19,9	23,2	26,8	40,7	61,5	431,0
Aircomp 16 (gyors üzem) ¹	0,7	4,0	8,0	8,0	9,5	15,0	23,0	30,0
Primo ²	1,4	9,2	17,6	20	20	32	51	84
Dragon (gyors üzem) ²	0,7	5,1	9,9	10,7	11,7	17,9	25,0	64,0
TVC	1,9	9,0	21	28	30	44	58	250

¹ ÖTLET-BITLET 1984. január

² Mikromagazin 1984. 6. szám

2. táblázat. A KILOBAUD-teszt eredménye

bővítésben megvalósított lemezegység-kezelőn keresztül a CP/M 2.2 egy változata (UPM) biztosítja a kompatibilitást a korábban erre a rendszerre kifejlesztett programokkal. A gyártó más gépekre kifejlesztett táblázatkezelőjét, szövegszerkesztőjét, formanyomtatvány-kezelőjét, formátumgene-

rátorát, adatrögzítés-vezérlőjét, levélnyilvántartóját, FORTH rendszerét kívánja ehhez a rendszerhez átdolgozni. A rendszer további perifériákkal, például a már gyártott nyomtatóillesztővel és soros illesztővel is ellátható. Úgy tűnik, hogy a gyártó teljes rendszert igyekszik kiépíteni.

Mikroszámítógépek a tavaszi fesztiválon



A háztartási program-pályázat zsűrizése. Az egyik versenyző és a zsűri elnöke, Dr. Kocsis János

A Budapesti Tavaszi Fesztivál eseményei között első ízben megrendezett mikroszámítógépes seregszemle gerince a „Számítástechnika mindenkié, a számítástechnika mindenkiért” című kiállítás volt, amelyen több mint 60 nagyvállalat, kutatóintézet, szövetkezet és kisvállalkozás mutatta be mikroszámítógépeit, alkalmazói programjait és szolgáltatásait.

A μ '86 ugyancsak kiemelkedő rendezvényei voltak azok a bemutatók, amelyek a számítástechnikának az oktatásban használható lehetőségeit tágitották.

Elhozták alkotásaikat az amatőr számítógép-építők és programozók is. A legjobbak értékes díjakban részesültek, ám az amatőrök részvétele nem az önmagában is érdekes verseny miatt volt fontos, hanem mert megmutatta, hogy mozgalmuk olyan lendítő erő lehet a számítástechnika hazai elterjedésében, mint a rádióamatőröké a század elején a híradástechnikai ipar kibontakozásában.

A hatnapos találkozón az említetteknek kívül sor került többek között egy számítógépes hangversenyre, muzeális gépek bemutatására, vitákra a távoktatásról és a szaksajtóról, érdekes demonstrációs előadásokra, alkatrész- és chipcsereberére és vásárra, könyvbemutatóra és -vásárra, film- és videovetítésekre.

A μ '86 egyes programjait a számítástechnika magyar nagyjairól elnevezett helyszínek: a Neumann János-, a Kalmár László-, a Kozma László- és a Tarján Rezső-terekben rendezték. Az úttörőknek kijáró tiszteletadás fejeződött ki abban is, hogy kiállították az első hazai számítógépek néhány darabját. Két gép alkotójának özvegye: Kalmár Lászlóné és Tarján Rezsőné kedves vendége volt a kiállításnak.

— Férjem készítette az első hazai jellegű számítógépet, 1955-ben — mutat Kalmár Lászlóné egy külsőre is imponáns szerkezetgyűtesre. — Örülök a meghívásnak, nagyon megható, hogy külön gondoltak azokra is, akiknek munkásságához a hazai számítástechnika első jelentős eredményei fűződnek. Ami a kiállítást illeti, látványosnak, sokoldalúnak tartom, mindenki megtalálhatja az ő legjobban érdeklődő dolgokat.

Tarján Rezső annak a csoportnak volt a tudományos vezetője, amely a Szovjetuniótól kapott előtervezés alapján 1959-ben az első M 3-at építette. Özvegye most azokról a nehéz időről beszél, amikor az 50-es évek elején a koncepciók perек során férjét is elítélték:

— Férjemet a börtönben is az az idő izgatta, amikor majd Magyarországon lesznek számítógépek. Milyen kár, hogy nem lehet már közöttünk, és nem láthatja mindent a saját szemével!

Az icedi év legjelentősebb mikro-számítástechnikai rendezvényén jelen levő nagyvállalatok közül a Videoton kapcsolódott bele a legelszántabbban az iskolai számítógépprogramba. Velkei Zoltán tájékoztatta az újságírókat arról, hogy a fehérvári nagy-

meg. Itt is sikere volt a floppytisztítónak, amelyet egyébként korábban a japánok kizárólagos ázsiai értékesítési joggal megvettek. Jó volna, ha a rendezőnek, a NJSZT-nek volna egy állandó bemutatóterme, ahol a kis cégek rendszeresen felvonulathatnák újdonságaikat.

— Hogy megy az üzlet? — kértem Madaras Károlyt, aki az EL-CON kisközzöveket vízmentes, „veggyálló és vandálbiztos” fóliatászturáit árulta.

— Nem rosszul, bár a termék jellegénél fogva készpénzes üzletekről egyelőre nem beszélhetek. Az igénybejelentők nagy száma azonban mindenképpen biztató.

A SZAMALK butikjában többedmagával Freschli Ottóné állta az érdeklődők ostromát.

— Mi a sláger? — A reklámáron kínált kétféle floppy. Különösen a Parrot nevű hajlékony mágneslemezt keresték, amelyet először itt most, mi árulunk hazai üzletben.

— Maradt-e ideje másutt is körülnézni? — Nem sok, de valamennyire azért megismertem a többi helyszínt is. Azt hiszem, az a lényege az egésznek, hogy találkozzanak a pult különböző oldalain állók, felmérjék egymás igényeit, lehetőségeit. Végül is, ez sikerült.

A chipcsere és -vásár „felelőse”, dr. Simonyi Endre is nagy forgalomról számolt be.

— Mindkét vásári napon óriási tömeg tolongott itt. Voltak érdekes adásvételi megoldások. Valaki például egy táblára felírta, hogy mire volna szüksége, és mit tudna ő ezért cserébe felajánlani.

Illés Lászlóné, a Műszaki Könyvtárház igazgatója is meglepődött, mennyire kelen-dők a könyvek.

— Mint kereskedőknek, természetesen fontos volt a jelenlét. Igyekeztünk a legkülönbözőbb, hivatásosoknak és amatőröknek szóló szakkönyvekkel kijönni. Az üzleti siker minden várakozásunkat felülmúlta, kö-

μ '86



'86 vendége volt
Heinz Zemanek osztrák professzor,
akinek Szoldán Péter,
a Ságvári Endre gyakorlatiskola
8. osztályos tanulója tart
oktatásprogram-bemutatót



A µ '86 vendégei; V. Trapezanoglou,
a görög Computer Society létközönő elnöke
és az új elnök, N. Pavlidisz



A HCC standján mindig telt ház volt.
A legifjabbak próbálgatják
az amatőrök által épített számítógépeket

Witold volt az első, aki egyedül tervezte és építette gépét, méghozzá profi színvonalon. A perifériák között személy szerint nagyon tetszett Kerekes Tibor rajzoló pad-je, amely szintén első díjat kapott.

— Mi volt az ön számára a találkozói leginkább említésre méltó tanulsága?

— Talán az, hogy a nagy cégek mellett teljesen ismeretlen kis vállalkozások is felvultak. Ezek zömmel rugalmasak, gyorsak, s olyan gépekkel lépnek színre, amelyekre az egész világ szoftverpiac lehet. Nem hinném, hogy ennek hasznát különösen kellene ecsetelnem, annyira nyilvánvaló.

A µ '86 egyik legmeglepőbb közönségikerét az oktatási programbemutatók aratták. Mikés Gábor programozótanár szervezte annak a 44 előadásnak és százat meghaladó számú programnak menetrend szerinti lebonyolítását, amelyet ekkora nyilvánosság most kísért először.

— Milyen volt a színvonal?

— Változó, de hát ez nem baj, sőt természetes. Hiszen tanárok és diákok egyaránt „felléptek”, de különböző felkészültséggel, érdeklődéssel. Így mindenki számára akadt érdekes téma. Néhány nagyon jó dolgot láttam. Például Hanák Péter, a BME tudományos munkatársának ELANØ nevű, újszerű oktatási nyelvét, a debreceni Kertész Béla hullámszimuláló programját, a fonyódiak munkáját, amely 37 számítógép összekapcsolására képes, Jankó Domokos, a Közlekedéstudományi Intézet munkatársának KRESZ-és munkavédelmi vetélkedő-programját. Jól szerepeltek az ELTE Ságvári Endre gyakorlatiskola tanulói. Bemutattak például egy fényceruzát Spectrumhoz, egy nyolcadikos diájkuk pedig az itt jelenlevő osztrák professzornak németül magyarázta programját.

Ugyanitt zajlottak a TIT programbemutatói.

— Kettős céllal jöttünk ki — mondja Molnár Andor, az intézet munkatársa. — Egyrészt, hogy bemutassuk, népszerűsítsük

az általunk menedzselte programokat, másrészt, hogy gépi hátteret biztosítsunk azoknak, akik erre az alkalomra hozták el saját programjaikat.

Eddig 350-féle programot fogadtunk, ezekből 14 ezer kelt el összesen. És ez csak középiskolás program! Az általános iskolák belépésével a programok száma nyilvánvalóan nagyságrendekkel fog növekedni.

A szervezők helyet biztosítottak az egészségkárosultaknak is. A Vakok és Gyengénlátók Szövetségének standján mutatkozott be a Braille-lab beszélő számítógép. Segítségével a vakok számára olyan új munkakörök válnak elérhetővé, mint például a telekezelő, az adatbázis-kezelő vagy a diszpečseré. Arató András, a KFKI főmunkatársa, tervezőmérnök, részt vett az első hazai fejlesztési beszélő számítógép létrehozásában.

— A Braille-lab az első Homelab alapján készült. Lényege az alapgéphez kifejlesztett beszélő BASIC-interpreter. A dialóg formába átvitt BASIC programok révén a vak ember is mindazt tudja a számítógépen, amit egy látó. Jó lenne, ha a Homelab-osokhoz hasonlóan minél többen magukéknak éreznék az ügyet, hozzájárulnának a látáskárosultak életlehetőségeinek javításához!

Már csomagoltak a résztvevők, amikor megkértük Kovács Győzöt, a µ '86 rendezőbizottságának elnökét, hogy készítsen gyorsmérleget a találkozóról.

— Nem más vásárokkal, kiállításokkal való vetélkedés volt a célunk, hanem a számítástechnika népszerűsítése. Az a társadalom, amely nem képes alkalmazni az új technikat, könyörtelenül lemarad! Mi hátrányban vagyunk olyan országokhoz képest, ahol egy Commodore árát öt nap alatt meg lehet keresni, mert nálunk ez az idő 14 hónap. A törülköző bedobása helyett azokat az erőket kívántuk mozgósítani, amelyek részesei lehetnek a hátrány ledolgozásának! Másrészt a fogadóközönség megtretemtését és kiszélesítését is célul tűztük ki. Ezzel kapcsolatban reménytelnek tartom, hogy szokatlanul sok felnőtt jött el, és megpróbált komolyan programozni! Számunkra kissé meglepő volt az oktatóprogramok nagy sikere is, ezekből most volt először több, mint játékprogramokból. Mindezen kívül talán ezúttal azt is be tudtuk bizonyítani, hogy a számítógépnek helye van a lakásban! Örülünk, hogy a µ '86 nem bizonyult idegen testnek a Budapesti Tavasz Fesztivál egészében. Igenis, a számítástechnika hozzá tartozik az egyetemes emberi kultúrához! Valamennyi tapasztalatunkat most nem tudom felsorolni, de az bizonyos, hogy hasonló találkozót évente érdemes megtartani. Úgyhogy a legközelebbi a szervezés már most megkezdjük. Szeretnénk jövőre tágasabb helyen, jobb propagandával megrendezni, és lehetőleg minél több vidéki várost is bevonni.



A budapesti találkozóval egyidőben az NKSZT Bács-Kiskun megyei szervezete Kecskeméten „minifesztivált” rendezett, amelynek fővédőnke Tohai László, a megyei szervezet elnöke volt. Sokan nézték meg a szoftverkiállításon és -börzén ajánlott termékeket, és jól sikerült a közép- és általános iskolás diákok vetélkedője is.

LACZKA MIKLÓS

rűlből 100-120 ezer forintos bevétellel számolhatunk.

A µ '86 talán azok számára marad a legemlékezetesebb, akik a különböző pályázatokon díjakat nyertek. A háztartási programok versenyében egy fiatal miskolci házaspár, Jászay György és felesége lettek az elsők. A Commodore 64-re írt programjuk egy család napi, heti, havi és éves kiadásainak és bevételeinek vezetésére, különböző összesítők készítésére alkalmas.

Dr. Horváth László, a Műszaki Egyetem adjunktusa az általános zsűrititkár feladatát látta el.

— Mit mond a szakember a bemutatott pályamunkákról?

— A pályázatokat mindenképpen jóval előbb kell majd meghirdetni, hogy az amatőröknek a mostaninál több idejük legyen felkészülni. A szoftvereknél voltak jó, értékelhető ötletek, de a kivétel kevés esetben ütötte meg a remélt színvonalat. Sokan azt hiszik, hogy a naiv művészek mintájára „naiv programozók” lehetnek. Ha többször foglalkoznánk irodalommal, figyelembe vennének igényes módszertani megoldásokat, már meglévő modulokra jobban építenék, sokkal komolyabb eredményekről számolhatnák be most én is. Úgy vélem, a hardveresek képviselték a magasabb színvonalat. Többeknél viszont az volt a gondom, hogy a magukkal hozott Homelabokhoz fabrikáltak mindenfélét. Nem véletlen, hogy a rendszer-kategóriában Kusinszky

Programozás profi szinten

Új hagyomány méltó folytatása volt az 1986. március 21-én Szekszárdon immár harmadszor megrendezett országos program pályázat döntője. A pályázatot az NJSZT, a szekszárdi Garay János Gimnázium és szerkesztőségünk hirdette meg közép- és általános iskolások részére, játékprogram és valamely tantárgyhoz kapcsolódó oktatóprogram kategóriában.

Felhívásunkra alig egy hónap leforgása alatt 83 pályázó küldte be programját. Lapunk diákszerkesztősége Bayer József, Énekes Ferenc, Koltai Márta, Pesti Gyula és Siegler Gábor tanárok segítségével kipróbálta, analizálta a pályaműveket. Javasataik alapján az NJSZT Ifjúsági Bizottsága 10-10 programot jelölt ki a döntőre.

A döntőbe jutott programok nagy sikert arattak a Budapesti Tavasz Fesztiválon az I. Országos mikroszámítógépes találkozó közönsége előtt.

A 90 éves fennállását ünneplő szekszárdi Garay János Gimnázium ünnepi rendezvényeinek sorában tanárok és diákok között nagy érdeklődést kiváltó esemény volt a döntő. Külön teremben futtaták a HT gépre írt programokat és külön teremben a C64 és Spectrum programokat. A zsűrinek a pályázók egyenként mutatták be „produkciójukat”. A legjobbak imponáló magabiztossággal, egyenrangú partnerként, néhányan kedves humorral fogadták a tizenhárom tagú zsűri mélyeségeken kutató kérdéseit.

A tiz oktatóprogramból nyolc HT-1080-as gépre és kettő ZX-Spectrumra készült. Természetesen egy részük a matematika, számítástechnika oktatására szolgál, de némelyik meglepően alapos biológia-, földrajz- és nyelvtudást, közlekedési ismereteket tartalmaz.

A játékprogramok közül három HT-1080-as, kettő C64-es és öt ZX-Spectrum gépre készült. Voltak közöttük olyanok is, amelyek egy szusz-



A versenyprogramok bemutatóit élénk érdeklődés kísérte



A zsűri eredményt hirdeti

ra többórás szórakozást nyújtottak.

Késő délután született meg a zsűri döntése.

Az oktatóprogram kategóriában

1. díjat nyert Cziráki Gábor, a szekszárdi Garay Gimnázium diákja. A TII ajándékát, egy Primo A32-es számítógépet Kovács Győző, a zsűri elnöke adta át.

2. díjat nyert Nyilas István László, a nyíregyházi Krúdy Gyula Gimnázium tanulója. A TII ajándékát, egy MK-29 magnetofont Kovács Győző adta át.

A 3. díjat, az 1500 forintos vásárlási utalványt lapunk ajánlotta fel, és Farkas Sándornak, a zalaegerszegi Zrínyi Miklós Gimnázium diákjának Kós Géza, a diákszerkesztőség tagja, az 1985. évi Nemzetközi Matematikai Diákolimpia győztese adta át.



Munkában a játékprogram-verseny győztese, Tihor Miklós



Az oktatóprogram-verseny győztese, Cziráki Gábor árvészi a díjat

A Városi Tanács különdíját Nyilas István László és Farkas Sándor kapta.

A játékprogram kategóriában

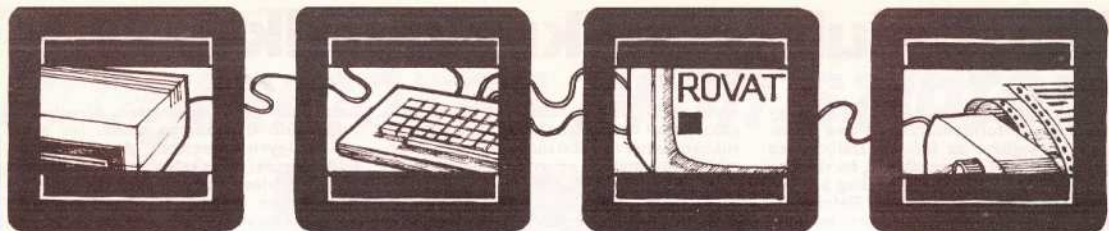
1. díjat nyert Tihor Miklós, a budapesti Steinmetz Gimnázium diákja. A KISZ Központi Bizottság ajándékát, egy Primo számítógépet Ókrós László adta át.

2. díjat nyert Rátkai István, a budapesti Fazekas Mihály Gimnázium diákja. Az NJSZT ajándékát, az MK-29 típusú magnetofont Kovács Győző adta át.

A 3. díjat, a lapunk által felajánlott 1500 forintos vásárlási utalványt Kintly Lajos, a zalaegerszegi Zrínyi Miklós Gimnázium diákja nyerte. A díjat Kós Géza adta át.

A Városi Tanács különdíját Pallagi László, az érdi Vörösmarty Gimnázium tanulója, Versits Zsolt és Parrag Zsolt, a budapesti Berzsenyi Gimnázium tanulóikapták.

A díjkiosztás végén Zentai András igazgató, a rendezvény házigazdája összegezte a vetélkedő eredményeit és bejelentette, hogy jövőre ismét megrendezik a versenyt.



ÉKEZETES ÁBÉCÉ C64-RE

Mi is több C64-re készült ékezetes ábécét látnunk már. Amiért mégis hozzáfogtunk egy újabb elkészítéséhez, azt azért indokoljuk, hogy az általunk ismertek vagy nem tartalmaznak az összes ékezetes betűt, vagy ha igen, akkor azok nem nyomtathatók, illetve ha mindezeket a feltételeket teljesítik, akkor az egyes ékezetes betűk billentyűi fixek, nem a felhasználó vagy a programozó választja meg azok elhelyezését tetszése szerint.

Olyan ékezetes ábécé megalkotását tűztük feladatunként magunk elé, amely:

1. a teljes magyar kis és nagy ábécét tartalmazza,
2. képernyőn és nyomtatón egyaránt megjeleníthető,
3. az ékezetes betűk billentyűzetkiosztásáról a programozó dönt,
4. a kész ábécé lemezen vagy magnón tárolható.

Már kezdetben élesen különválasztottuk az ábécé elkészítését és felhasználását. Az elkészítés lehet hosszabb, időigényesebb feladat, a felhasználás viszont egyszerű legyen.

Az első lépés a képernyő és a nyomtató karaktereinek megtervezése volt. A képernyő betűinek tervezésekor találkoztunk azzal a problémával, hogy egy karakter 8x8-as mátrixában a betű az első hét bájtot foglalja el, a nyolcadik csupa nulla a sorköz, az ékezetek viszont a betű felett helyezkednek el. Így adódott az az ötlet, hogy az egész karakterkészletet hozzuk le egy bájttal, ezáltal a képernyősorokat elválasztó pontsora a karakterek tejetjére kerül, helyet biztosítva az ékezetek részére. Problémát csak az úgynevezett alsó állású



karakterek, a , ; p q g j y okoztak. Ezek új alakot kaptak. A nyomtatón az ékezetes betűk mellett új alakúak a p q g azért, hogy ne emelkedjenek ki a kisbetűk magasságából. Nyomtatón a C64-hez közvetlenül csatlakoztatható nyomtatókat értjük.

Az új karakterkészlet az l-es helyet, azaz a memória 2 k-4 k közötti területét foglalja el. Közvetlen utána helyezkedik el a kazettára vitelt végző gépi kódú rutin, ezért az l. program begépelése előtt feltétlenül adjuk ki a POKE44,16:POKE45,65:POKE4096+64,0:NEW parancsot!

A karakterkészlet elkészítése az l. program begépelését és futtatását jelenti. A futtatás kezdetén a képernyő kicsit szokatlan, látszólag hibás, de a karakter ROM átmosása és az új karakterek betöltése után helyreáll a kép.

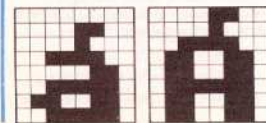
Az új karakterek billentyűjének meghatározásakor a felhasználó választhat, hogy az ASCII kódot adja meg, vagy egyszerűen lenyomja azt a billentyűt, ahova az új karaktert szeretné elhelyezni. A választáskor a SHIFT,C=, CTRL, RVS ON, RVS OFF billentyűk nyugodtan használhatók. A program sorban megkérdezi az új alakú és az ékezetes betűk helyét. A bevétel befejeztével választhatunk, hogy kazettára vagy lemezre kívánjuk-e menteni az új karakterkészletet. A képernyő karaktereit a KAR. nevű programfájlból, a nyomtató karaktereit pedig a PRINT.SEQ nevű szekvenciális fájlba menti el a program.

1. program

```

3 REM *****
4 REM * ALAP KARAKTERKESZLET KESZITES *
5 REM *****
10 PRINTCHR$(8):POKE53272,19
20 PRINT"Q*** / * ! \ L \ T ! ***"
30 F1=55296:F2=2048:F3=F2+768:F4=4096
40 V$=CHR$(34):M=1:DIMN(20)
50 FORA=0T07:READS%(A):NEXTA
51 N(0)=71:N(1)=80:N(2)=81:N=3
52 :
60 FORA=F4+20TOF4+56:READB
61 POKEA,B:NEXTA
70 :
80 POKE56334,0:POKE1,51
90 FORA=0T02047STEP512:H=ABS(H-1)
100 IFA=1024THENP=255
110 FORC=0T0511STEP8
120 IFH=1THENPOKEF2+A+C,P
130 FORL=0T07
140 POKEF2+A+C+L+H,PEEK(F1+A+C+L)
150 NEXTL:IFH=1THENPOKEF2+A+C,P
160 NEXTC:A:P=0
170 FORA=512T01536STEP1024
180 IFA=1536THENP=255
190 POKEF2+A+8,P:FORC=8T0216
200 POKEF2+A+C+1,PEEK(F1+A+C):NEXTC,A
210 POKE1,55:POKE56334,1
220 :
230 FORA=0T06:READS%(A):NEXTA
240 FORA=0T06:FORC=0T07:READP
250 POKEF2+1024+S(A)*8+C,P:255-P
260 POKEF2+S(A)*8+C,P:NEXTC,A
270 :
280 DATA128, 32, 0, 64,191, 96, 64, 96
290 :
300 DATA174, 0, 16,172, 1, 16, 32,186
310 DATA255,173, 2, 16,162, 3,160, 16
315 DATA 32,189,255,169, 0,133,254,169
320 DATA 0,133,255,169,254,162, 0,160
330 DATA 16, 32,216,255, 96
340 :
350 DATA 7, 16, 25, 17, 10, 44, 59
360 DATA 0, 0, 0, 62, 96,110,102, 62
370 DATA 0, 0, 0,124,102,124, 96, 96
380 DATA 0, 0, 0,102,102, 60, 24, 24
390 DATA 0, 0, 0, 60,102,102, 60, 14
400 DATA 0, 0, 0, 6, 6, 6,102, 60
410 DATA 0, 0, 0, 0, 0, 24, 24, 48
420 DATA 0, 0, 24, 0, 0, 24, 24, 48

```



```

430 :
440 REM *****
441 REM * UJ KAR. HELYENEK MEGADASA *
442 REM *****
450 PRINT "JAJALASSZD KI A MEGFELELO"
460 PRINT "BILLENTYUT VAGY ASC KODOT!"
470 PRINT "AJA RENDBEN VAN, AKKOR"
480 PRINT "NYONJAN MEG A <RETURN>-T!"
490 :
500 PRINT "*** BEVITEL ***"
510 PRINT "(1) ASC KOD"
520 PRINT "(2) BILLENTYU"
530 INPUTM:ONMGOTO560,560
540 PRINT "J":GOTO530
550 :
560 FORA=0T07:READB:IFB=-1THEN942
570 POKEF3+A,B:NEXTA
580 PRINT "*** AZ UJ KARAKTER:"
590 PRINTV$;CHR$(160);V$:"**"
600 ONMGOSUB670,730
610 N(N)=AS:N=N+1
620 SH=INT(AS/32):SL=AS-SH*32
630 S=F2+(S%(SH)+SL)*8
640 FORA=0T07:POKES+A,PEEK(F3+A)
650 NEXTA:PRINT "T":GOTO560
660 :
670 INPUT "KEREM A KODOT",A:PRINT "J";
680 IFSGN(A)=-10RA>255THENAS=A:GOTO670
690 RETURN
700 :
710 GETG$:IFG$=""THEN710
720 IFG$=CHR$(13)THENG$="" : RETURN
730 PRINT "A BILLENTYU: ";V$;G$;V$;"J"
740 AS=ASC(G$+CHR$(0)):GOTO710
750 :
760 DATA 0, 12, 8, 60, 6, 62,102, 62
770 DATA 12, 8, 60,102,102,126,102,102
780 DATA 0, 12, 8, 60,102,126, 96, 60
790 DATA 12, 8,126, 96, 96,120, 96,126
800 DATA 0, 54, 0, 60,102,102,102, 60
810 DATA 54, 0, 60,102,102,102,102, 60
820 DATA 0, 54, 0,102,102,102,102, 62
830 DATA 54, 0,102,102,102,102,102, 60
840 DATA 0, 54, 36, 60,102,102,102, 60
850 DATA 54, 36, 60,102,102,102,102, 60
860 DATA 0, 54, 36,102,102,102,102, 62
870 DATA 54, 36,102,102,102,102,102, 60
880 DATA 0, 12, 8, 60,102,102,102, 60
890 DATA 12, 8, 60,102,102,102,102, 60
900 DATA 0, 12, 8,102,102,102,102, 62
910 DATA 12, 8,102,102,102,102,102, 60
920 DATA 0, 12, 8, 0, 56, 24, 24, 60
930 DATA 12, 8, 60, 24, 24, 24, 60
940 DATA -1
941 :
942 FORA=0T07:POKEF3+A,0:NEXT
943 :
944 REM *****
945 REM * UJ KARAKTERKESZLET MENTESE. *
946 REM *****
950 PRINT "J *** BEVITEL VEGE ***"
970 PRINT "(1) MENTES KAZETTKARA"
980 PRINT "(2) MENTES LEMEZRE "
990 INPUTA:ONAGOTO1010,1010
1000 PRINT "J":GOTO990
1010 PRINT "JIEGYE BE A LEMEZT!"
1020 PRINT "(X)AGY ALLITSA BE A MAGNOT!"
1030 PRINT "YOMJON MEG EGY BILLENTYUT!"
1040 WAIT203,62:ONAGOTO1050,1060
1050 E=1:F$="KAR.":GOSUB1080:GOTO1130

```

```

1060 E=8:F$="@:KAR.":GOSUB1080:GOTO1140
1070 :
1080 POKEF4,E:POKEF4+1,0:H=LEN(F$)
1090 FORA=1T0H
1100 POKEF4+A+2,ASC(MID$(F$,A,1)):NEXTA
1110 POKEF4+2,H:SYSF4+20:RETURN
1120 :
1130 OPEN1,1,1,"PRINT.SEQ":GOTO1150
1140 OPEN1,8,8,"@:PRINT.SEQ,S,W"
1150 FORA=0T017:PRINT#1,N(A)
1160 PRINT#1,CHR$(8);
1170 FORC=0T05:READB
1180 PRINT#1,CHR$(B+128):NEXTC
1190 PRINT#1,CHR$(15):NEXTA:CLOSE1
1200 :
1203 DATA 56, 68, 68, 84,116, 0
1206 DATA124, 20, 20, 20, 8, 0
1209 DATA 24, 36, 36,100, 80, 0
1211 DATA 32, 84, 86, 61, 64, 0
1220 DATA120, 20, 22, 21,120, 0
1230 DATA 56, 84, 86, 85, 8, 0
1240 DATA124, 84, 86, 85, 68, 0
1250 DATA 56, 69, 68, 69, 56, 0
1260 DATA 61, 66, 66, 66, 61, 0
1270 DATA 60, 65, 64, 61, 64, 0
1280 DATA 60, 65, 64, 65, 60, 0
1290 DATA 56, 70, 69, 70, 57, 0
1300 DATA 60, 66, 71, 66, 61, 0
1310 DATA 60, 66, 65, 62, 65, 0
1320 DATA 60, 66, 65, 66, 61, 0
1330 DATA 56, 68, 70, 69, 56, 0
1340 DATA 60, 66, 70, 66, 61, 0
1350 DATA 60, 64, 66, 61, 64, 0
1360 DATA 60, 64, 66, 65, 60, 0
1370 DATA 0, 68,126, 65, 0, 0
1380 DATA 0, 68,126, 69, 0, 0
1390 END

```

Az 1. program működésének vázlatos magyarázata

- 10 Kisbetű – nagybetű karakterkészlet beállítása
30–40 Kezdő értékek beállítása
F1 a karakter ROM kezdete
F2 az új karakterkészlet kezdete
F3 a SHIFT SPACE helye, szerepét lásd később
F4 a gépi rutin kezdete
50 ASCII – POKE kódátszámítás, a hozzá tartozó DATA sor: 280
60–61 A gépi rutin betöltése, DATA sorai 300–330
80–210 A karakter ROM átirása és egy sorral lejjebb helyezése
230–260 Az alsó állású karakterek képernyőképének betöltése, DATA sorai 350–420
450–540 Menü a képernyőre, válasz fogadása
560–570 A SHIFT SPACE helyére teszi az új karaktert a megjelenítéshez
580–600 Az új karakter megjelenítése, elágazás billentyű vagy ASCII kód szerint
610–650 Helyére teszi az új karaktert
670–690 ASCII kód szerint
710–740 Billentyű alapján
760–930 Ékezetes betűk képernyőadatai, rendre: á, Á, é, É, ő, Ő, ü, Ű, ó, Ó, ú, Ű, ö, Ó, ú, Ű, í, I
940 DATA végjel
942 A SHIFT SPACE helyére visszatölti az eredeti értékeket
960–1040 Készülékválasztás a mentéshez és a válasz fogadása
1050–1060 Fájlnév és készülékszám
1080–1110 Gépi rutin paraméterezése és hívása
1130–1190 SEQ fájl készítése a nyomtatóhoz és készülékszám a mentéshez
1210–1380 A nyomtató három új alakot kapott és ékezetes karaktereinek adatai, az ékezetes betűk sorrendje ugyanaz, mint a képernyőnél

2. program

```

10 REM **** FO PROGRAM ****
20 GOSUB100
30 J=1:PR$="7LDAPROGRAM KI...RAT RSHOZ!"
40 GOSUB110
50 END
60 :
100 REM **** EKEZETES PRINT TOLTO ****
101 IFA=0 THEN A=1:LOAD"KAR.",8,1
102 PRINTCHR$(8):POKE53272,19
103 DIMPR$(255)
104 OPEN#8,B,"PRINT,SEQ,S,R"
105 INPUT#8,B:INPUT#8,PR$(B)
106 IFST=0 THEN I05
107 CLOSE8:RETURN
108 :
110 REM **** EKEZETES PRINT ****
111 OPEN#4,4,7,CHR$(15):FORB=1 TO LEN(PR$)
112 P$=MID$(PR$,B,1):P=ASC(P$+CHR$(0))
113 IFPR$(P)=" " THEN I115
114 P$=PR$(P)
115 PRINT#4,P$:NEXTB:IFJ=1 THEN PRINT#4
116 CLOSE4:RETURN

```

Betű	SHIFT	C=	CTRL+
á		A	
Á			A
é		E	
É			E
í		I	
Í			I
ó		O	
Ó			O
ő	ő	ő	
Ő	—	—	
ú		U	
Ú			U
ü		*	
Ü			*
ű		£	
Ű			£

Javaslat az ékezetes karakterek elhelyezésére

Az új ábécé használatát mutatja be a 2. program. Beírása, illetve használata előtt itt is el kell tolni a BASIC terület kezdetét. Ez vagy egy betöltőprogrammal, vagy a már ismertett parancssor 4 k-ra történő módosításával érhető el. Az ékezetes betűk eléréséhez a felhasználói program elején hívjuk meg a 100-as rutint. Ekkor a képernyőn már használhatók az új karakterek. Nyomatathoz a szöveget a PR\$ változóba helyeztük el, a nyomtatás a 110-es rutin hívásával aktivizálható. J=1-re a következő nyomtatás új sorban, J=0-ra az aktuálisban folytatódik.

A bemutatott elhelyezéssel az ékezetes karakterek elérése nem a legkényelmesebb ugyan, de egyszerű. Az új CTRL+betű elérésű karakterek idezőjelben belül simán, azon kívül használva az inverz üzemmódban be-ki kapcsolásával érhetőek el.

FÖLDI ENDRE—ÉNEKES FERENC

PTA-4000 PROGRAMOK

Gépi kódú programok

1. Az első program bármely BASIC-programnál gyorsabban invertálja a képernyőn levő szöveget vagy ábrát.

Működése

- 2 A képernyő-memória utolsó bajtját beteszi a U regiszterbe. (A képernyő-memória &7600—&764D és &7790—&774D)
- 3 Csökkenti U-t
- 4-6 U-t beteszi az A akkumulátorba A EOR&FF (kettes számszerben 1→0; 0→1), majd A-t visszatesszi U-ba
- 7-9 Egymásba ágyazott ciklusok &77-ről &76-ra és &4D-től &00-ig
- 10 Visszatér a BASIC-be
2. A második program egy meghatározott alakzatot vagy betűket mozgat végig balról jobbra a képernyő bármely pontjától bármely más pontjáig, változtatható sebességgel.

Működése

- 1 XL-be beteszi a képernyő azon oszlopának a sorszámát (0—155), ahonnan az alakzatot mozgatni kívánom
- 2-5 Növeli XL-t, majd kiírja az &7875 címre
- 6-7 Y-ba betölti a kirajzolandó alakzat kezdőcímét (&4140)
- 8 UL-be betölti az alakzat hosszát
- 9-12 Ciklus, mely a teljes alakzatot kiírja egyszer a képernyőre. (Az &EDEF-en levő rutin az A-ban levő szám grafikus jelét kirajzolja azon oszlopokra, melynek sorszáma az &7875 címen van)
- 13-17 Két egymásba ágyazott lassító ciklus
- 19-20 Eggyel lépteni az alakzatot, míg el nem éri a megadott oszlopszámot

A program állítható értékei

Az első oszlop: &4101 (0—155)
 Az utolsó oszlop: &4126 (nagyobb legyen, mint az első)
 Az alakzat kezdőcíme: alsó bajt: &411A; felső bajt: &411C
 Az alakzat hossza: &410E
 A lassítás mértéke: alsó bajt: &411C; felső bajt: &411A
 A program könnyen átalakítható úgy, hogy jobbról balra mozgassa az alakzatot. Az &4113 címre 2-t vagy 3-at töltve, érdekes változást tapasztalunk. Miért?

BASIC

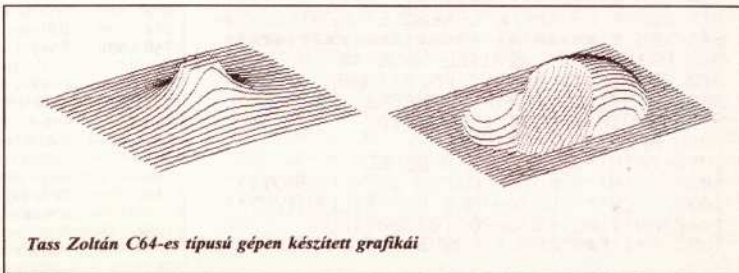
NEW &4200 (a BASIC-terület eltolása)
 10 CALL&40C5
 20 GOTO 20
 50 CLS:POKE&4140,&00,&3F,&7F,&79,&3F,&3F,&38,&78,&7C,&78,&38,&78
 60 CALL&4100
 70 GOTO 70

GÉPI KÓD

1 40C5 LDI UH 78H	68 78
2 40C7 LDI UL 4DH	6A 4D
3 40C9 DEC UH	FD 62
4 40CB LDA U	25
5 40CC EAI FFH	BD FF

6 40CE STA U	2E
7 40CF LOP 06H	88 06
8 40D1 CPI UH 77H	6C 77
9 40D3 BCS -0EH	93 0E
10 40D5 RTN	9A
1 4100 LDI XL,00H	4A 00
2 4102 INC XL	40
3 4103 PSH X	FD 88
4 4105 LDA XL	04
5 4106 STA 78H,75H	AE 78 75
6 4109 LDI YL,40H	5A 40
7 410B LDI YH,41H	58 41
8 410D LDI UL,0AH	6A 0B
9 410F LIN Y	55
10 4110 ADI 78H,75H, O1H	EF 78 75 01
11 4114 SJP EDH,EFH	BE ED EF
12 4117 LOP 0AH	88 0A
13 4119 LDI UH,40H	68 40
14 411B LDI UL,FFH	6A FF
15 411D LOP 02H	88 02
16 411F DEC UH	FD 62
17 4121 BZR -08H	99 08
18 4123 POP X	FD 0A
19 4125 CPI XL,9AH	4E 9A
20 4127 BZR -27H	99 27
21 4129 RTN	9A

BALÁZS GYÖRGY



Tass Zoltán C64-es típusú gépen készített grafikái

A SZÁMÍTÓGÉP ANATÓMIÁJA

Előző számunkban a rendszer működéséhez feltétlen szükséges vezérlőjelekkel foglalkoztunk. Most a rendszer működésének néhány újabb mozzanatával ismerkedhetünk meg.

Vegyük például a bekapcsolás mozzanatát! Mint a programozásból tudhatjuk, a központi egység (CPU) sok tárolóelemet tartalmaz, és ezek állapota bekapcsolás után véletlenül változhat. Ilyenkor a rendszer működésképtelen. Az üzembe helyezéshez egy impulzust kell adni a processzor és néhány „intelligens” áramkör RESET bemenetére. Ez egy megadott alaphelyzetbe hozza a központi egységet, például a programot elindítja egy megadott címről. Ez a cím lesz a PC vagy IP regiszter tartalma. A bekapcsolás utáni „resetelés” feladatát látja el például az 1. ábra szerinti, ún. auto-reset áramkör.

Amikor egy bonyolult és lassú, hosszú hozzáférési idejű periféria-áramkör, például egy lemezvezérlő és a processzor között kell adatot átvinni, a következő játszódik le:

1. A központi egység jelzi, hogy szüksége van az adatra.
2. A periféria-áramkör megkeresi azt.
3. A periféria jelzi a központi egységnek, hogy megvan az adat.
4. A központi egység átveszi az adatot.

Amíg a periféria a keresést végzi (2. pont), a központi egység „rээр”, futhat a program. Ahhoz azonban, hogy az adatot átvegye, meg kell szakítani a program futását. Ezt a külső egységnek kell kezdeményeznie. Erre szolgál a vezérlőbusz INT (interrupt = megszakítás) ve-

zetéke. Ha ezen a vezetéken egy aktív él jelenik meg, a program futása felfeszakad, és a vezérlés egy megadott címre kerül.

Az ugrás címe és az, hogy a vezérlésátadáson kívül még mi mindent történik, az adott rendszertől függ.

A 6800-as központi egységnél, mint sok más processzortípusnál, kétféle megszakítás van: az egyik szoftver úton letiltható, a másik nem. Az utóbbi neve NMI = Non-Maskable Interrupt. Ezek bármilyenek érvényre jutásakor (a letiltható IT nem jut mindig érvényre) a központi egység verembe menti a regisztereit, majd elugrik egy megadott címre. Ez a cím IT esetén a memória FFF8-FFF9 rekeszében, NMI esetében FFFC-FFFD rekeszében található.

A 8080-nak egy IT bemenete van, ez letiltható. Ha a központi egység elfogad egy megszakításkérélmel, INTA nevű kimenete aktív szintű lesz (INTA = INTRupt Acknowledge = megszakításelfogadás). Ekkor a külső eszköznek egy 8080 gépi nyelvű utasítást kell elhelyeznie az adatbuszon, ezt a processzor végrehajtja. Ez az utasítás általában egy RST. Ez egy egybájtos CALL, ami megadott címre ugrik. Összesen 8 RST utasítás van, ezek 0-7-ig számozódnak. Az ugrási cím a sorszám*8 lesz, tehát például az RST 2 utasítás a 10H címre ugrik.

A Z80-nál van NMI és letiltható IT is. Az NMI egy CALL 66H utasítást hajt végre, és elmenti az IFF (megszakításengedélyezés) állapotbitet.

Az INT hatása háromféle lehet: szoftver úton választható ki, az IM utasítás segítségével.

A 0-as mód (IM 0) megegyezik a 8080 megszakításkezelésével. Az INTA jelet az IORQ+MI (lásd egy későbbi részben) együttes jelenléte helyettesíti.

Az 1-es mód (IM 1) esetén egy CALL 38H utasítást hajt végre a központi egység a megszakítás elfogadásakor.

A 2-es mód (IM 2) az úgynevezett vektoros megszakítás. Ekkor szintén megjelenik az IORQ+MI jel, aminek hatására a periféria-áramkörnek egy bájtot kell elhelyeznie az adatbuszon. Ez a bájtot a processzor I regiszterével együtt egy címet alkot. Ezen a memóriacímen van a tényleges ugrási cím.

Gyakorlati példaként bemutatjuk egy óra kapcsolását az iskolaszámítógéphez és a hozzá szükséges szoftvert. VIGYÁZAT! A készüléket csak kikapcsolt állapotban, kikapcsolt gépre szabad csatlakoztatni! Bekapcsolni csak a szoftvert betöltése után szabad!

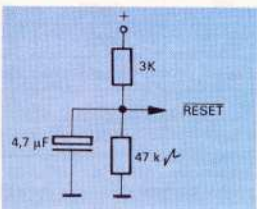
A kapcsolást az 1. ábrán, a programot a 2. ábrán találhatójuk.

A program systemmel indítható, kezdőcíme 32000. Ide kell beállítani a RAMTOP-ot, a READY? kérdésre adott válasszal. Ekkor a bal felső sarokban megjelenik az óra, a perc és a másodperc. Ezt kikapcsolni LINE utasítással lehet. Az óra tetszőleges, akár BASIC, akár gépi kódú program futása közben látható. A kapcsolás tartalmaz egy számláló frekvenciaosztót és egy ún. buszmeghajtót. Ez csak annyit csinál, hogy a bemenetien levő értéket megjelenti a kimenetén, ha az engedélyező bemenete (V) aktív, egyébként úgy viselkedik, mintha „ott se lenne”. Ennek segítségével helyezkedik el az adatbuszon a kívánt ugrási cím címének első bájtiát, esetünkben 2-t.

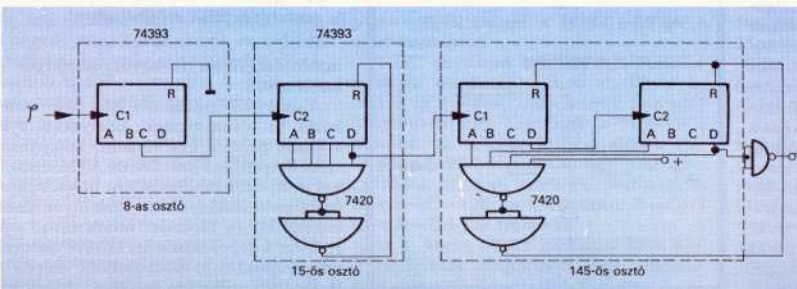
FEHÉR TAMÁS-
MOLNÁR ATTILA
(Folytatjuk)

```

1 LINE EQU 41A3H
2 ORG 32000
3 LDAD 32000
4 IN Z
5 LD HL, LINE
6 LD (HL), 0F3H
7 INC HL
8 LD (HL), 0D9H
9 LD A, 7FH
10 LD I, A
11 EI
12 JP 1A19H
13 ORG 7F00H
14 LOAD 7F00H
15 DW 7F02H
16 LD HL, ORA
17 LD A, (HL)
18 INC A
19 CP 100
20 JR NZ, BEF
21 LD (HL), 0
22 LD B, 2
23 C1 INC HL
24 LD A, (HL)
25 INC A
26 CP 30H
27 JR NZ, BEF
28 LD (HL), 30H
29 INC HL
30 LD A, (HL)
31 INC A
32 CP 36H
33 JR NZ, BEF
34 LD (HL), 30H
35 DJNZ C1
36 INC HL
37 LD A, (HL)
38 CP 36H
39 JR NZ, BEF
40 LD (HL), 30H
41 INC HL
42 LD A, (HL)
43 CP 33H
44 JR NZ, BEF
45 LD A, 30H
46 BEF LD (HL), A
47 LD HL, ORA+7
48 LD DE, 3C00H
49 LD B, 3
50 C3 PUSH BC
51 LD B, 2
52 C2 LD A, (HL)
53 LD (DE), A
54 DEC HL
55 INC DE
56 DJNZ C2
57 POP BC
58 INC HL
59 DJNZ C3
60 LD A, " "
61 LD (3C02H), A
62 LD (3C04H), A
63 EI
64 RETI
65 ORA DB 0,0,0,0,0,0
66 END
    
```



1. ábra



2. ábra

Méréstechnikai alkalmazások

A PTA-4000 professzionális személyi számítógép két tulajdonsága megkülönböztetett figyelmet érdemel. Az egyik, hogy saját belső áramforrása van, s az nemcsak memóriavédelmet, de tartós, hálózattól független működést is lehetővé tesz. A másik, hogy a géphez kapcsolt készülékek és programok igen széles körben alkalmazhatók. A PTA-4000 számítógépként igen jól felhasználható, különösen akkor, ha az elvégzendő munka inkább igényli a bonyolult számítási feladatokat, mint a klaviatúra használatát.

A Mbájtok felé kacsingató személyi számítógépek között a rendelkezésre álló mintegy 2-20 kbájtnyi RAM-terület szereznekin tűnhet, de ehhez hozzá kell tenni, hogy ez a RAM statikus CMOS, így csak nagyon ritkán van szükségünk a rendszer újragenerálására, ami nélkülözhetetlen akkor, ha a számítógépet mérő vagy vezérlő rendszerben is alkalmazni kívánjuk.

A továbbiakban közreadjuk néhány, általunk már elkészített berendezés leírását azzal a céllal, hogy megkönnyítsük azok munkáját, akik hozzánk hasonlóan már gondoltak arra, hogy egy 10 000 forintos árkategóriába eső számítógéppel nem csupán a műszaki, hanem a gazdaságossági feltételeket is kielégítve lehet mérőrendszereket konstruálni.

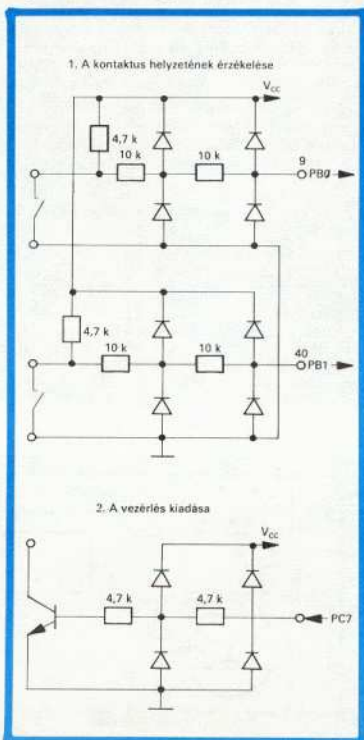
A PTA-4000 felhasználását megkönnyíti, hogy egy 60 pólusú tűferntkezés csatlakozón keresztül rendelkezésünkre áll a processzorhoz kapcsolódó belső buszrendszer, és a készülékkel járó magyar nyelvű leírás tartalmazza a csatlakozókiosztást is. A címezhető tárterületen még elég sok szabad hely van, amit jól felhasználhatunk.

A PTA-4000 sorozat első része szerint a memória két 64 kbájtnyi lapból áll. A processzorból mindkét terület elérhető, de programot csak a ME0 területen futtathatunk. A ME1 adattárolásra és periféria IC-k elhelyezésére használható. A tárterület szétválasztására szolgál még két további (PV- és PU-) vezetékek, ami így a címvezetékek számát a 8 bites processzoroknál szokásos 16-ról 19-re emeli, figyelembe véve a ME0 és ME1-et. Ez virtuálisan 1/2 Mbájt használható tárterületet jelenthetne, de a PU- és PV-vonalakat csak a szabadon hagyott helyekre általunk telepített hardverekben használhatjuk.

A viszonylag kis méretű, 195 x 85 x 25 mm-es gép 26 karakteres mátrix rendszerű LCD kijelzője igen jó kontrasztot ad, és ké-

zi mérőműszerként is használható nagyságúak az alfanumerikus karakterek. A géphez közvetlenül kapcsolható printer-plotter ugyan lepreolló nyomtatására nem alkalmas, de igazán impozáns, amikor 36 karaktert rajzol (igen, rajzol és nem nyomtat!) egymás mellé, az 57 mm-es papírcsíkbl 44 mm-t kihasználva. A printer-plotter könnyen, egyszerűen programozható, és jól kihasználható az is, hogy négy különböző színű toll áll rendelkezésre, amiből a gép programozhatóan kiválasztja a kívánt színt. Az már csupán ráadás, hogy ez a hardver olcsóbb, mint egy jó minőségű analóg regisztráló kompenzográf. Így aztán nem csoda, hogy azonnal előtérbe kerülnek a

1. ábra. Védett bemenet és kimenetek kialakítása a PTA-4000 60 pólusú csatlakozójához



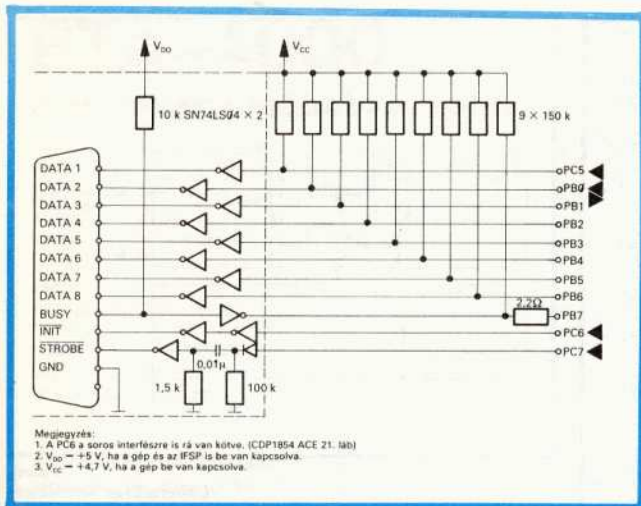
1	AD7	AD8	31
2	AD6	AD9	32
3	AD5	AD10	33
4	AD4	AD11	34
5	AD3	AD12	35
6	AD2	AD13	36
7	AD1	AD14	37
8	AD0	AD15	38
9	*NC	V _{GG}	39
10	PC7	**NC	40
11	V _{CC}	V _{CC}	41
12	V _{CC}	V _{CC}	42
13	NC	FGND	43
14	NC	FGND	44
15	PV	V _{BAT}	45
16	PU	V _{BAT}	46
17	D7	V _{BAT}	47
18	D6	V _{BAT}	48
19	D5	NC	49
20	D4	BFO	50
21	D3	ΦOS	51
22	D2	GND	52
23	D1	GND	53
24	D0	GND	54
25	INHIBIT	GND	55
26	WEX	DME0	56
27	CMT IN	R/W	57
28	W1	DME1	58
29	CMT OUT	ME1	59
30	INT	OD	60

* PB0 9. ponton
** PB1 40. ponton hozható ki

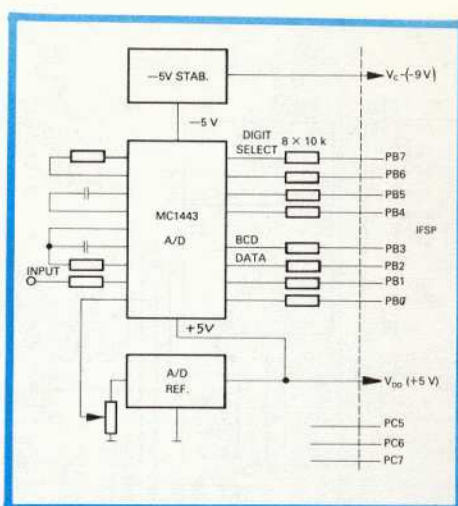
A KA 160 60 pólusú csatlakozójának bekötése (a csatlakozás felüli oldalról nézve)

mérésadatgyűjtés területén lehetséges alkalmazások.

A gyártórt sajnos ma még nem szerethetők be olyan elemek, amelyekkel a készüléket például egy vonaliró kompenzográfot helyettesítő működésre alkalmassá tehetnénk, de a laboratóriumi műszerezéssel foglalkozó szaklapokban több olyan kisebb ismertetést és hirdetést találhatunk, amelyek ezt a gépet alkalmazzák többszörös adatgyűjtőként és bonyolultabb mérési eljárásokhoz szükséges vezérlési feladatok el-



2. ábra. A CE 158-ból eltávolítható vagy más célra felhasználható párhuzamos illesztőkészülék



3. ábra. A/D átalakító beépítése a CE 158 Centronic paneljának helyére

látására. Némi számítógép-építés gyakorlatlan nem jelent nehézséget, hogy a jelentkező feladatok megoldásánál a PTA-4000-ret mi is alkalmazzuk, és ezzel gazdaságosan is reprodukálható berendezést hozunk létre. Ezt a feladatot megkönnyíti az a tény, hogy a tartozékként kapható RS232 csatlóba könnyen építhetünk kábeljeinket kielégítő készülékeket.

Ahhoz, hogy a gépben rejli lehetőségeket kihasználjuk, szükség van némi mechanikus és elektronikus konstrukciós munkára is, amit megfelelő gépi nyelven írt szoftverrel kell tetézni. A legegyszerűbb eset az, amikor a kábeljeit által a 6. mellékletben felkínált csatlakozásokat kívánjuk alkalmazni. A 9., 10., 27., 40. pontok lennének felhasználhatók, de némi eltérés van az egyes szeriák és a dokumentáció között. A KA 160 csatlakozójának kiosztását a táblázatban adjuk meg. A kiosztás megfelel az általunk vizsgált példányoknak. A 9—PB0 és a 40—PB1 NC pont egyszerűen kihozható. Lényeges eltérés, hogy a ház földpontja két egymás melletti ponton ki van vezetve. A KA 160 akkumulátorának pozitív pólusa a VBAT pontra soros diódn keresztül van kötve, ami maximálisan 50 mA áram levételét engedi meg. Zárlat esetén ez a dióda tönkremegy. A javítással ne kísérletezzünk, ezt inkább bizzuk a szervizre. A többi csatlakozópont az ábrának megfelelően van kivezetve.

A kiválasztott 4 csatlakozási pont az 5011 jelű ME1 terület hexa F000 (*F000) bázisimen elhelyezett párhuzamos illesztő PB0, PB1, PB2 és PC7 pontja (a jelölések a PTA-4000 sorozat jelölésű). A PB2 kivezetett port azonos a CMT IN kazettás adatbemenetponton. Ha ezeket a pontokat alkalmazni kívánjuk, akkor cél-

szzerű az 1. ábrán megadott védőellenlítás + dióda + tranzisztor kombináció, hogy elkerüljük a portok feszültséggel vagy árammal történő túlterhelését. A felhasználáshoz csupán az szükséges, hogy ismerjük a kiválasztott portok vezérléséhez tartozó kódokat.

A PTA-4000 1/0 portja a (hexa) *F000 alapsímen van, a ME1 területen, amire a PEEK #, illetve a POKE # utasításokkal férünk hozzá. Ezek szintaktikailag megfelelnek a PEEK és POKE utasításnak, de azt a ME1-es memóriaterülettel kezelik. Bekapcsolás után a PTA-4000 a *F000-ra elhelyezett 5011 perifériacsatló PB portjait bemenetnek definiálja. Ezt ellenőrizhetjük azzal, hogy megvizsgáljuk a PB portokhoz tartozó adatirány-regisztert, ami PEEK # & F00D utasítással történik. Az utasítás eredményeként 0-t kell kapunk. A PB portok bemenetnek és kimenetnek konfigurálhatók a *F008-be írt bájtal. Nekünk csak arra van jogunk, hogy 0-7-ig írjunk erre a memóriahelyre, azaz az általunk használt, pontosabban a gép által szabadon hagyott portokat konfiguráljuk. A kimenetnek definiált PB portok állapotát megváltoztatni a POKE # & F00F, A utasítással lehet.

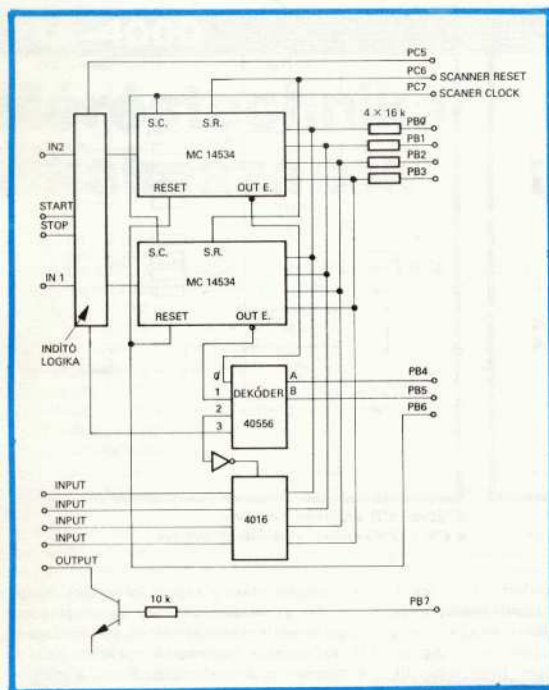
A PC7 port csak kimenet és a POKE # & F008, B utasítással érhető el. Mivel mi csak a PC7 portot vezérelhetjük és eredeti állapotában kell hagynunk a PC0—PC6 portokat, az utasítás kiadása előtt a C = PEEK # & F008 utasítással meg kell arról győződni, hogy mi van a PC portokon, és csak B = C - 128, ha B < 0, akkor B = C, illetve B = C + 128 vihető ki.

Az 1. ábrán közölt kapcsolás több egyszerű feladat megoldását teszi lehetővé, ahol elegendő 3 kontaktus figyelése és egy

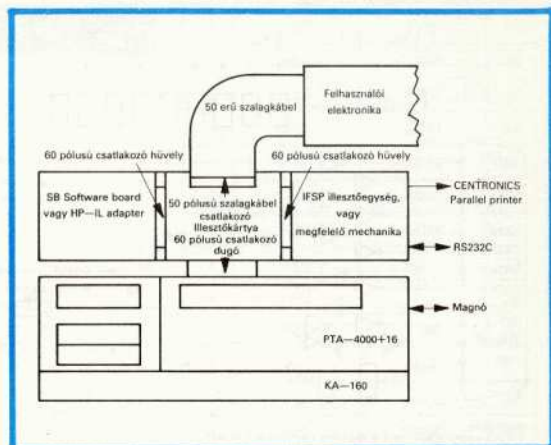
relé meghúzatása a kapott információ alapján. Ne feledkezzünk meg itt arról sem, hogy rendelkezésünkre áll az idő és dátum is, valamint a bemenet rovársára akár 4 kimenetet is képezhetünk. Ezzel a megoldással a forgószárnyas vízsebességmérőnél alkalmazott impulzusszámlálókat helyettesíthetjük. Az elrendezés alkalmazható energiatakarékos fűtésszabályozásra úgy, hogy különböző hőfokokon kapcsoló érzékelőket kötünk a bemenetekre, amelyek állapotának figyelembevételével a készülék kimenetére kötött jelfogó a napszaknak megfelelően kapcsolja a kazánt. Akár a teljes évre is kialakíthatunk programot a TIME funkció felhasználásával.

Ha az előbbieken változt feladatnál többre van szükség, amit csak kiegészítő hardver alkalmazásával valósíthatunk meg, akkor az építés feladatait jelentősen megkönnyíti, ha rendelkezésre áll egy SHARP CE158, illetve HT IFSP típusjelű Centronics és RS232C szabványú kimeneteket megvalósító illesztőelem. Ez a készülék megfelelő szakismerettel átalakítható a kívánt feladat elvégzésére. Egyszerűbb feladatok ellátására ezt az egységet minden átalakítás nélkül is alkalmazhatjuk. A két 25 pólusú csatlakozón rendelkezésre álló ki- és bemenő pontokat elővigyázattal felhasználhatjuk az eredetileg tervezettől eltérő feladatokra is.

Ez a készülék megbontása nélkül is megvalósítható. A párhuzamos kimenet (első csatlakozó) rendelkezésre áll a 9*SN74LS04 inverter, amit HIGH-ba és LOW-ba kapcsolhatunk. Ezen túlmenően még egy bemenet és még egy további kimenet is van (2. ábra), de ezekkel csak nulla felé tudunk impulzust adni és jelezni, hogy a készülék bekapcsolt állapotban van, ami-



4. ábra.
5 dekádos számláló
illesztése



5. ábra.
Univerzálisan használható
fejlesztőrendszer a PTA-4000+ KA 160-hoz

kor ezen a ponton HIGH szintet kapunk. Az alsó RS232 csatlakozón további 3 bemenet és két kimenet áll rendelkezésre SN 75198 és SN 75188 áramkörökön +9,5, illetve -9,5 V tápfeszültséggel. Kis fantáziával már ezzel is lehet egy kisvasutat vezérelni vagy az előbbieken vázolt fűtésszabályozást több paramétert figyelve megoldani.

Ha ez is kevés, akkor szétbontva a készüléket, néhány tokból álló áramkört tehetünk az IFSP Centronics meghajtóinak helyére, felépítve benne azt a funkciót, ami a számunkra elképzelt feladat megvalósításához szükséges. Ebben az esetben rendelkezésünkre áll egy, a számítógép bekapcsolásakor aktivizálódó +5 V, +9 V -9 V feszültségforrás, ami néhány milliamperrel terhelhető, egy kb. 50 x 80 mm nagyságú nyomtatott áramkör elhelyezésére alkalmas hely, továbbá egy 25 pólusú csatlakozó. Ha az RS232C változatlanul hagyása mellett döntünk, akkor 9 bemenetnek és kimenetnek, valamint 3, csak kimenetnek alkalmazható port áll rendelkezésünkre. További portokat is felszabadíthatunk, de ekkor már igen körültekintően, kezdőknek egyáltalán nem ajánlható módon kell elvégezni a szükséges programozást. Ha az RS232 kimenetet feladjuk, ez további, összesen 23 szabad be-kimeneti portot jelent, mivel tanácsos meghagyni a beépített nikkel-kadmium akkumulátoregység feszültségét jelző áramkört.

Az 50 x 80 mm nagyságú NYÁK-on 3 ki-

menet és 8 be-kimenet felhasználásával sok mindent felépíthetünk. Példánkban szerepel egy integráló A/D illesztése és egy impulzusokat adó áramlásmérőket közböző tartállyal hitelesítő készülék (3. ábra és 4. ábra). A berendezések megadjuk a blokkvázlatát is, így azok akár el is készíthetők.

Ha a túlságosan kicsi NYÁK-unkra nem tudjuk elhelyezni a szükséges áramköröket és az IFSP egységet is használni kívánjuk, ami nem rendelkezik a KA 160 printeren levő belső busz-továbbcsatlakoztatással, akkor jól felhasználható az 5. ábrán megadott csatlakozóelrendezés, amely a buszt két csatlakozóval kapcsolja szét, és lehetővé teszi két IFSP-nek megfelelő mechanika (vigyázat, nem két IFSP!) géphez történő kapcsolását.

Ebből az elrendezésből olyan kitet tervezünk, amely a készülékeket megkönnyítő mechanikát és a csatlakozási feladathoz szükséges elektronikát tartalmazza. A mechanika alkalmas 3 db 200 x 70 mm NYÁK befogadására, és egy további IFSP vagy SB szoftver board csatlakoztatására. Egy előre szerelt nyomtatott áramkört lemez címekódokkal és perifériaillesztéssel ellátva 100 x 70 mm, 2,54 raszterpontokra elhelyezett furatolt lemezt tartalmaz.

A konstrukció végeredményben a PTA-4000-hoz kialakított univerzális csatlakozó tekinthető, amit az elektronikához értő felhasználók előnyösen tudnak alkalmazni.

Nagyobb rendszerek kialakítására a közvetlen csatolás nem javasolt. Ott rendszerint nem elegendő egyetlen mikroprocesszor, és szükség van arra, hogy a gépet egy IEC625 ajánlással egyenértékű csatlakozóval lássuk el. Sajnos azonban ennek a csatlakozási módnak a fogyasztása nem a mW, hanem a W-os nagyságrendben van, ami miatt elveszítjük a kis fogyasztásból adódó előnyöket. A PTA-4000-hez illesztve rendelkezésünkre áll a HP-IL csatlakozórendszer, amely a HP-IB-vel (IEC625) azonos feladatok megoldására alkalmas, 250 kbit/s fizikai átviteli sebességgel. Ez még mindig igen jelentős, ha az RS232-nél alkalmazott sebességekkel hasonlítjuk össze. További előnye az IL-ben (Interface Loop, csatlakozóhurok) történő periféria- és készülék elhelyezésnek, vagy 2 fizikai éren történik a jelzésátvitel az első készüléktől a másodikig, majd innen a harmadikig, és így tovább. Mindegyik készülék veszi a jelet és kiértékelés után továbbítja az öt követő készülékhez. A hurokban így elvileg nem lépnek fel a buszrendszerű összekapcsolásnál fennálló terhelési problémák. Az alkalmazott készülékek közötti távolság 100 m nagyságrendű is lehet, és az egyes készülékek egymástól galvanikusan is szigeteltek. Ezzel a csatlakozóval a PTA-4000 géppel HP-perifériákat vezérelhetünk, vagy egy HP-géppel vezérelt rendszerben alkalmazhatjuk a PTA-4000-ből kialakított műszerünket, ún. folyamatközi perifériaként.

A lehetséges alkalmazások körét tovább sorolhatnánk, de ennyi elegendő annak bemutatására, hogy hogyan használható a PTA-4000 mérésmechanika célra. Megfelelő számú deklarációk esetén gondoskodunk arról, hogy az építéshez szükséges mechanikai szerelvények elfogadható áron beszerezhetők legyenek.

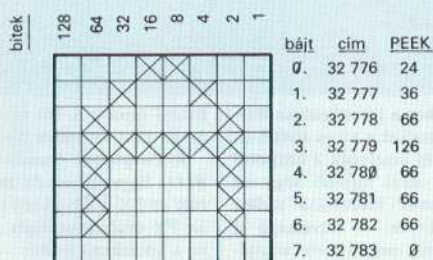
PÁLOS LÁSZLÓ—ARDAJ ZSOLT

VC20-használóknak

Egy-két hasznos ismeretet, tanácsot szeretnék átadni a VC20 használóinak. Nem állítom, hogy az ajánlott módszerek tökéletesek, de eddig mindig jól beváltak. Két témáról szólok és remélem, hogy mindenki tud valamit használni belőlük a maga „szakterületén”.

Karakterdefiniálás

A VC20 a ROM-ban 32768 (\$ 8000)-tól tárolja 8 bájtónként az egyes karaktereket. A karakterek a képernyőkódok szerinti sorrendben következnek (0-0, A-1, B-2...). Egy-egy bájt a magasra állított bitek összege. Például az A betű képét a tárban az 1. ábra mutatja. Ha tehát az N karakter bájtjait meg akarjuk keresni a tárban, a kezdőcím, ahonnan kiolvassuk a 8 bájtot, $32768 + N \cdot 8$ lesz.



1. ábra. Az A karakter képe a tárban



2. ábra. Egy négyzet definiálása a RAM-ban az A karakter helyére

A 32768-as cím megfelelője a RAM-ban 7168 (\$ ICOO), inentől kezdve írhatjuk be az általunk definiált karakterek bájtjait. Itt is ugyanolyan sorrendben helyezkednek el a karakterek, mint a ROM-ban. Tehát ha az N karaktert akarjuk átdefinálni, akkor a kezdőcím itt is $7168 + N \cdot 8$ lesz.

A 2. ábrán azt látjuk, hogy hogyan írhatunk például egy négyzetet az A karakter helyére.

Ha így elvégeztük a karakterek átdefinálását, jeleznünk kell a gépnek, hogy ne a ROM-ból, hanem a RAM-ból olvassa ki a karaktereket. Ezt az üzenetet a 36869-es cím tartalmazza. POKE 36869,255 utasítással jelezzük a gépnek, hogy álljon át a RAM-beli karakterek használatára. Ha mindent jól végeztünk, az „A” billentyű lenyomásakor az általunk beírt négyzet fog megjelenni. Ugyanígy ha az „A” betű képernyőkódjára vagy CHR\$ kódjára hivatkozunk (vagy akár egy PRINT„A” utasításban), ez az általunk definiált négyzet fog vonatkozni.

A POKE 36869,240 utasítással állíthatjuk vissza a normál karakterkészletet.

Sajnos az átdefiniált és a normál karakterkészletet együtt nem használhatjuk, tehát ha akármilyen karaktert meg akarunk jelentetni átdefiniált üzemmódban, azt a karaktert is meg kell határozni.

Csak 64 (0-63) karakter definiálására adódik lehetőség ezen a módon, mivel a $7168 + 64 \cdot 8 = 7680$, ami köztudottan a képernyőtár kezdőcíme.

Vigyázzunk, ha játékprogramban használnánk az átdefiniált karakterkészletet, ne használjuk a GET utasítást mozgatóskor, mivel ez (a C64-nél is) átírja az átdefiniált karakterek bájtjait a RAM-ban. Ezért alkalmazzuk inkább a 197-es cím tartalmát, ami az utoljára lenyomott billentyű kódja.

Sorok eltüntetése

Sok fiatal programozó gondolja, hogy programját idegen szemek elől megvédi. Erre a VC20-on tudok egy lehetőséget ajánlani. Sajnos van azonban a megoldásnak egy hátránya: két képernyősornál (44 karakter) hosszabb sorokat nem lehet eltüntetni.

Az eljárás a következő. Gépeljük be a sort, majd írjunk az utolsó utasítás után egy kettőspontot és egy REM utasítást. A REM után egy idézőjel mögé gépeljünk annyi inverz „T” betűt, ahány karaktert visszafelé le akarunk törölni. Így azok a listázásnál csak felvillannak, és rögtön el is tűnnek.

Vigyázzunk, hogy ha egy sort nem tüntetünk el teljesen, azaz a sorszám látható, nehogy a RETURN billentyűvel bevigyük a programba, mivel az interpreter nem tudja megkülönböztetni az eltüntetett sort az üres sortól.

BAGÓ ATTILA

**Felajánlunk
megvételre
egy kifogástalan állapotú**

**Siemens
404/2 típusú**

**16 munkahelyes adatrögzítő rendszert,
mely az alábbi egységekből áll:**

- 1 db 64 kbájtos központi egység,
- 2 db mágnesszalagegység
- + vezérlés,
- 2 db mágneslemezegység
- + vezérlés és konzol.

**A berendezéshez komplett
rendszer szoftvert is adunk.**

DATORG

KÜLKERESKEDelmi ADATFELDOLGOZÓ
ÉS SZERVEZŐ RT.

**Érdeklődni lehet
Perity László munkatársunknál.**

Bp. V., Dorottya u. 6. Telefon: 184-055/1429 m.

Ki ad magyarázatot?

A D100-as nyomtatónak nagyon sok jó tulajdonsága van. A Magazin 1986. februári számában feltett kérdések egyikére szeretnék itt magyarázatot adni, a gép részletes ismertetését átengedem a szakembereknek.

A gyártó által mellékelt illesztőprogram valóban hibás. Az inicializálás után JP 66H-val tér vissza a rendszerbe, ami nem

biztosít semmiféle programvédelmet, lévén a RESET gomb megnyomásával egyenértékű. Mivel a program a 7FE1H—7FFAH területen található, ezért minden olyan sztringművelet, amely a sztringterületet használja (alaphelyzetben az utolsó 50 bájtt), felülírhatja. Ennek kiszámíthatatlan következményei lesznek a nyomtató meghívásakor. Ezt általában nem tapasztaljuk 64 k-s gépeknél, mivel ott a memória közepére kerül az illesztőprogram. A 16 k-s gépeknél megoldást jelenthet az, ha a számítógép bekapcsolása utáni READY? kérdésre beírjuk a 32736 NL választ, majd ez után olvassuk be kezdetéről az illesztőprogramot.

A fentiek miatt módosítottam az illesztőprogramot a lista szerint. A program megmaradó részét változatlanul átvettem, csak két területre töltöttem egyszerre: 7FE2H—7FFBH-ig a 16 k-s gépek, FFE3H—FFCH-ig a 64 k-s gépek számára. A program startcíme: 7FC0H. Indításkor megvizsgálja, hogy 16 vagy 64 k-s gépbe olvastuk-e be. A DCB vektort ettől függően NEWPR vagy NEWPR 2 címmel írja felül. A RAMTOP vektorba (40B1H) beír a fenti címnél eggyel kisebbet, majd a 3000H bővítésre ugrik, ahol megtörténhet többek között a programvédelem is. Így elérhetjük, hogy az illesztőprogram mindkét géptípusnál a sztringterület mögé, a RAM végére kerül levédve.

Végezetül szeretnék segíteni a zFORTH-felhasználóknak. A nyomtató illesztéséhez nem kell az itt közölt vagy az eredeti programot beolvasni. Elegendő az alábbi négy szót begépelni: HEX A 5123 C!

LÁNYI TAMÁS

```

1      ; HT D-100 ILLESZTO PROGRAM
2      ;
3      ; LANYI TAMAS 1986.01.31.
4      ;
5      ORG 7FC0H
6      CR: EQU 0DH
7      NL: EQU 0AH
8      DCB: EQU 4025H
9      BASWR: EQU 4049H
10     ; KEZDETI BEALLITASOK
11     INIC: LD A,0
12           LD (0FFFFH),A
13           LD A,(0FFFFH)
14           OR A
15           LD HL,NEWPR2
16           JR Z,MASIK
17           LD HL,NEWPR
18           LD IX,DCB
19           LD (DCB+1),HL
20           DEC HL
21           LD (BASWR),HL
22           LD (40B1H),HL
23           JP 3000H
24     ; 16 kB-OS RENDSZER
25     NEWPR: PUSH BC
26           LD A,C
27           CP CR
28           JR NZ,TMTLP
29           CALL KIIR
30           LD A,NL
31           TMTLP: CALL KIIR
32           POP BC
33           RET
34     ;
35     KIIR: PUSH AF
36     VAR: CALL 5D1H
37           JR NZ,VAR
38           POP AF
39           OUT (0FDH),A
40           RET
41     ; 64 kB-OS RENDSZER
42     ORG 0FFE3H
43     NEWPR2: PUSH BC
44           LD A,C
45           CP CR
46           JR NZ,TMTLP2
47           CALL KIIR2
48           LD A,NL
49           TMTLP2: CALL KIIR2
50           POP BC
51           RET
52     ;
53     KIIR2: PUSH AF
54     VAR2: CALL 05D1H
55           JR NZ,VAR2
56           POP AF
57           OUT (0FDH),A
58           RET
59     ; / jelre a program indul
60     ;
61     END

```

SZÁMÍTÓKÖZPONT- VEZETŐK!

Ne dobják ki sérült
mágneselemeiket!

Gyorsan,
olcsón,

6 havi garanciával megajátjuk!

Commodore 64 számítógépükhöz
— az adatrögzítést megkönnyítő —
kiegészítő numerikus klaviatúrát
raktárról szállítunk,
1 éves garanciával.

Megvásárolható
az Econorg

1. sz. számítástechnikai
szaküzletében:

Bp. VI., Szinyei Merse Pál u. 1.
Tel.: 127-628

Égészségügyi Elektronikai Gm

1045 Bp., Erzsébet u. 14. X. 59.

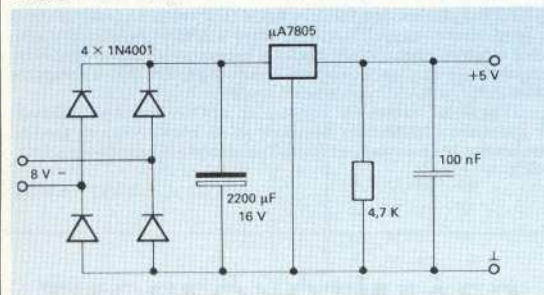
Telefonügyelet: Grósz Andor, 632-720 (9—17 óráig)

Külső tápegység ZX81-hez

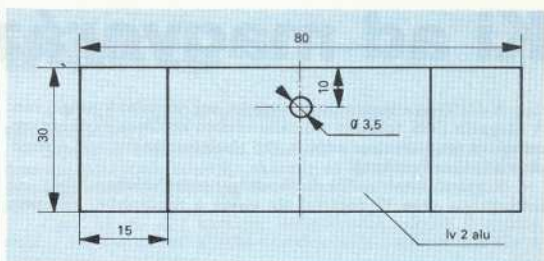
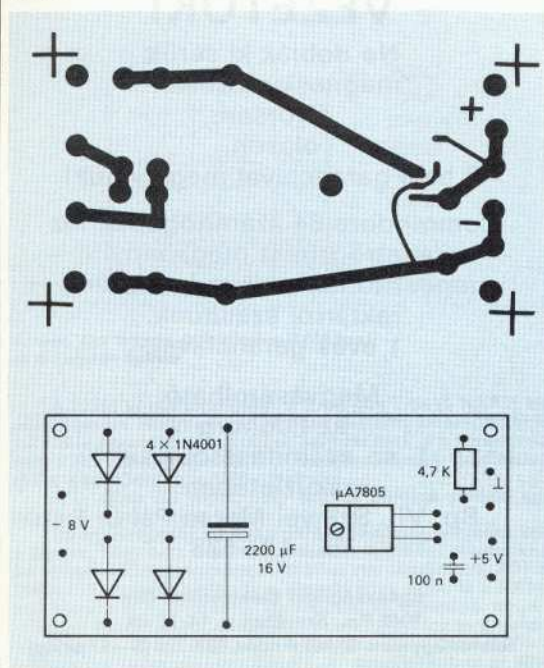
Ha ZX81-hez vagy más mikrogéphez több kiegészítést, illetve bővítést szeretnénk csatlakoztatni, akkor érdemes a már amúgy is túlfeszített üzemben dolgozó belső stabilizátort tehermentesíteni egy külső 5 V-os tápegységgel. Ez a tápegység +5 V-os stabilizált feszültséget szolgáltat, max. 3-400 mA terhelhetőséggel. Hálózati transzformátorként nyugodtan alkalmazhatunk csendőtranszformátort. A tápegység kapcsolási rajza az 1. ábrán, a NYÁK rajza a 2. ábrán és a hűtőlemez rajza a 3. ábrán látható.

Az elkészített tápegységet lehetőleg az alapgéppel együtt kapcsoljuk be és ki.

1. ábra



2. ábra



3. ábra

A tápegység elkészítéséhez szükséges anyagok:

- 1 csendőtranszformátor, hálózati zsinórral
- 1 hűtőlemez
- 4 1 N 4001 dióda
- 1 μ A 7805 IC
- 1 2200 μ F 16 V ek.
- 1 100 nF kk.
- 1 4,6 kohm el.

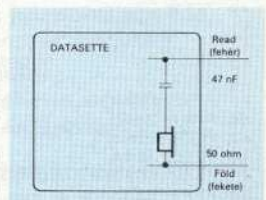
NAGY GÁBOR

Hallható programok a Datasette-en

A Commodore számítógép üzemeltetése során hasznos segítségnek bizonyult, hogy a kazettás magnóra kerülő vagy onnan betöltendő programot hallani is lehet. Csekély átalakítás után már jól megfigyelhető a szalagra vett startfrekvencia, a programnév és az azt követő program. Programok betöltésénél hallhatjuk az esetleges szalaghibából, a rossz felvételtől adódó vagy a beletörlési hibákat, programok szalagra vételekor pedig a felvenni kívánt jelet.

A Datasette átalakítása nagyon egyszerűen megvalósítható. A magnóban az olvasóvezeték és a föld közé egy 50 ohmos fülhallgatóbetétet kötöttem, soros kondenzátoron keresztül. Ez 47 nF-os fóliakondenzátor. A hallgatót a számlálómű melletti üres részbe szereltem be a következő módon: A SAVE-et jelző LED-dióda tartólemezének sarkába 3 mm-es lyukat fúrtam. Ide egy 10 mm hosszú, szigetelt talpatzatú forrasztófület csavaroztam fel. A hallgató egyik kivezetését ehhez forrasztva, megoldottam ennek rögzítését. A hallgató másik kivezetésére a kondenzátor egyik rövidere vágott kivezetését forrasztottam. A kondenzátor má-

sik kivezetése megy az olvasó vezetékre (fehér színű). A hallgató forrfulhöz forrasztott kivezetését kell földre kötni (fekete színű kivezetés). Az olvasó- és földre menő huzalokat a magnó belsejében úgy vezettem, hogy esetleges elmozdulásuk esetén se kerülhessenek a meghajtó gumik vagy a mechanika alkatrészei közé. A vezetékeket a magnó NYÁK lapján kötöttem azokra a pontokra, ahová az olvasó-, illetve földvezeték csatlakozik.



Az átalakítás a Datasette működését nem zavarja. A hallható program kis hangerejű, így másokat nem zavar. Közel fél éve dolgozom az így átalakított magnóval C64-es gépen. Az átalakítás a számítógép és a magnó üzemelésében semmi zavart nem okozott.

KOVÁCS LÁSZLÓ

Egy jó döntés!

Lokális

hálózat *Az IBM kompatibilis* *PROPER—16* *gépeken*

A helyi
hálózattal *lehetővé*
válik

a különböző szintű vezetők közötti munkamegosztás,
a fokozott ellenőrzés minden szinten,
a gépek személyi jellegének megtartásával
a kiterjesztett erőforrások közös használata,
az igényeknek megfelelően fokozatos kiépítés.

Jó ár/teljesítmény viszony, *világszínvonalú perifériák*

**SZÁMÍTÁSTECHNIKAI KUTATÓ INTÉZET
ÉS INNOVÁCIÓS KÖZPONT**



**1015 Budapest
I., Donáti u. 35—45.**

**Információ: SCI—L Számítástechnikai
Informatikai Fejlesztő Leányvállalat, Budapest I., Iskola u. 10. 1011. Tel.: 350-180**

Ahhoz, hogy a program jól sakkozzon, szükséges, hogy egyrészt ismerje a sakk szabályait, tehát minden helyzetben ki tudja választani a legális lépéseket, másrészt két állás közül meg tudja állapítani, melyik számára az előnyösebb. Már a legelső sakkprogramoknál sem okozott gondot a sakkjáték szabályainak beprogramozása, de a legjobb lépés kiválasztása annál inkább. A jelenlegi sakkprogramok főként az állásértékelésének módszerében különböznek egymástól, amelynek legkisebb eltérése is jelentős változást hozhat létre a legjobb lépés kiválasztásában.

A sakkhadállások értékelését a program úgy végzi, hogy azokat különböző szempontok szerint vizsgálja, és így egy-egy álláshoz rendelt érték(ek) alapján a számítógép meg tudja állapítani, hogy mely pozíció előnyösebb számára. Ez az algoritmus tulajdonképpen nem más, mint egy többváltozós függvény függvényértékének kiszámítása, melyet a következőben értékelőfüggvénynek nevezünk.

Gondoljuk át, hogy egy sakkozó valamely állás értékelésekor milyen gondolatmenettel választja ki a legjobb lépést. Először a sakkjáték alapvető stratégiái szerint próbál értékelni, és csak ezután nézi át a taktikai lehetőségeket. A számítógép esetében ugyanezt kell megvalósítani, csak konkrétan, vagyis a stratégiai alapelveket élesen el kell határolni egymástól, és pontosan meg kell határozni, hogy az egyes stratégiai elemeket milyen súlyozással vegye figyelembe az adott állás értékelésénél. Ez azért nehéz feladat, mert amíg egy sakkozó számára azok a stratégiai elvek egybeolvadnak, nem határolódnak el élesen, és könnyen kiszemelik, hogy egyes típusállásoknál mely stratégiai elemeket vegye nagyobb mértékben figyelembe, addig egy programban ezt pontosan meghatározni nagyon bonyolult feladat. Egy sakkprogramnak ezért több értékelőfüggvénye van. Nézzük meg konkrétan, hogy az értékelőfüggvényekben melyek a legfontosabb azonos stratégiai elemek!

Mindegyiknek figyelembe kell vennie az egyes állások anyagi értékét, a figurák mozgékony-ságát, a centrum ellenőrzését, az ellenfél királyával szembeni támadás lehetőségét, a saját király biztonságát, a gyalogszerkezetet, az egyes figurák megtámadását és védelmét. A leg-

fontosabb különböző stratégiai elemek: a figurák fejlődése, a sáncolás lehetősége és az egyes végjátéktípusokra vonatkozó legfontosabb elvek támogatása. Ezek alapján egy közepesen játszó sakkprogram minimálisan három értékelőfüggvénnyel rendelkezik. Ezeknek a bekapcsolódását, hogy mikor melyikkel dolgozik a program, általában a lépésszámtól teszik függővé.

Az első 10-15 lépésig olyan értékelőfüggvény dolgozik, amely a sáncolás lehetőségeit nagymértékben figyelembe veszi, illetve az egyes figurák ki-fejlődését segíti. Megakadályozza, hogy egy figurával sokszor lépünk egymás után, amíg a többi alaphelyzetében tartózkodik. Ezután következik a középjáték, amelyben az egyes stratégiai elemek súlyozását általában kiegyenlítik — nincs olyan stratégiai elem, melyet kiugróan nagyobb súllyal látnának el —, így a taktika előtérbe kerül. Majd a harmadik értékelőfüggvény bekapcsolódásának feltétele, hogy a táblán levő figurák anyagi értékének összege egy bizonyos érték alá essen. Ebben az esetben a program végjátéknak minősíti az aktuális állást, és az ennek megfelelő szempontokat veszi figyelembe. Így különösen számon tartja a királyok ún. oppozícióját, valamint, hogy gyalogbevétel lehetősége esetén az ellenfél királya a négyzetben belülről tud-e kerülni, vagy sem.

Oppozíció. Ha világos lép, fel kell adnia az oppozíciót, és sötét döntetlen tart: 1. Kc5 Kc7 2. Kd5 Kd7 3. Ke5 Ke7 4. d5 Kd7 5. d6 Kd8! 6. Ke6 Ke8! 7. d7! Kd8 8. Kd6 patt. Ha sötét lép, neki kell feladnia az oppozíciót és világos nyer: 1. — Kc7 2. Ke6! Kd8 3. Kd6 Kc8 4. Ke7 Kc7 5. d5 Kc8 6. d6 és a gyalog bemej.

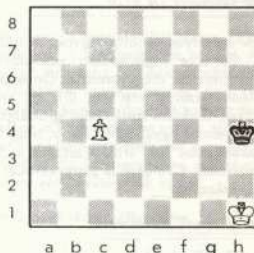
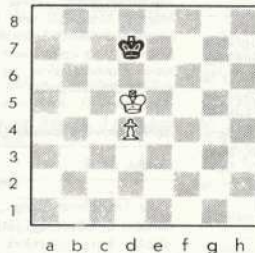
Négyzet szabály. Ha világos lép: 1. c5! és a sötét király nem tud belépni a négyzetbe, a gyalog bemej. Ha sötét lép: 1. — Kg4 vagy Kg5 és a sötét király a négyzetben van, utoléri a gyalogot és döntetlen: 2. c5 Kf5 3. c6 Ke6 4. d7 Ke7.

Anyag

Vizsgáljuk meg, hogy az előzőekben felsorolt stratégiai elemek tulajdonképpen mit takarnak. Köztudott, hogy a sakkban az egyik legfontosabb törekvés anyagi előny szerzése. Az anyagot a többváltozós értékelőfüggvény egyik változója-

BITEK ÉS FIGURÁK

Állásértékelés I.



Világos: Fischer — Sötét: Szpazszi.
Elutasított vezércsel

ként foghatjuk fel. S ez nem más, mint egy adott helyzetben a sakk táblán levő figurák összessége. Általában az anyagi előnnyel rendelkező játékos nyeri a játszmát, ezért ez nagy fontosságú. A köztudatban elterjedt, hogy ha a gyalogot egységnek vesszük, akkor a huszár és a futó háromegységnyi, a bástya öt-, a vezér kilencegységnyi értékű, a király pedig minden érteken felüli, mivel a király elvesztése egyben a játszma elvesztését is jelenti. Ezek az értékarányok azonban finomításra szorulnak. Az alapfokú sakk-könyvek is azt tanítják, hogy nyílt állásokban a futópár többet ér, mint két huszár, illetve egy huszár és egy futó. Ezért, ha a táblán futópár van, akkor a futó értékét nem három, hanem három és fél egységnek vesszük. Az is nyilvánvaló, hogy egy védett szabad gyalog többet ér, egy elszakított gyalog kevesebbet az alapértékénél.

Ha olyan sakkprogramot ír-nánk, amely a legális lépéseket generálja, és értékelőfüggvénye csak az anyagot veszi figyelembe, ez a program már egy sakkjátékra emlékeztető módon játszana. Nem számolna előre több föllépésig, csak az adott állást elemezné, és ha az ellenfél bármely figurája ütésben állna, akkor a legnagyobb értékűt ütné ki. E körülménytől eltekintve lépései véletlenszerűek lennének, mivel a lépéskiválasztásnak egyetlen

döntési kritériuma van, az anyag. A sakkjáték viszont sokkal magasabb rendű annál, semmint hogy csak az ellenfél figuráinak a leütését kellene figyelembe venni. Nézzük a további kritériumokat.

Mozgékony-ság

Az anyag után a sakkjáték második legfontosabb tényezője a mozgékony-ság; így a többváltozós értékelőfüggvényben is a második a mozgékony-sági komponens.

Nagyon fontos, hogy figuránk a tábla mely részén áll, és hány mezőre tud lépni. E tényező figyelembevételével játékuink során figuráink aktivitását fokozhatjuk. Ennek mértékét a leg-egyszerűbben összes figuránk megegyező valamennyi lépés-számanak összegéből állapíthatjuk meg. E változó kiszámításánál nem vesszük figyelembe a sakkadás ideiglenes állapotát; a számítást ebben az esetben is minden figurára elvégezzük, amely a félig legális lépések listáján szerepel, annak ellenére, hogy a konkrét lépés tulajdonképpen nem is legális, mivel királyunkat sakkban hagyja. David Levy e komponens és az anyag változó kombinációját egy példával mutatta be, amelyben az egyszerűség és érthetőség kedvéért feltételezi, hogy az értékelőfüggvénynek csak két változója van: az anyag és a mozgékony-ság.

1251 BUDAPEST, Fő u. 68.
Telefon: 154-090/661
Telex: 22-6835



C64, M08X, IBM PC XT kompatibilis rendszerekre ajánljuk programmagjainkat:

- főkönyvi könyvelés,
- költségfelosztás,
- állóeszköz-nyilvántartás,
- rendelés-nyilvántartás,
- bérelszámolás,
- adóelszámolás.

Bővebb felvilágosítással készséggel állunk rendelkezésükre!

1251 BUDAPEST, Fő u. 68.
Telefon: 154-090/661
Telex: 22-6835

Bérmé-
ntéses

TECHNOORG

Alkalmazástechnikai

Gmk

Budapest,

Rákó utca 11.

1 1 1 2

Alulírott utánvétellel meg-
rendelem a „Mikroelektro-
nikai berendezésorientált
áramkörök tervezése” c. ki-
adványt 1200 Ft ellenértékért
..... példányban.

Kérek tájékoztatást az
ebben a témában szervező-
dő tanfolyamról

Cím

Aláírás

Az LSI Alkalmazástechnikai Tanácsadó Szolgálat könyvkiadási tevékenységéről

Az elmúlt másfél év alatt közel 200 ezer példányszámban jelent meg szakkönyv az LSI ATSZ gondozásában. Ezek közül a legnagyobb siker dr. Úry László Commodore 64 című könyve, amely két kiadásban és többszöri utánnyomással eddig összesen 38 ezer példányban jelent meg és fogyott el. A könyv oly mértékben segítette a számítógép gyakorlati megismerését, hogy a belőle szerzett tudásanyagára épülve több száz ezer forint értékű program készítéséről számoltak be az olvasók.

Dr. Úry László Commodore 16 című, néhány hónapja kiadott műve 12 ezer példányban készült, ugyancsak utánnyomással.

Dr. Kovács Magda Egyszerűen a mikroszámítógépről című könyvének anyaga a szerzőnek a Technika folyóirat nívódíját elnyert cikksorozatára épült. A könyv elsősorban oktatóközpontok, szakiskolák, egyetemi vizsgára készülők, a mikroszámítógép alkalmazásának alapjait tanulók körében vált népszerűvé. 12 ezer példányban jelent meg, két kiadásban és többszöri utánnyomással.

Tízret megközelítő darabszámot, vagyis többszöri utánnyomást ért el az alábbi kiadványok:

Kriszán György—dr. Kovács Magda: A Zilog cég mikroprocesszor családjai

Donát János: Z80 software táblázatok

Dr. Úry László: Commodore 64 Info kártyával

Dr. Szemes Katalin és munkatársai: CP/M operációs rendszer Varga Imre: PASCAL Spectrumra és Commodore 64-re

Lipovszki György—Subai László—Beszeda Tamás: FORTH programozási rendszer és nyelv Erdős Iván és munkatársai: 1001 játék és a Graphics BASIC Commodore 64-en

Bartha Tamás: Az IEC busz és alkalmazása

Többszöri kiadást, illetve utánnyomást elért könyvek:

Dr. Úry László: Commodore 64 BASIC és felhasználói kézikönyv

Erdős Iván: Commodore 64 Assembler

Dr. Ádám Sándor: Népszerű elektronikai minilexikon

Dibuz Ágoston: Hardware katalógus. Alkatrészek, elemek

Salgó Iván: Hardware katalógus. Hazai készülékek

Szilassy Bertalan: 18086 mikroprocesszor utasításkészlet

Előkészítés alatt álló kiadványok:

Dr. Tokodi Jenő: Laser mikroszámítógép-család

Nagy—Várdin—Knizslár: Adatfeldolgozó programcsomagok

Ágoston Mihály: Mikroszámítógépes újdonságok
Dr. Ferenczi Antal: C64 Start
Dr. Szemes (szerkesztő) és mások: IBM PC sorozat

Robot sorozat. (Sorozatszerkesztő: dr. Hajnal Miklós—Kardos Zsuzsa)

Az LSI ATSZ kiadványai egy-két kivételtől (például katalógusok) eltekintve nem részesülnek központi támogatásban. Így a kiadványok viszonylag alacsony ára a 24—32%-os kereskedelmi árrés mellett a nagy népszerűségnek, a magas példányszámnak — a kiadás ugyanis kb. 5000 darab felett önfenntartó — köszönhető.

SZIKLAI KLÁRA
LSI ATSZ

Commodore 16 — Commodore plus/4 Bevezetés a BASIC nyelvbe 1. rész

A könyvet nyugodtan nevezhetnénk munkafüzetnek is, hiszen tematikája és alapvető céltűzései a BASIC nyelv önképzés alapján történő elsajátítását célozzák. Forrásgyűjteményként a mindkét géphez mellékelte angol nyelvű programok, valamint egy önképző tanfolyamsorozat 1. része.

A könyv 15 leckére és 3 függelékfejezetre osztható. Egényi munkát vár az olvasótól; feltételezi, hogy a feladatok megoldásához rendelkezésre áll egy számítógép és egy kezeltetés egység vagy lemezegység. A 15 lecke megértése és kipróbálása elvezet a legalapvetőbb ismeretektől az egyszerűbb játékok készítéséig. Minden lecke egy bevezető részből, több gyakorlatból és feladatokból épül fel. A feladatok megoldásainak vázlataihoz a szöveg után üres helyek vannak, hogy az olvasó oda jegyezhesse fel a kérdésekre adott válaszait, majd azokat később is ellenőrizhesse. A tanfolyam igyekszik az olvasót áttekinthető programírással nevelni, mert a bonyolultabb feladatokat csak így fogja tudni megoldani.

A tanfolyam első része alapvető ismereteket ad a BASIC nyelv használatáról és lehetőségeit az egyszerűbb feladatok megoldására. Ez az az ismeretszint, amely ahhoz szükséges, hogy valaki felhasználóként biztosan és idegenkedés nélkül tudja kezelni a számítógépet.

A könyv tipográfiája megegyezik az eredeti angol kiadásával, amely jól olvasható és egyértelmű jelöléseket használ. A mű szerzőjének nevét sajnos csak a forrásanyagból tudhatjuk meg, pedig biztosan vállalja munkáját, amely igazán jól ismertül.

SIEGLER GÁBOR

Számítógép-tulajdonosok, figyelem!

Mindenfajta printerhez
„védtett technológiával”,
speciális nejlon anyagból
szalagot készítek
a kazetták felújításához.
Minden típusú kazettát
rövid határidővel javítok,

carbonszalagos kazetták kivételével.

Nem szétszedhető kazettákat
átalakítom szétszedhetővé.

MPS 801-es kazettákat
azonnal cserélem.

Postai utánvétellel is küldhető.

ÓDOR JÓZSEF
kisiparos



Telefon:
175-716

Bp., Tanács krt. 8. 1052

Felajánlunk
megvételre

1 db FELLER OCR—2002 típusú
optikai bizonylat- és oldalolvasót,
BPI mágnesszalagegységgel,
valamint a berendezés tartozékait:

— 1 db IBM gömbfejes írógép
2 db OCR betűtípusú gömbfejjel,
— 1 db bizonylatbélyegző,
valamint tartalék alkatrészek.

DATORG

KÜLKERESKEDELMI ADATFELDOLGOZÓ
ÉS SZERVEZŐ RT.

Érdeklődni lehet
Perity László munkatársunknál.

Bp. V., Dorottya u. 6.

Telefon: 184-055/1429 m.

ORSZÁGOS SZERVIZHÁLÓZATUNK

útján VÁLLALJUK:

az elektronikus rendszerű írógépek,
számológépek,
gyorsmásológépek,
ügyviteltechnikai berendezések,
személyi számítógépek,
kisszámítógépek,
számítógéprendszerek
és egyéb elektronikus berendezések

- üzembe helyezését,
- garanciális és garanciaidőn túli javítást,
- szervizkiszolgálását.

INFORMÁCIÓTECHNIKAI VÁLLALAT

Központ: Budapest V., Bécsi u. 8.
Telefon: 184-899

Levél cím: 1369 Budapest, Pf. 314.
Telex: 22-4381, 22-6841





SZOFTVERKERESKEDELMI ÉS FEJLESZTÉSI BETÉTI TÁRSULÁS

ajánl

IBM PC/XT és azzal kompatibilis gépekre
rendszerfejlesztési és programozási munka
hatékonyágát növelő eszközöket.

INFOMIX

Újdonság!

Szabad, azaz kódolatlan — akár szöveges — információk tárolására és rendkívül rugalmas visszakeresésére alkalmas szoftver. Párbeszédés formában — számítástechnikai ismeret nélkül — építheti és lekérdezheti információbázisát.

dACCESS III.

a SOFTWARE '86 első díjas termékei

Szabványos dBASE III. fájlokat és indexeket kezelő hostnyelvi relációs adatbázis-kezelő rendszer. Fájlmanipulációs szolgáltatásai megegyeznek a dBASE III. megfelelő szolgáltatásával. (Host-nyelv, BASIC, C, FORTRAN stb.)

FWINDOW

Használatával látványosan felgyorsítható a képernyő-kommunikáció, kényelmesebbé válik az ablakkezelés. (Szolgáltatásai tetszőleges programból érhetők el.)

VIEWS

Az operációs rendszer felhasználó-orientált kiterjesztése, szolgáltatása a komplex szövegszerkesztéstől, a kvázigrafikus ábrageneráláson, a legkorszerűbb nyomtatási szolgáltatáson keresztül a nagyméretű virtuális képernyőkezelésig terjed.

A VIEWS hidat teremthet bármely felhasználói vagy rendszerprogramban megjelenő képernyők között, akár előre programozottan is.

ELTPAINT

Tetszőleges programból hívható eszköz grafikai ábrák megjelenítésére, elmentésére, kinyomtatására.

MicFORTH

Elsősorban az ügyviteli adatfeldolgozási alkalmazásokat támogatja. Tartalmazza a FORTH—79, FIG—FORTH, MVP—FORTH és a STRING—FORTH szavait, továbbá screen-editort, lebegőpontos számbázis-kezelőt, FORTH adatbázis-kezelőt, hibakezelőt és javító programokat (a program tetszés szerinti helyen megszakítható.)

Budapest, Petrezselyem u. 6.
1024
Telefon: 351-950

IBM
kompatibilis
professzionális
személyi számítógépek

COMMODORE PC 20

lízing:

~~688 000,— Ft~~

helyett

496 000,— Ft

NOVOTRADE

NOVOTRADE RT.
Budapest
Balzac u. 35.
Telefon: 402-954

Száztizenötezer

Tavaly Ausztriában 115 ezer mikroszámítógépet adtak el, több mint két és fél milliárd schilling értékben. Erre az évre mintegy kétszáz ezer darabos eladással számolnak. Az otthoni számítógépek iránti keresletet az is élénkíti, hogy az 1985/86-os tanév kezdetén a középiskolákban bevezették a számítástechnikai oktatást.

Japán fordítógép

A Toshiba cég olyan számítógépet gyártott, amely 90 százalékos pontossággal fordít angolról japánra. Az UX—700-as típusú, 32 bites felolgozási szélességű minigép 5000 szó/óra sebességével háromszor olyan gyors, mint egy gyakorlott fordító.

A gép az angol és japán szöveget egyidejűleg megjeleníti a képernyőn. Többjelentésű szavak esetén az összes variáns megjelenik. A tároló 130 000 szó befogadására képes. Megosztása a következő: 30 000 szavas általános szótár, 50 000 szavas szakkifejezéstár, 50 000 szó szabadon bevethető.

Profi

Commodore 64?

Rengeteg Commodore 64 típusú, eredetileg házi számítógépnek szánt mikrogépet vásároltak a hazai közületek, természetesen professzionális feladatok megoldására. Időközben persze kiderültek a gép korlátai, s a panaszok többsége a kicsi, lassú és nem eléggé megbízható hajlékonylemezességre vonatkozott.

Többen Winchester-tárat csatlakoztattak a Commodore 64-hez. Ez esetben viszont fordított helyzet állt elő: a Winchester-tár lehetőségei mellett most a Commodore 64 teljesítménye törpült el. Ezen az elmentmondáson segített most a Novarat Kiszöveket, kifejlesztve a WOS nevű Winchester operációs rendszerét. Ez lehetővé teszi, hogy maximum 8 db Commodore 64-et kapcsoljunk rá egy vagy két Winchester-tárra, melyek kapacitása lehet 5, 10, 15, 27 vagy 70 Mbájt.

Maga a WOS egy BASIC-bővítés, mely tartalmazza a Winchester kezelő utasítások csoportját, a képernyő szerkesztését elősegítő utasításokat, valamint az adatbevitel meggyorsítására szolgáló parancsokat. A Winchester mellett továbbra is korlátozás nélkül használhatók a hajlékonylemez tárolók.

Közben elkészült már a WOS operációs rendszernek az a változata, amely lehetővé teszi szintén maximum 8 db Commodore 620/710 típusú gép csatlakoztatását egy vagy két Winchester-tárhoz.

Hibatűrő számítógépek

A számítógépek hibáinak és állásidőinek egyre növekvő költsége különösen akkor okoz súlyos veszteségeket, ha a számítógép a vállalat termelésében kulcsszerepet tölt be.

A megbízhatóság iránti igény különösen azoknál az alkalmazásoknál nagy, ahol a vállalati folyamatirányítást és az adatfeldolgozási-adminisztrációs munkákat közvetlenül számítógéppel végzik.

Ez ráirányítja a figyelmet a hibatűrő számítógépekre, amelyek működőképesek maradnak akkor is, ha a rendszer egy része meghibásodik. A számítógép elromlásának a kiküszöbölését a gyártók redukáltan egységek és elemek beépítésével érik el.

Az amerikai Frost and Sullivan piackutató intézet szerint e fokozott megbízhatóságú rendszerek kereslete az évtized végéig hatalmas mértékben, évi 70%-kal növekszik.

Az alkalmazásokat tekintve a hibatűrő számítógépek piacának túlnyomó részét a közvetlen feldolgozás adja. A piac második, egyben legdinamikusabban növekvő szegmense az elosztott adatfeldolgozás.

Táskaszámítógép

36 millió dolláros kormány-megrendelésre táskaszámítógépet fejlesztett ki az IBM. A három és fél kilogrammos készüléket az adóhivatalnokok fogják használni. Külön érdekessége az új gépnek a beépített modem, amely lehetővé teszi,

hogy az adóhivatalnokok a vizsgálat helyszínéről kapcsolatot teremtsenek a központi nagyszámítógépekkel. A 256 kb-jaos gépet a nagyközönségnek is árusítják, előzetes hírek szerint 2495 dolláros áron.

Zacskóban

A Microkey KFTT felkészül a népszerű Primo számítógép zacskóban való árusítására. Az előreláthatólag 8000 forintba kerülő készlet az alapgép valamennyi elemét tartalmazza, továbbá egy ismertetőt, melyben az összeszerelés hogyanját találjuk. Az ennek megfelelően összeállított gép a későbbiekben bővíthető nyomtatóillesztővel, lemezmaghajtó illesztővel és színes egységgel. Egyelőre 200 darab kerül árusításra, mely az igényeknek megfelelően növekedhet.

Tiltakozás

Több vezető amerikai tudományos intézmény támadja a Reagan-kormányzatot, hogy meg akarja akadályozni külföldi diákoknak bizonyos egyetemi számítógépek használatát. Az szivárgott ki ugyanis, hogy a Fehér Ház rendeletet készít elő, mely szerint a COCOM által korlátozott országok (azaz a KGST-országok és Kína) diákjai nem használhatják az USA négy új super számítógéppontjainak gépeit. Ha a rendelet hatályba lép, a korlátozás a világ több mint 200 super számítógépre vonatkozik majd — mondotta Robert Park, az Amerikai Fizikai Társaság képviselője —, többek között valamennyi IBM 1190 gépre, amelynek a teljesítményét az új IBM vektordobozzal növelték meg.

Autóversenyen

Már komoly hagyománya van hazánkban a számítógépek autóversenyen való alkalmazásának. A hetvenes években nagyszámítógépes rendszerrel próbálkoztak, mely helyhez kötöttségével, szolgáltatásainak kötegel jellegével, valamint a magas gépberleti árával természetesen nem felélt meg a rendszeres használat követelményeinek.

A nyolcvanas években átdolgozták a rendszert egy VT—20 típusú számítógépre. 1984-ben volt a premier a Porán Rallyen, s az új szolgáltatás nagy sikert aratott. Az eredményszolgáltatást két részre bontották. Ahogy az utolsó kocsit elment a gyorsasági futamról, rögtön tudták eredményt adni. A verseny végén pedig összesített eredménylistát nyomtattak.

Tavaly az egész rendszert átirták Commodore 64 gépre, több önálló feladatra bontva. A Budapest Rallyn 12 gépet állítottak munkába, melyekhez több monitort kapcsolva javíthatták a tájékoztatást. A programokat a KSH Számítástechnikai és Ügyvitelszervező Vállalat készítette, mely megrendelés esetén gépekkel, az üzemeltetést biztosító gépekkel száll ki a kért helyszínre.

1986-ra, nem utolsósorban a hazai FORMA—1-es pálya megnyitáshoz kapcsolódva, igen komoly tervek kövöcsölődnak. Szeretnék a rendszert egy igazi professzionális gépre teleíteni, e célra a VIDEOTON VT—16-os gépet szemelték ki. Ugyanakkor a felhasználáshoz igazodó mobilitás biztosítása érdekében szeretnék a gépet egy autóbuzson elhelyezni, mely így egy-egy verseny információs központjaként funkcionálhatna.

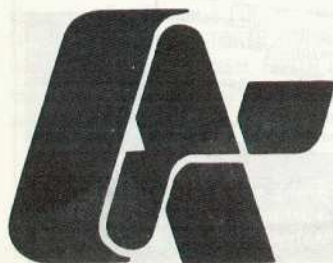
Személyiszámítógép-javítás, karbantartás közületeknek, magánszemélyeknek. Egyedi megrendelés alapján kiegészítő berendezések gyártása. Pl. Sinclair fényceruza, Joystick Interface, oktatósi intézmények kabinet kialakítása.

Pásztor Ferenc személyiszámítógép-javító és -karbantartó kisiparos. Szalónk, Mátyás király u. 2. V/3.

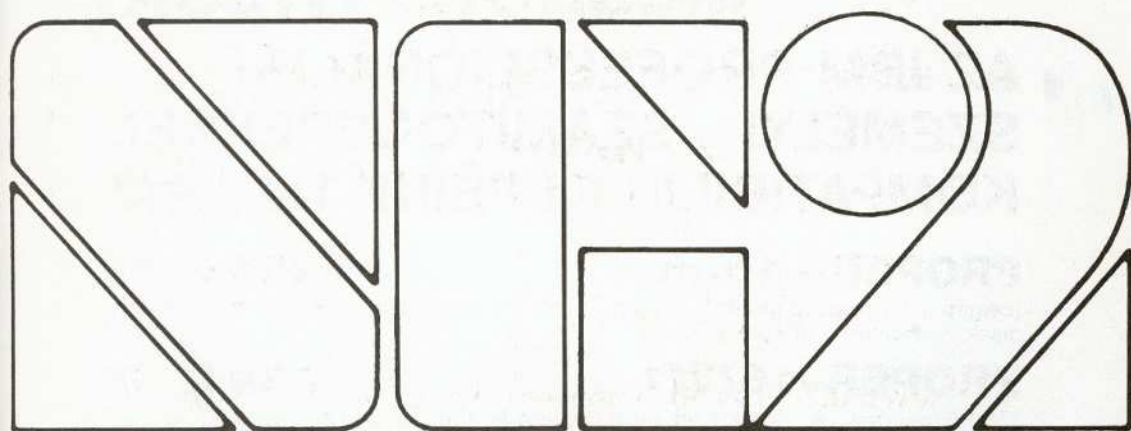
Apple II és vele kompatibilis számítógépekhez ajánljuk az alábbi

épületgépészeti programcsomagokat:

- *Klímaszimuláció:* ezzel a programmal méretezhető bármely CHREYS-LER—FÜTŐBER rendszerű építőelemes klímaberendezés vagy annak egyes elemei. A klímarendszer elemei tetszőlegesen választhatók ki, melyet kapcsolási rajzon követhetünk. Az eredmények a képernyőn levő kapcsolási rajzon a megfelelő pontokra beíródnak, melyek sornyomtatóra is lekérhetők.
- *Kéményméretezés:* a program lehetővé teszi, hogy bármilyen ismert típusú túlnyomásos tűzterű gázkazán füstcsatornáját és kéményét mind áramlástan, mind hőtechnikai szempontból méretezhessük.
- *Hőcserélő méretezése.*
- *Épületek hőszükséglet-számítása.*
- *Építészeti rajzolóprogram.*
- *Mesterséges szellőzőrendszerek méretezése:* a program alkalmas ventilátoros szellőzőrendszerek pontos áramlástechnikai méretezésére, a méretezett rendszerek nyomásdiagramjának felrajzolására.
- *Kétcsöves szivattyús központi fűtési rendszer méretezése.*
- *Deflektoros kéményméretezés.*



*Alkotó Ifjúság Egyesülés
Számítástechnikai Iroda
Budapest V., Garibaldi u. 2.
Levél cím: 1519 Budapest, Pf. 330.
Telefon: 112-666, 113-608
Telex: 22-7272*



**Az SG2–MKB Pénzügyi Informatikai Kft
kínálja a következő
rendszereket és szolgáltatásokat:**

- Pénzintézetek kölcsön- és betétkezelési tevékenységének számítógépes feldolgozási rendszere
- Bérszámfejtési rendszer
- Készlet- és raktárgazdálkodás
- Pénzügyi és számviteli információs rendszerek szervezése
- Termelés-előkészítő és irányítási rendszer
- Partnervállalati címjegyzék
- Software- és szakemberexport

DÖNTÉSÉT MEGKÖNNYÍTJÜK!

AZ IBM PROFESSZIONÁLIS SZEMÉLYI SZÁMÍTÓGÉPEKKEL KOMPATÍBILIS GÉPEINK ÚJ ÁRAI

PROPER-16/m **266 E Ft**

(célgép, LAN hálózati munkahellyé bővíthető) 256 KB operatív tár, 2 x 720 KB floppy diszk, monochrom monitor

PROPER-16/W1 **439 E Ft**

256 KB operatív tár, 720 KB floppy diszk, merevlemezes tár (27 MB), monochrom monitor

PROPER-16/W2 **549 E Ft**

256 KB operatív tár, 720 KB floppy diszk, merevlemezes tár (27 MB), monochrom monitor, streamer egység

PROPER-16/MEGA **699 E Ft**

704 KB operatív tár, 720 KB floppy diszk, merevlemezes tár (27 MB), színes monitor, streamer egység

ÚJDONSÁGAINK 1986-BAN

PROPER-16/G nagy teljesítményű grafikus munkahely
1500 E Ft-tól

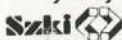
PROPER-16/MT a PROPER-16/W1-nél mintegy háromszor na-
gyobb teljesítményű gép
850 E Ft

**ÁRAINK 12 HÓNAP GARANCIÁT TARTALMAZNAK, TOVÁBBI
12 HÓNAPRA A HARDWARE ÁR 6%-ÁÉRT HARDWARE SZER-
VIZSZOLGÁLAT**

TOVÁBBI BŐVÍTÉSEK ÉS SZOLGÁLTATÁSAINK többek között:

Széles körű software-választék, bővítési lehetőségek világszín-
vonalú perifériákkal, országos szervizhálózat, kulcsra kész alkal-
mazói rendszerek, hálózatba integrálási lehetőség.

Kérje új árjegyzékünket!



Számítástechnikai Kutató Intézet
és Innovációs Központ

Információ: SCI-L Számítástechnikai Informatikai Fejlesztő Leányvállalat

Bp. I., Iskola u. 10. Telefon: 350-180

SCITEL Számítástechnikai Fejlesztő Leasing Leányvállalat

Bp. I., Donáti u. 35-45. Telefon: 350-180