

mikro

számítógép

magazin

Ára: 30 Ft



Budapesti Tavasz Fesztivál
1988 március 19/26





— μ '88 —
**III. ORSZÁGOS
 MIKROSZÁMÍTÓGÉPES
 TALÁLKOZÓ**
március 19–26.

A találkozót a SZÁMÍTÁSTECHNIKA MINDENKIÉ, A SZÁMÍTÁSTECHNIKA MINDENKIÉRT informatikai kiállítás köré épül. Hagyományosan jelen lesznek az NJSZT területi szervei, a Hobby Computer Club és a számítástechnika számos híres egyénisége.

Érdekes versenyek, vetélkedők, rendezvények!

Az oktatóprogramok egyéni és országos csapatversenye a legkülönbözőbb témákban, nemcsak pedagógusok számára indul. Látványos és műszakilag is érdekes a házi építésű számítástechnikai eszközök országos versenye. Rendszerek, perifériák és számítógéppel vezérelhető modellek, játék vasutak, autók és robotok mérik össze tudásukat.

A „Sütés-főzés számítógéppel” vetélkedőn azok indulnak, akik valamilyen módon felhasználják a számítógépet ételek, italok készítéséhez.

Újdonság a találkozón „Az ember elgépiesedik — a gép elemeriesedik” című karikatúr pályázat, amely hagyományos eszközökkel és számítógéppel készített karikatúrák kiállítása. A rendezvényen a látogatók is készíthetnek gunyoros rajzokat, és nemzetközi zsűri által értékelt karikatúrákból összeállított, reprezentatív albumot nyerhetnek vagy vásárolhatnak.

A találkozót egészségügyi szekciója a számítástechnikával foglalkozó mozgáskorlátozottak, vakok

és gyengénlátók taborát reprezentálja. Bemutják az első, vakok számára kifejlesztett számítástechnikai munkahelyet, és mindenki kipróbálhatja a KFKI-ban készült „beszélő” számítógépet. Itt ismerkedhetnek meg a látogatók „A rák ellen — az emberért, a holnapért” alapítvánnyal is.

Az olvasó—szerkesztő találkozón az érdeklődők elbeszélgethetnek a számítástechnikai lapok szerkesztőivel.

A rendezők azokra is gondoltak, akiknek feltalálói hajlamaik vannak. A „Szereld össze” akció során egy zacskóban kapott alkatrészszámból kell valamilyen „működő” berendezést összeállítaniuk az erre vállalkozóknak.

A sakkzók a legújabb gépekkel mérhetik össze tudásukat. Gép géppel is versenyez: kié a legjobb sakkzóprogram?

Hazánkban első ízben itt tekinthetik meg az érdeklődők a számítástechnikai eszközök múzeumát.

A kiállításon a számítástechnikában élenjáró cégek mutatják be újdonságaikat, melyekből vásárolni is lehet.

A rendezvényen kedvezményesen árusítják a Mikroszámítógép Magazin régebbi számait, és előfizetést is elfogadnak.

A μ '88-at nem szabad kihagyni!



mikro számítógép magazin

A NEUMANN JÁNOS SZÁMÍTÓGÉPTUDOMÁNYI TÁRSASÁG ÉS A KISZ KÖZPONTI BIZOTTSÁG LAPJA

A kiadvány a Tudományos-vezetési és Informatikai

Intézet együttműködve készül

A szerkesztőbizottság vezetője:

Kovács Győző

A szerkesztőség munkatársai:

Babos János
(tervezőszerkesztő)
Bakos Tamás
(programozástechnika)Broczkó Péter
(hírek)Énekes Ferenc
(KISZ)Jakab Ágnes
(olvasószerkesztő)Kovács Győző
(levelezés)Krasznai Éva
(diákrovat)Lindner László
(sakk)Petróczy Judit
(könyvek)Pinke György
(NJSZT, alkalmazások)Simonyi Endre
Szebenszki Sándor

Szulovszky Csaba

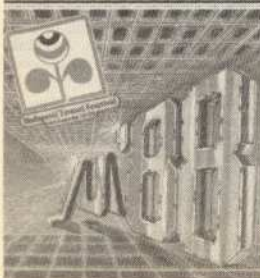
Tamásné Lakó Erika

Terebessy Ákosné
Varga András

(TII, iskola-számítógép)

Vizessy Mária
(µinform)

Címképünk: Ramocsai Imri munkája

Felelős szerkesztő:
Könyves Tóth PálSzerkesztőség:
1027 Budapest, Fő u. 68.
Telefon: 154-250Levél cím:
1371 Budapest
Pf. 433.Kiadja az Ifjúsági Lap-
és Könyvkiadó VállalatFelelős kiadó:
dr. Király G. István
igazgatóKiadóhivatal:
1065 Budapest, Révay u. 16.
Telefon: 116-660Terjeszti a Magyar Posta
Előfizethető a hírlapkezelés
hivataloknál
és a Posta Hírlapelőfizetési
és Lapellátási Irodáján
(1900 Budapest XIII.,
Lehel út 10/A.)
vagy átutalással a 215—96 162
pénzforgalmi jelzőszámra.Megjelenik havonta
Egy szám ára 30,— Ft
Előfizetési díj:
egy évre 360,— Ft
fél évre 180,— Ft
Külföldön terjeszti
a Kultúra,
1389 Budapest, Pf. 149.
és a Magyar Média
1932 Budapest, Pf. 279.
86—0253Szikra Lapnyomda
Budapest (88—40)
Felelős vezető:
Csöndes Zoltán vezérigazgatóINDEX: 25 629
ISSN 0236-6088

Tartalom

III. Országos Mikroszámítógépes Találkozó	2
Adok—veszek—cserélek	17
Mit tud a LINGUASOFT?	20
Konzervlevelek	22
Mikrokalauz	23
Logikai szintjelző	24
Zene és tánc számítógépen	27
Egy sarokkal olcsóbb	30
Programtermékek	31
Univerzális perifériakártya	32
Mi és a számítógép, a televízió, a video és egyebek	33
Újfróld	40
Olvastunk . . .	42
µINFORM	44

ISKOLA-SZÁMÍTÓGÉP

TechnoMIR	3
HT-1080Z Commodore floppival	4
Gyorsított rendezőprogram	5
Mire használjuk és mire ne használjuk a számítógépet?	6

DIÁKROVAT

Programcsemegék	7
Bázisjáték HT-re	8
Megszakításvezérelt manómozgató C64-re	8
ZX-Spectrum gyors feltöltése	9
Hangzó billentyűk Plus/4-re és C16-ra	9
Óra a keretben	10

PROGRAMOZÁSTECHNIKA

BASIC és gépi kód	11
Z80 programok haladóknak Spectrumra és Primóra	12
Számítógépes grafika Pascalban. Sík transzformációk	15
Bináris fák	18
A képernyő balra forgatása	19

µPROGRAMOK

Szekvenciális fájl kezelő rutin	34
256 szín ZX-Spectrumon	35

µKLUB

Integrált szoftver	36
Adom a magyarázatot! Ki ad magyarázatot?	38
A Mátra alján . . .	39

SAKK

AZ OLVASÓ ÍRJA	46
----------------	----

KÖNYVEK

HÍREK, ÉRDEKESSÉGEK	48
---------------------	----

PONTVADÁSZAT	48
--------------	----

„Nincs olyan bölcs a világon, ki még igen sok haszonnal ne vehetné tudományait, mint viszont alig van olyan tudatlan a földkereken, kitől egyet s más nem lehetne nagy haszonnal tanulni.”

(Széchenyi István: Hittel)

μ '88

III. ORSZÁGOS MIKROSZÁMÍTÓGÉPES TALÁLKOZÓ

március 18-23.

Ha március, akkor Budapesti Tavasz Fesztivál; ha Budapesti Tavasz Fesztivál, akkor Utazás kiállítás; ha Utazás kiállítás, akkor μ'88 — mondhatnánk logikai sorba rendezve a tavasz találokhoz kapcsolatait.

Most már kezdem remélni, hogy 1986-ban hagyományt teremtettünk, amikor a Budapesti Tavasz Fesztivál és az Utazás kiállítás vezetőivel közösen elhatároztuk, hogy ezután minden évben megrendezzük a mikroszámítógépet kedvelők és a szakma nagy tavaszi találkozóját.

Ha az 1987-es évre visszatérünk, megállapíthatjuk, hogy számítástechnikai kiállításokból bő termést hozott. Budapesten és vidéken vállalatok, kutatóintézetek, társadalmi szervezetek szinte havonta rendeztek ilyen-olyan elektronikai, számítástechnikai, szoftver-, alkalmazói és még ki tudja, milyen kiállításokat. Ez a „termés” tulajdonképpen nem rossz, ha van elég kiállító és van elég érdeklődő. De sajnos sokszor sem az egyik, sem a másik nem jött össze. Voltam például olyan kiállításon, ahol bizony a kiállító többet volt, mint a közönség. Erről az egyik műegyetemi professzor mondása jut eszembe, akinek az óráit a hallgatók nem nagyon látogatták: „Egyszer még megérem, hogy több lesznek, mint a hallgatóim.” Az idén ezt már számítástechnikai kiállításban megértük, és ha a tendencia folytatódik, akkor ez a kiállítás-infláció súlyos problémát fog okozni a szakmának is, hiszen se a kiállító (az eladó), se a látogató (a vevő) nem fogja tudni, hogy hova menjen, ahol megfelelő információt adhat, illetve kaphat, továbbá hogy hol tehet szert értékes kapcsolatokra.

Úgy gondoljuk, hogy a μ'88 ebben a kiállítás-öbönben is hézagpótló szerepet tölt be, és az előző évek hagyományait követve, sok kiállítót vonz, és még több látogatót segít hozzá az informatika alkalmazásainak megismeréséhez.

A μ'88 hagyományos, szinte már patinás rendezvénye az SZM SZM — a számítástechnika mindenkié, a számítástechnika mindenkié — kiállítás, tavaly óta már vásár. A rendezőség a kiállítókat minden évben arra kéri, hogy azokat az eszközeiket és programjaikat hozzák el bemutatni, amelyeket az informatikában nem képzett emberek is könnyen használhatnak. Azt szeretnénk, ha a kiállítók látogatói ötleteket kapnának, hogy melyek azok a rendszerek, amelyeket a munkahelyükön, esetleg az otthoni környezetükben is alkalmazni tudnak, ily módon is segítve az informatika társadalmi méretű elterjedését. Az SZM SZM valószínűleg abban különbözik a többi hasonló kiállítástól, hogy itt általában a fejlesztők és alkalmazók elsősorban nem a le-

geslegjübb eredményeket (persze azokat is), hanem leginkább bevált és főleg széles körben alkalmazható, talán azt is mondhatjuk, hogy népszerű eszközeiket mutatják be a látogatóknak. Csak példaként említhetjük, hogy be szeretnénk mutatni a posta 1988-ban üzembe álló videotex rendszerét, hogy a hálózat jövőendő felhasználói már jó előre megismerkedhessenek ennek az olyan sok országban népszerű rendszernek a szolgáltatásaival.

Az idén is — mint korábban — külön gondot fordítunk az iskola-számítástechnika eredményeinek bemutatására. Ami az előző alkalmakkor jól sikerült, azon most sem változtatunk; ebben az évben is pályázatot hirdettünk a legjobb oktatási programok, módszerek díjazására. A számítástechnikát alkalmazó tanárok és diákok közös fóruma a nem pedagógusok részére is nagy élmény szokott lenni és igen hasznos alkalom a tapasztalatok kicserélésére. Ezel a lehetőséggel — reméljük — ebben az évben is jócskán élnek majd mind a diákok, mind a pedagógusok.

Az idén iskolai programunk külön színfoltja lesz a LOGO-bemutató; ez a magas szintű programozási nyelv különösen a legifjabb számítógépes gondolkodásának fejlesztésére alkalmas. Szándékosan nem írtam programozástani, ti. a gyerekek ezt is megtanulják, de végül is nem ez a cél. A cél, ismétlem, az algoritmikus gondolkodásra, a problémamegoldásra való hajlam kifejlesztése. Sajnos a LOGO eddig nem volt népszerű sem az iskolákban, sem a házi számítógépes munkában, de reméljük, hogy ezen együttes erővel sikerül változtatni.

A Neumann János Számítógéptudományi Társaság keretében egyre szorban terebélyesedik a hazai szoftver, hardver és alkalmazói amatőr mozgalom. A μ'88 rendezői az idén is köteleességüknek érzik, hogy a mozgalom eredményeit, az amatőrök által kifejlesztett új eszközöket, programokat, alkalmazói rendszereket ne csak a nagyközönséggel, de a szakmával is megismertessék. Szeretnénk egy fórumot is rendezni, ahol a profi számítástechnikusok — nevezetesen gyári konstruktorok, kutatók, mérnökök — az amatőrökkel találkoznak, és néhány, valamennyiüket érdeklő problémát megvitatnak, mint például, hogy használja-e a szakma az amatőrök eredményeit vagy segítik-e a professionalista intézmények az amatőröket, és ha nem, akkor mit kell az együttműködés érdekében tenni. Azt hiszem, hogy egy olyan év kezdetén, amikor a műszaki haladás támogatása, a műszaki értelmiség helyzete és megbecsülése az állami politika egyik legfontosabb

kérdésévé vált, különösen fontos volna ennek a problémának a megvitatása.

Annak ellenére, hogy többször is megpróbáltuk, nem tudtunk lényegesen előbbre lépni a hazai klubok összefogása és tevékenységük fellendítése érdekében. Sok helyen, például művelődési házakban, iskolákban, társadalmi szervezetek keretében igen virágzó klubélet folyik, máshol semmi. Talán egy, a klubokról szóló újabb tapasztalatsereget segítené a mozgalomnak új célokat és mozgósító feladatokat adni.

Nem felejtjük ki a programból a számítástechnika vagy az informatika népszerűbb alkalmazásait sem, mint például a játékokat. Ezek közül elsősorban a sakkot és a bridszet játszó gépekkel rendezett versenyeket, sőt a tavaly nagyon jól sikerült szimuláns is megrendezzük, egy kicsit más formában.

Meghirdítettünk egy karikatúrákiállítást „A gép is ember” címen, ahová részben a hagyományos módszerekkel alkalmazó karikatúristák, részben a számítógéppel alkotó művészek pályaműveit várjuk.

Lesz ismét — talán jobb propagandával megszervezve — számítógépes hangverseny; ha sikerül, akkor komolyzenei és könnyűzenei is, neves művészek részvételével, valószínűleg nem a kiállítás területén, hanem valamelyik jó akusztikájú hangversenyteremben.

Ismét rendezünk családi versenyt, amelyel azt szeretnénk bemutatni, hogy a drága pénzén vásárolt otthoni számítógépeket a játékon kívül mennyi másra is lehet alkalmazni.

Az Országos Műszaki Múzeum ismét új kiállításra készül. Ezúttal nemcsak öreg számítástechnikai eszközöket hoz, de az adatvitelt, az információörögzítés történelmi emlékeit is láthatják a látogatók.

Írásom elején említtetem, hogy a μ'88 már nemcsak kiállítás, nemcsak találkozó, de vásár is. Tavaly is voltak üzletek, amelyek tulajdonosai főleg könyvekből, számítógépes médiumokból, de perifériákból és számítógépekből is jó forgalmat csináltak. Így akik kijötték a μ'88-ra, általában nem mentek haza üres kézzel, mert olyasmit is megvehettek, amit egyébként csak hosszas utánjárás után tudtak volna megkapni. Reméljük, az idén is lesz olyan vásárfia, ami szívesen vesznek meg a látogatók a μ'88 üzleteiben.

Egy folyóirat cikktérjedelmé véges. Nyilvánvaló, hogy a μ'88 minden eseményéről nem lehet írni; csak egy kis izeltőre vállalkoztam. Azt tanácsolom, hogy jöjjenek ki a μ'88-ra, ahol a rendezőbizottság nevében szíves örömmel várja a kedves olvasókat

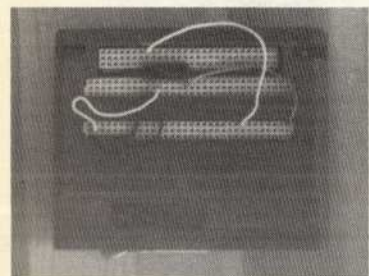
KOVÁCS GYÖZÖ

Másfél DINT és másfél DOUT egyszerre!

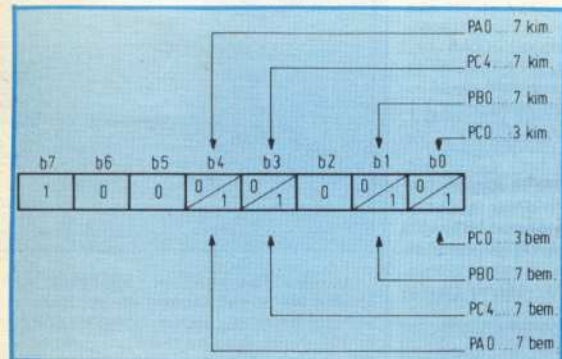
A Proto Board (PB) modul próbapanelnak is nevezhető, mert a tetején elhelyezett három csatlakozósávon különböző elektronikus áramkörök valósíthatók meg. Az alkatrészek (IC, LED, tranzisztor stb.) a szabványos méretű csatlakozókba dugaszolhatók, az összeköttetéseket pedig vékony, merev, szigetelt vezetékkel, ún. gyengeáramú szerelődhuzallal lehet kialakítani.

A modul fő előnye, hogy a megépített áramkör vagy a problémás alkatrész állapota — ha azok a TTL szinteknek megfelelnek — számítógép segítségével, programból vizsgálhatók. A programozhatóságot a csatlakozósávról kivezetett, háromszor 8 bites univerzális I/O felület biztosítja.

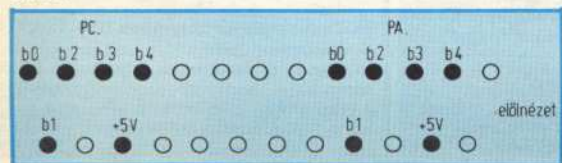
A csatlakozók kialakítását a modulról készült 1. kép, illetve a 3. ábra mutatja. Jól



1. kép



1. ábra
2. ábra



TechnoMIR

A próbapanel



2. kép

látható, hogy két sáv üres; itt célszerű az áramköri elemeket elhelyezni. A harmadik csatlakozósávról vannak kivezetve a ki- és bemeneti pontok (az I/O regiszterek pontjai), valamint a gerinchálózat néhány vonala.

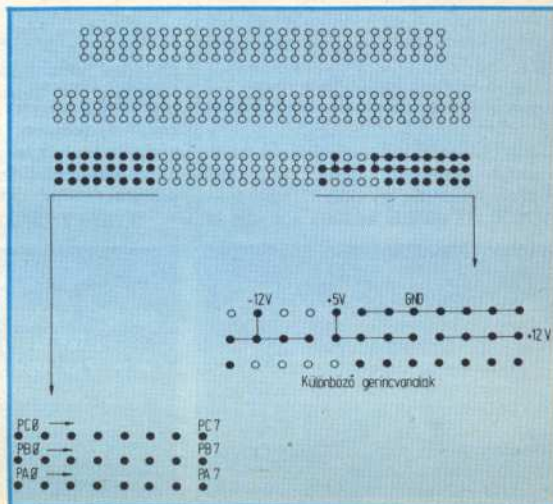
Felhívjuk a figyelmet arra, hogy a +12 V, -12 V tápfeszültséghez csak a POW (tápegység) modul alkalmazásával juthatunk, a számítógép csak a +5 V-os táplálást biztosítja. Külső tápegységet akkor is kell alkalmazni, ha a kialakított áramkör és

a PB modul áramfelvétele együttesen meghaladja a számítógép megengedett terhelhetőségét.

A PB modul lelke egy 8255 típusú IC. Ez az LSI áramkör az Intel MCS-80 családjába tartozik, és több különböző üzemmódban párhuzamos adatforgalmat képes lebonyolítani. A modulban a legegyszerűbb, ún. 0 üzemmódban dolgozik.

A mi szempontunkból a felépítését úgy képzeljük el, hogy három darab, egyenként 8 bites programozható regisztert (PA, PB és PC azaz Port A, Port B és Port C) tartalmaz. A programozhatóság itt azt jelenti, hogy beállítható, melyik regiszter legyen kimenet és melyik bemenet. Az utolsó, PC re-

3. ábra



giszter 4 bitenként programozható, tehát lehet olyan állapota, amikor az alsó 4 bit (PCB-PC3) például bemenet, de a felső 4 bit (PC4-PC7) kimenet.

Az egyes regiszterek állapotát hordozó információ tárolására egy kontrollregisztert helyeztek el az áramkörben; a PB modul használatát ennek feltöltésével kell kezdeni. A regiszter HT címe 43, a megfelelő utasítás tehát az OUT 43,X. Egyéb számítógépeknél a megfelelő cím irandó (lásd a Magazin 1987/12. számában!).

Az X szám, azaz a kontroll szó az 1. ábra alapján képezhető. A PA, PB és PC regiszterek HT címei rendre 40, 41 és 42. A kimenetek aktív 5 V-os kimenetek, a bemenetek nem tároló bemenetek.

Programozási példát nem hozunk, mert a PB nem más mint (beállítástól függően) egy DINT és egy DOUT együtt, amit pedig remélhetően mindenki kiválóan programoz már. A fényképen látható összeállítási példa egy 7400 logikai IC egyik NAND kapujának vizsgálatára alkalmas.

Egy illesztő interfész

A JOY (joystick-illesztő) modul két szabványos botkormány fogadására alkalmas. Elsősorban természetesen azoknál a számítógépeknél van jelentősége, amelyekbe nincs ilyen áramkör beépítve, tehát a HT, a Spectrum vagy a Primo esetében. Ez utóbbi tartalmaz ugyan botkormányillesztőt, de hatpólusú DIN aljzatához a legelterjedtebb típusok nem csatlakoztathatók.

A JOY modul segítségével ugyanazt a botkormányt használhatjuk, mint a Commodore gépekhez. Sőt, ha a modult például C64-hez kötjük, a csatlakoztatható botkormányok száma és így a számítógéppel közvetlenül kapcsolatot tartó személyek száma négy lehet.

Egy lehetséges összeállítás mutat a 2. kép. A csatlakozó bekötése a 2. ábrán látható.

A felhasználói program elején a modul kontrollregiszterébe 155-öt kell írni. A kontrollregiszter HT címe 47, tehát HT-n OUT 47,155 utasítás hatására a modul ké-

```

*****
# RAJZPROGRAM PRIMORA #
*****

10 CLS
# KERET MEORAJZOLASR #
20 FOR I=0 TO 255
30 SET(I,0) : SET(1,191)
40 NEMT I
50 FOR I=0 TO 191
60 SET(0,I) : SET(205,I)
70 NEMT I
75 X=128 : Y=96
# CONTROLL SZÓ BETÖLTÉSE #
80 OUT 152+47,155
# BILLENTŰZET VIZSGÁLATA #
90 A$=INKEY$
100 IF A$="" THEN 110 ELSE 10
*****
# JOY MODUL LEKÉRDEZÉSE #
*****

110 A=INP( 192+47 )
120 IF A AND 2 THEN 200
# JOYSTICK LEKÉRDEZÉS RAJZOLASR #
130 IF A#1 THEN Y=Y+1
140 IF A#4 THEN Y=Y-1
150 IF A#8 THEN X=X+1
160 IF A#16 THEN X=X-1
170 SET (X,Y) : GOTO 90
# JOYSTICK LEKÉRDEZÉSE TORLÉSRE #
200 IF A#3 THEN Y=Y+1
210 IF A#6 THEN Y=Y-1
220 IF A#10 THEN X=X-1
230 IF A#18 THEN X=X+1
240 RESET (X,Y) : GOTO 90
    
```

pes a botkormány jeleit fogadni és továbbítani a számítógépnek. Az információt a PA bemenet esetén a 44 HT címről nyerjük — INP(44) —, a PC bemenetet használva 46 a HT cím. Egy programon belül természetesen mindkettőt használható.

A megfelelő címről beolvasott adat a botkormány állapotát bitenként tartalmazza a következőképpen:

- 0 bit — fel
- 1. bit — tűz
- 2. bit — le
- 3. bit — balra
- 4. bit — jobbra

Tehát a 44, illetve 46 címről beolvasott 1; 2; 4; 8; 16 adat alapján eldönthető a botkormány állapota.

A közölt program Primóra készült egyszerű rajzprogram, elsősorban a modul használatának gyakorlásához. Minimális változtatás után érdemes HT gépen is kipróbálni.

KIRÁLY LÁSZLÓ—ALBU LÁSZLÓ

A HT-1080Z számítógép ROM programja egy sor, alapképzésben nem használt kulcsszót tartalmaz. Az alapépeben ezeknek a kulcsszavaknak ugrási címei az SN hibára mutatnak. Teljes kiépítésű csatlakoztatott floppyval például a TRSDOS NEWDOS 2.1 verziója jelentkez be. Ilyenkor élnék a *táblázataiban* felsorolt DOS kulcsszavak és rutinjai. (Vö. Sztróky: A Z80 Assembler. Műszaki Könyvkiadó, 1985. Felföldi—Lukács: A HT-1080Z iskolaszámítógép fix tárolója. Ipari Informatikai Központ, 1985.)

CVI	5E46	GET	627C
FN	55BE	PUT	627B
CVS	5E49	CLOSE	606F
DEF	5655	LOAD	5F7B
CVD	5E4C	MERGE	600B
EDF	61EB	NAME	6346
LDC	6231	KILL	63C0
LOF	6242	&	50B7
MKIB	5E2D	LSET	60E6
MKSS	5E30	RSET	60E5
MKDS	5E33	INSTR	582F
CMD	56C4	SAVE	6044
TIME\$	5714	LINE	5756
OPEN	6349	USR	5679
FIELD	60AB		

A végrehajtott rutinok címeiből látható, hogy a rendszer a memóriát is átszervezi.



Az iskolákba telepített, zömükben 16 kb-ot RAM-al üzemelő gépeknél ilyen kiépítés esetén alig marad szabad memóriaterület. Pedig de jó lenne a HT gépeknél is floppyt használni! A hajlékonylemez tárolónak a kazettás tárolóval szembeni előnyeit úgy vélem, nem kell ecsetelni.

Idestova egy éve használom a Real-Tem Gmk sorosbusz illesztőjét a HT-1080Z és Commodore perifériák összekapcsolására (lásd az *ábrát*). Az összeállításban a HT jelentéktelen korlátokkal ugyanígy képes kezelni a VC-1541-et, mint azt a C64 teszi. A bővítő csatlakoztatása után a HT értelmezi és lemez meghajtó egység (és/vagy

Minden hétfőn 17-től 19 óráig

a Mikroszámítógép Magazin munkatársai és felkért szakértők válaszolnak az olvasók kérdéseire a szerkesztőségben: Budapest II. Fő u. 68. I. em. 109. vagy a 154-090 és a 154-250-es telefonon.

Minden kedden 17-től 20 óráig ENTERPRISE klub

a VSZM Közösségi Házban (Bp. XI., Fehérvári út 120.) Klubvezető: Romvári Gábor Telefon: 450-950/473

Commodore floppyval

nyomatató) kezelésére felhasználja a LOAD, LOAD? (=VERIFY), SAVE, OPEN, CLOSE, CMD, PRINT*(=PRINT*), GET*, INPUT*, PUT* kulcsszavakat.

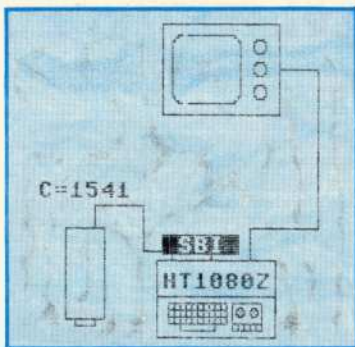
A HT ROM-ban lévő, TRSDOS részére fenntartott — és ezzel az összeállításnál nem értelmezett — kulcsszavak saját rutinok meghívására továbbra is rendelkezésre állnak.

Az illesztőegység a HT RAM-jából csak a 403EH-407FH területet és az I/O puffert használja fel. Az illesztést és az új kulcsszavak végrehajtását az interfész saját 2 kb-át rendszerprogramja biztosítja, mely párhuzamosan él a HT ROM-jával.

Mint ahogy a HT a VC—1541-et kezeli, a Commodore gépeknél alkalmazott lemezformátumot hozza létre, így ugyanúgy alkalmazhatók a parancssatornán a közvetlen elérésű üzemmod DOS parancsai is. Ez a lehetőség új távlatot nyit a HT programozásában és az adatkezelésben. Az itt közölt program lemezfigyelési térképet rajzol a képernyőre.

Az illesztő használatának nagy előnye, hogy nagymértékben csökkenti a HT iskolai oktatásban való alkalmazásának „rizikófaktorát”. A saját készítésű programokon kívül a TII oktatóprogram-jegyzékében közölt mintegy 300 program is az iskolák rendelkezésére áll. Az órákon való felhasználás fő megnehezítője, a kazettás tárolás miatt azonban e programokat viszonylag kevésbé használják.

A floppy alkalmazása az 500 bit/s átviteli sebességet 3200 bit/s-ra emeli; ugyanak-



kor tudjuk, hogy a programbeolvasás időszükségletét nemcsak a LOAD parancs tényleges végrehajtásából adódik. Így bizvást állítom, hogy a program betöltési ideje és megbízhatósága nagyságrenddel gyorsabb, jobb, mint korábban, a „kazettás világban” volt.

A BASIC programok minden további nélkül átmásolhatók kazettáról lemezre. A gépi kódú (játék) programok kazettáról lemezre töltésének több módja van. Az egyik megoldás a következő lépésekből áll:

1. A gépi kódú program kezdő-, vég- és indítási címének meghatározása
2. A gépi kódú program kazettáról való betöltése
3. Indítás helyett RESET vagy BREAK (öninduló programnál RESET)
4. A BASIC-terület programcímmutatóinak átírása a gépi kódú program megfelelő címire
5. BASIC programként a program lemeze vétele: SAVE „gépi”, 8

Az így kimentett program LOAD „gépi”, 8,1 parancssal betölthető és SYSTEM / KEZDŐCÍM begépelésével indítható.

(Az egyszerűbb, „nagyüzemi” megoldáshoz a gmk által gyártott, több gép összekapcsolását biztosító Minet interfészt is alkalmazni kell.)

A közkezen forgó gépi kódú HT játékok és felhasználói programok zöme ezzel a módszerrel lemeze vihető. Vannak azonban olyan kazettás gépi kódú programok is, amelyek használják a DOS memóriaterületét. Az ilyen programok betöltése a rendszer összeomláshoz vezet. A sorosbusz interfész használata során a kulcsszavak ugrási címeként (4173A—41E2H), a DOS munkaterületnek (403EH—407FH) felülírásán, valamint a SYSTEM/8 alkalmazásán kívül más módszerrel még nem sikerült kialakítani.

Nagy kár, hogy nem korábban és nem nagyobb darabszámban került az interfész alkalmazásba — de lehet, hogy ebben nem is ő a hibás.

PRIMO

Gyorsított rendezőprogram

A Mikroszámítógép Magazin 1987/8. számában az Integrált szoftver című cikkben olvastam, hogy hosszú szövegrészek mozgatása esetén néha úgy tűnik, mintha a számítógép „leagyott” volna.

Ezt a jelenséget tapasztaltam a Primónál is nagyobb méretű szöveges adathalmazok rendezése esetén. Ilyenkor hajlamosak vagyunk a feladat megoldását kizárólag a gépi kódú programban keresni. A BASIC nyelv gépi lehetőségeit felhasználva rendezőprogramunk gyorsítható, ha rendezés közben a szöveges adatokat nem mozgathatjuk. A program működéséhez tudni kell, hogy a Primo a következő módon tárolja a szöveges változókat:

```

10000 REM GYORSITOTT RENDEZO .....
10010 N=A
10011 AI=0 : AJ=0 : I=0 : J=0
10012 AO=0 : AI=0 : A2=0
10160 DIM A$(4)
10170 A$(1)="FAL"
10180 A$(2)="ASZTAL"
10190 A$(3)="TV"
10200 A$(4)="PAPIR"
10410 REM RENDEZES.....
10420 FOR I=1 TO N
10422 FOR J=1 TO N : PRINT N-I
10430 IF A$(J)>A$(I) THEN 10470
10440 AJ=VARPTR(A$(J))
10441 AI=VARPTR(A$(I))
10442 AO=PEEK(AJ)
10443 AI=PEEK(AI+1)
10444 A2=PEEK(AJ+2)
10447 POKEAJ,PEEK(AI)
10448 POKEAJ+1,PEEK(AI+1)
10449 POKEAJ+2,PEEK(AI+2)
10450 POKEAI,AO
10451 POKEAI+1,AI
10452 POKEAI+2,A2
10470 NEXT J
10480 NEXT I
10490 REM KIRATAS.....
10495 CLS
10500 FOR I=1 TO N
10510 PRINT A$(I)
10520 NEXT I
    
```

1. a változó VARPTR függvénnyel visszakereshető címén a változó hosszát karakterekben,
2. a következő és a következő utáni címen azt a címet, ahol a szöveget ténylegesen tárolja.

A rendezés elvégzéséhez megírunk egy „normál” rendezőprogramot, de a szöveges adatok cseréjét nem végezzük el, hanem a két kicserélendő változó tárolási helyét megkeressük a VARPTR függvénnyel, és az ezeken a címeken és az utánuk következő két címen lévő értékeket egymással felcseréljük. A mellékelt program segítségével ellenőrizhetjük a leírtakat.

SOMOGYI GYÖRGY

```

10 CLEAR1000:DEFINT A-Y
20 CLS:PRINTTAB(5) "LEMEZ FOGLALTSÁG"
30 FORI=10T045STEP2:SETC(16,I):NEXT
40 FORI=11T036STEP2:SETC(1,9):NEXT
50 FORI=9T09:SETC(1,27):NEXT
60 FORI=10T026:FORI=32T035:SETC(I,1):NEXT:NEXT
70 FORI=27T039:FORI=36T035:SETC(I,1):NEXT:NEXT
80 FORI=34T039:FORI=29T035:SETC(I,1):NEXT:NEXT
90 FORI=40T044:FORI=28T035:SETC(I,1):NEXT:NEXT
100 PRINT@320,"SAV:";PRINT@340,"SZEKTOR:";
110 OPEN15:8:15:"10"OPEN2:8:2:"#
120 PRINT#15,"B-R";2:0:18:0
130 PRINT#15,"B-P";2:144:N#=""
140 FORI=1T020:GET#2:J#
150 IF J#=CHR$(13) OR J#=CHR$(160) THEN J#=""
160 IF J#="" THEN#10T010
170 N#=#+J#;NEXT:CLOSE2:CLOSE15
180 PRINT#64,"NEV:ID:";J#;
190 OPEN15:8:15:"10"OPEN2:8:2:"#
200 PRINT#15,"B-R";2:0:18:0
210 S=0:FORA=10T46STEP4:PRINT#15,"B-P";2:R
220 PRINT# 324:R/4;
230 FORI=1T04:GET#2:J# IF J#="" THEN#3=CHR$(8)
240 N#=#+J#;IF I=1ANDR/4<24#S#4;
250 IFC1 THEN#10S#2#0
260 NEXT:NEXT:CLOSE2:CLOSE15
270 PRINT@128,"SZABD:";J#;"BLOK:";
280 IF INKEY#="" THEN#280 ELSE END
290 F#=#R/4:V#=#+1#0:0:128
300 Z#=#/4:V#=#+1:FC1 THEN#0S#3#0
310 C#=# IF R/4<18#NDV#31 THENC#-21
320 IF R/4<17#NDV#4<24#NDV#29 THENC#-29
330 IF R/4<17#NDV#4<24#NDV#29 THENC#-28
340 IF R/4<30#NDV#26 THENC#-27
350 PRINT@406,C-10;
360 N#=#-INT(C/2):O#=#/2:1FC1 THEN#300
370 RETURN
380 IFC0INT(V,F)=0 THENSET(V,F):RETURN
390 RESET(C+1,F):O#=#:RETURN
    
```

SZ. LUKÁCS JÁNOS

Mire használjuk és mire ne használjuk a számítógépet?

Az NJSZT diplomaterv-pályázatán első díjat nyert dolgozat egyik fejezete a számítógépek középiskolai alkalmazási lehetőségeivel foglalkozik. A pályamunkának ezt a részét közöljük, némi rövidítéssel.

Számítások végzése

Jól felhasználható a számítógép mérések kiértékelésénél, ahol a grafikus megjelenítés szemléletesebb teszi a megoldást. Szerepe lehet bonyolult képletrendszer kiszámításában vagy olyan feladat elvégzésénél, amely ha egyszerű is, de sokszor kell ismételnél (például a statisztikus fizikában, egyenletek, egyenletrendszerek megoldásánál). Nem szabad alkalmazni a számítógépet, ha a számítás menete az érdekes, vagy akkor, ha egy zsebszámológép is el tudja végezni a számítás — azaz ne löjünk verébre ágyúval.

Oktatási segédprogramok

Olyan feladatok megoldását segíthetik, ahol — már vagy még — nem a tevékenység elvégzése a lényeges. Ide tartoznak a versek ritmikai és hangtani elemzését segítő programok, a billentyűzetről lejátszott zenéhez kottarajzolást végző programok, a függvényábrázolást, függvénytranszformációt végző programok stb. Mindezek hasznos eszközök ugyan, de önmagukban nem indokolják a számítógépek széles körű alkalmazását az iskolában.

rendszerben oldhatók meg, ahol lehetőség van nagy mennyiségű adat tárolására és visszakeresésére.

A programozott oktatás számítógépes segítése

E programok pedagógiai haszna erősen megkérdőjelezhető, ráadásul a jelenlegi számítógépes kiépítéssel mellett még egy könyv lehetőségeit sem érik el. Magyarországon nem gyakoriak az ilyen programok.

Logikai játékok

Hasznosak lehetnek nemcsak a számítógéppel való ismerkedés idején, de a programozás tanulásakor is. Értelmes játékprogramok írása a logikus gondolkodást is fejleszti.

Számítógépes szimuláció

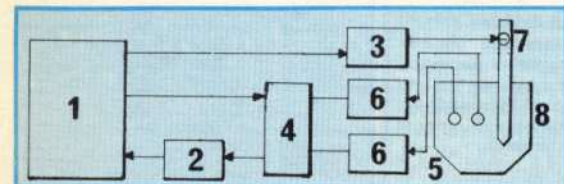
A számítógép nem helyettesítheti a kísérletet. Akkor jól szerephez, ha a kísérletet valamilyen okból akadályba ütközik, mert a jelenség túlsá-

— számítástechnikai szakkörön készítsük el a számítógépes modellt és a programot,
— használjuk a számítógépes modellt, keressünk vele érdekes paramétereket, vizsgáljuk meg a kapott eredményeket, próbáljuk kitalálni az összefüggéseket,
— ha lehet, kísérletezzünk valós rendszerrel és használjuk fel a szimulációs eredményeket,
— ha kísérletünkön, vessük össze a kísérlettel és a szimulációval kapott eredményeket,
— értelmezzük a kapott eredményeket,
— ha szükséges, ennek alapján módosítsuk a modellt és a számítógépes modellt.

A szimuláció témakörében a számítógépes feladatok száma határátalan, így nagyon sokféle program készülhet. Ezért vált a szimuláció a számítógépek felhasználásának egyik legfontosabb területévé.

Számítógépes mérés és vezérlés

A számítógépek használatának ugyancsak fontos területe a számítógépes mérés és vezérlés. Az igazán magas szintű mérések azok, amelyeket számítógép vezérel. Ezekben a számítógép kü-



1. ábra

Tesztek végzése és kiértékelése

Ez a számítógépek legsematikusabb felhasználási területe. A jó programok is alig nyújtanak többet, mint a hagyományos tesztek, de a legtöbb esetben még ezt sem teszik, csupán a szöveg megjelenítését és az értékelést végzik. Sokszor maguk a tesztkérdések is sok kívánnivalót hagynak maguk után. (Számomra az is kétséges, hogy helyese-e az iskolai számonkérésben tesztek alkalmazni, különösen akkor, ha figyelembe vesszük, hogy megoldások eredményessége a matematikai képességekkel korrelál.)

Egyéni gyakoroltatás

Elvileg lehetséges ilyen jellegű felhasználás is, például idegen szavak vagy idegen nyelvű ragozás gyakorlására. Ebben az alkalmazásban azonban különösen sok buktatója van egy program elkészítésének, és egy rossz program — például a válaszok túl merev értelmezésével — többet árt, mint használ. Elképzeltető azonban olyan alkalmazás is, hogy a pedagógus személyre szabottan készít kisebb-nagyobb programokat ismétlésre, otthoni tanulásra, olyan gyerekek számára, akik versenyre készülnek vagy hosszabb ideig betegednek, és ezért szükséges, hogy egyénileg gyakoroljanak, megfigyeljenek bizonyos törvényszerűségeket, vagy éppen számoljanak.

Információtároló és -visszakereső programok

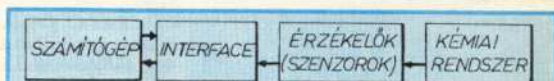
Ilyen például egy iskolai könyvtár könyvváltozatát kezelő vagy egy növényhatározó létrehozó program. Az iskolai alkalmazás kerékkötője, ha nincs megfelelő háttérrel; ezek a feladatok ugyanis csak lemezegységet tartalmazó

gosan gyors (az atomban lejátszódo folyamatok), lassú (csillagászati, geológiai jelenségek), drága (atomreaktor helyes működéséhez szükséges paraméterek beállítás), veszélyes (mérgező anyagok, láncreakció esete), bonyolult (az akk) — a kísérlet túl sokszor kell elvégezni (kockadobás), vagy etikai akadályba ütközik (emberrel végzett kísérlet) stb.

A számítógépnek nemcsak hiánypótló szerepe lehet, hanem intelligens kísérletvezetőként is megállja a helyét. Segítségével gyorsan és hatékonyan megválaszthatók az elvégzendő kísérlet paraméterei. A szimulációs modell elkészítésére kétféle lehetőség adódik. Amikor elemi folyamatokat modellezünk, a kezdeti paraméterek megadása után a jelenség szimulálása az események véletlenszerű kiválasztásával történik. Ekkor a számítógépes modell igen közel áll a természethez. A másik lehetőség, hogy a szimuláció a természeti törvények már felállított egyenletei alapján modellezi a jelenséget. Az ilyen megoldás akkor hasznos, ha nagyon sok folyamat együttes hatását kell vizsgálnunk, amelynek áttekinthetése és a kezdeti paramétereitől függő eredmény megadása nagyon nehéz lenne. Ilyen például egy atomreaktor folyamatának modellezése, ami azért is kínázkodik az egyetlen lehetőségnek, mert egy valódi reaktor paramétereinek változtatása igen veszélyes.

A szimulációs megismerési folyamatnak iskolai alkalmazás során a következő lépési képzeltetők el:

- foglalkozzunk a valós rendszerrel, ismerjük meg a törvényeit, jelenségeit,
- beszéljük meg a vizsgálni kívánt jelenséget; ha lehet, figyeljük meg,
- szakköri foglalkozáson készítsük el a modelljét,



2. ábra

lőnböző eszközök segítségével, a program alapján beavatkozik a mérés menetébe, és például valamihhez hozzáönt egy reagenst, vagy elzár egy csapot.

Egy számítógéppel vezérelt kísérlet rendszerfelépítését mutatja az 1. ábra, amelynek értelmezése a következők. A folyamatok által szolgáltatott jelek általában nem digitálisak, hanem rendszerint analógok, ezért a számítógép (1) és a külvilág közvetlen csatlakozása egy analóg-digitál (2), illetve egy digitál-analóg (3) átalakítást kíván. A több mérőhelyről érkező jelek egymás utáni befogadását a multiplexer (4) teszi lehetővé. A kémiai és fizikai adatok felfogására és a szükséges elektromos jellel alakítására érzékelők kellenek (5), amelyek jele először rendszerint egy jelfeldolgozó (6) mérőműszerbe jut (például Wheatstone-híd). A mérés vezérlése a megfelelő kibocsátó eszközök (búrtacsaps, termoszát stb.) működtetésével történik (7). Természetesen a főként hagyományos laboratóriumi eszközökből álló reaktióter (8) jelenléte is szükséges.

Ilyen típusú mérést ma még nem végezhetünk az iskolában, mert a beavatkozáshoz szükséges eszközök nagyon drágák. Ezért be kell érni azokkal, hogy a beavatkozás a kísérletelő végzi, a számítógép pedig csak mér és figyelmeztet; jelzi, mi a következő teendőnk. Az iskolai alkalmazást szem előtt tartva talán nem is túl jó, ha egy mérés teljesen automatizált. Nagyon probléma, hogy az ilyen, „félíg automatizált” méréshez szükséges érzékelők, illesztők sem állnak korlátlanul rendelkezésünkre. A 2. ábra a számítógéppel csatlakoztatott mérés felépítésének vázlatát tartalmazza, amely tájékoztat arról, hogy milyen egy-egyek kellenek egy ilyen kísérlethez.

Az Országos Pedagógiai Intézet számára 1985-ben készült felmérés szerint az iskolák 69 százalékában van az ilyen mérésekhez megfelelő számítógép, de már csak alig 10–15 százalékuk tudja biztosítani a legfontosabb érzékelőket és mérőműszereket is. Az illesztők hiányán enyhít a TechnoMIR-készlet megjelenése, de a probléma végleges megoldása itt is várat magára.

GÖRBICS LÍVIA

Programcsomagok 660S módra

Optimizátor

A program azoknak nyújt segítséget akik szeretnék minél jobban kihasználni a mágneslemez kapacitását. Igyekszik úgy összeválogatni a tárolandó programokat (méreteik alapján), hogy a lehető legkevesebb szabad hely (blocks free) maradjon a lemezen, a túréshatár 'szem előtt' tartásával. Az üres szektorok száma ezen értéken belül lesz. Tehát ha TURESHATAR-nak 8-t adunk, a programot arra kényszerítjük, hogy a rendelkezésre álló blokkméretekkel addig sokkoczon vagyis válogasson közülük, míg összegüként 664-et nem kap. A futási időre való tekintettel, ezt rögtön kiírja, majd csököntevő a készletet tovább kéréséig.

A programlista PRINT utasításában szereplő kibetűs rövidítések:

sc = SHIFT+CLR HOME bill.
ion = CTRL+RUS ON bill.
iof = CTRL+RUS OFF bill.

Örökélet POKE-ok

BLACK WYCHE	POKE 4248,44
COMMANDO	POKE 2409,44
CRACKERS REVENGE	POKE 32170,44
FIRE ANT	POKE 16696,44
JET SET WILLY	POKE 14711,234 POKE 14712,234
MUTANT MONTY	POKE 19231,44
ROCKET ROGER	POKE 21233,238 POKE 21237,69 POKE 21238,7
SON OF BLAGGER	SVS 5600 POKE 6370,234 SVS 3052
SPELUNKER	LOAD "UUU" 8,1 POKE 19125,234 POKE 19126,234 SVS 3838
THUNDERBIRDS	POKE 17449,44 POKE 25404,44
TROLLIE WALLIE	RESET utón: POKE 7711,44 SVS 51712

```

100 PRINT "sc"; PRINT; PRINT TAB(12);"ion DISK OPTIMIZER ion"
102 PRINT; PRINT SPC(7);"FILE-OK LEMEZEN TÁROLÁST"
104 PRINT SPC(9);"OPTIMALIZÁLO PROGRAM."
106 PRINT; PRINT
110 INPUT "PROGRAMOK SZÁMA";A
120 IF A<2 OR A>1000 THEN 110
130 DIM I(A), T(I), U(A): A=A-1
150 FOR I=1 TO A
154 : PRINT
156 : FOR J=1 TO Q
160 : PRINT J;"BLOKKMERET ";
162 : C=PEEK(211): PRINT T(J);
164 : POKE 215,C-1: INPUT E
166 : IF E<1 OR E>64 THEN 160
170 : T(J)=E: TT(J)=E: B=0
180 : NEXT J: PRINT
184 : IF I=1 THEN PRINT "ELLEHŐZÉS!"
190 NEXT I
192 INPUT "DENTER VAGH";U; PRINT
194 IF U#="" THEN 192
196 IF U#="I" OR U#="IGEN" THEN OPEN 4,4: Q=1
200 INPUT "TÜRESHATAR";E; PRINT
210 IF E<0 OR E>64 THEN 200
220 REM-----BACKTRACK ALGORITMUS-----
230 COSUB 400: S=0: P=1: 2=0
240 FOR I=1 TO A
250 : S=S+T(I)
260 : IF S>64 THEN S=S-T(I): COTO 290
270 : 2=2+1: U(2)=I
280 : IF 64-S <= E THEN COSUB 500: COTO 230
290 NEXT I
300 S=S-T(U(2)): P=U(2)+1: 2=2-1
310 IF 2=0 THEN 600
320 IF P=0 THEN 240
330 COTO 300
400 REM-----SZELEKTIV DENEZES-----
410 FOR I=1 TO A-1
420 : FOR K=I+1 TO A
430 : IF T(K) > T(I) THEN M=T(I): T(I)=T(K): T(K)=M
440 : NEXT K
450 NEXT I
460 RETURN
500 REM-----EREDMENY KIÍRÁS-----
505 IF Q THEN PRINT#4,S;"BLOKK";
510 PRINT S;"BLOKK";
520 FOR J=2 TO 1 STEP -1
525 : IF Q THEN PRINT#4,T(U(J));
530 : PRINT T(U(J)); : T(U(J))=T(A): A=A-1
540 NEXT J: PRINT; IF Q THEN PRINT#4
550 IF A > 1 THEN RETURN
600 REM-----
605 IF A#0 THEN PRINT "ion NINCS MEGOLDÁS!"
610 IF Q THEN PRINT#4, " MARADEK.";
615 PRINT; PRINT " MARADEK.";
620 FOR I=1 TO A
625 : IF Q THEN PRINT#4,T(I);
630 : PRINT T(I);
640 NEXT I: PRINT; PRINT
645 IF Q THEN PRINT#4: PRINT#4
650 INPUT "MECFELEL";U; PRINT
660 IF U#="I" OR U#="IGEN" THEN CLOSE#4: END
670 A=A-1
680 FOR I=1 TO A: T(I)=TT(I): NEXT I: COTO 200
    
```



Folyt.
Köv.

Készítette:
HÁMÁN KATÓ
Közgazdasági Szki
MHSZ klubja

Bázisjáték HT-re

A játékban űrhajónk meteoritokkal csatározik. Úgy kell masinánkat irányítani, hogy az össze ne űtközzön a kóbor hulló csillagokkal — ezek egyébként a képernyőn 0-val jelzettek —, ugyanakkor minél több ellenséges objektumot (X) semmisítsen meg. Az űrhajó a talaj felett repdes, s közben változik az attól mért magassága. A játék izgalmát fokozandó, a képernyő balra tolódik.

A figurákat a fel- és lefelé mutató nyilakkal mozgatjuk, bombáznai pedig a szóközzel tudunk. A gép a képernyőeltolását l, a sikeres találatért 30 pontot „fizet”.

Ha az űrhajó meteoritba űtközik vagy a talajba fúródik, illetve ha elhagyja a képernyőt, akkor vége a játéknak. Cél, hogy minél hosszabb idő alatt, minél több pontot gyűjtsünk össze.

BACZÓ TAMÁS

```

3 REM # READY? 29999 #
10 CLS:RANDOM:CLEAR 500
20 DEFINT X,G,Z,I-N
30 X=462:G=14400:Z=16159
40 B$=" "+CHR$(152)+CHR$(191)+CHR$(164)
50 C$=STRING$(32,191)
55 F=15361
60 POKE 32763,4:Z1=INT(Z/256)
70 POKE 32761,Z1:POKE 32760,Z-256*Z1
80 POKE 16526,48:POKE 16527,117
82 IF PEEK(32767)=10,READ F#,F#,F#,F#:GOTO 200
83 POKE 32767,10
85 PRINT"*KAPCSOLJ AT FEL KEPERNYORE !"
90 FORI=1 TO 4:READ A#
100 FORJ=1 TO LEN(A#) STEP 2
110 D#=MID$(A#,J,1):E#=MID$(A#,J+1,1)
120 IFASC(D#)>64,D=(ASC(D#)-55) ELSE D=ASC(D#)-48
130 IFASC(E#)>64,E=(ASC(E#)-55) ELSE E=ASC(E#)-48
140 POKE 30000+H,D*16+E:H=H+1
150 NEXT J,I:CLS:GOTO1000
200 FORI=0 TO 13:READ D
210 OUT 31,I:OUT 30,0:NEXT
220 CLS:PRINT@0,STRING$(32,131);:PRINT@X,B#;:E=7
230 FORI=768 TO 960 STEP 64:PRINT@I,C#;:NEXT
299 REM *** FOPROGRAM ***
300 IFPEEK(G)=8,GOSUB1200:GOTO 325
310 IFPEEK(G)=16,GOSUB1300:GOTO 325
320 IFPEEK(G)=128 ANDM=0,GOSUB1500:GOTO 350
325 IFM=1,GOSUB1510 ELSE FORI=1 TO20:NEXT
350 IFL/3=INT(L/3),D=RND(9):PRINT@64+32;"0";
360 Y=USR(0):L=L+1:IFPEEK(15364+X)<32,2000
370 PRINT@X,B#;:GOTO 300
    
```

Megszakításvezérelt manómozgatás C64-re

Manókat használó játékprogramjainknál — és különösen az ablaktechnikát alkalmazó felhasználói programoknál — sokszor szükség lehet arra, hogy a képernyőn egy manót, esetleg egy kurzort a program futásától függetlenül botkormányán mozgatassunk. Ezt a feladatot oldja meg az alábbi program úgy, hogy lehetővé teszi a nullás manók mozgatását az egész képernyőn. Négy bájttal is meghatározza, hogy a monitor mely részén mozoghassunk a manóval. A bájtcímek: \$CF24, \$CF46, \$CF31, \$CF68, \$033C. Az első kettő a bal felső, a második kettő a jobb alsó sarkát adja meg a manó mozgására rendelkezésre álló ablaknak. A programban ezek úgy vannak beállítva, hogy a manó bal felső sarka mindig a látható képernyőterületen marad. A programcimen állandóan 00-t tart, és ennek mértékét csak azokban a pillanatokban változtatja meg FF-re, amikor lenyomjuk a botkormány tűzgombját.

OLÁH GERGELY

CF00 78	SEI	CF38 B0 1F	BCS \$CF59
CF01 AD 14 03	LDA \$0314	CF3A AA	TAX
CF04 8D 8A CF	STA \$CF8A	CF3B AD 10 D0	LDA \$D010
CF07 A9 19	LDA #\$19	CF3E 6A	ROR
CF09 8D 14 03	STA \$0314	CF3F 6A	TXA
CF0C AD 15 03	LDA \$0315	CF40 E0 07	BCS \$CF49
CF0F 8D 8B CF	STA \$CF8B	CF42 A2 00 D0	LDX \$D000
CF12 A9 CF	LDA #\$CF	CF45 E0 18	CPX #\$18
CF14 8D 15 03	STA \$0315	CF47 F0 10	BEQ \$CF59
CF17 58	CLI	CF49 CE 00 D0	DEC \$D000
CF18 60	RTS	CF4C D0 0B	BNE \$CF59
CF19 EA	NOP	CF4E AA	TAX
CF1A AD 00 DC	LDA \$DC00	CF4F AD 10 D0	LDA \$D010
CF1D 4A	LSR	CF52 4A	LSR
CF1E B0 0A	BCS \$CF2A	CF53 18	CLC
CF20 AE 01 D0	LDX \$D001	CF54 2A	ROL
CF23 E0 32	CPX #\$32	CF55 8D 10 D0	STA \$D010
CF25 F0 03	BEQ \$CF2A	CF58 8A	TXA
CF27 CE 01 D0	DEC \$D001	CF59 4A	LSR
CF2A 4A	LSR	CF5A B0 1F	BCS \$CF7B
CF2B B0 0A	BCS \$CF37	CF5C AA	TAX
CF2D AE 01 D0	LDX \$D001	CF5D AD 10 D0	LDA \$D010
CF30 E0 F9	CPX #\$F9	CF60 6A	ROR
CF32 F0 03	BEQ \$CF37	CF61 8A	TXA
CF34 EE 01 D0	INC \$D001	CF62 90 07	BCC \$CF6B
CF37 4A	LSR	CF64 AE 00 D0	LDX \$D000
		CF67 E0 57	CPX #\$57
		CF69 F0 10	BEQ \$CF7B
		CF6B EE 00 D0	INC \$D000
		CF6E D0 0B	BNE \$CF7B

C64

Óra a keretben

Az alábbi program egy mindig leolvasható, hat számjegyű órát varázsol a C64 alsó keretének jobb szélére. Az óra folyamatosan jár, látható az órák, percek, másodpercek értéke. Ez olyan BASIC-programokkal egybeépítve lehet hasznos, ahol szükség van a pontos idő állandó kijelzésére, miközben a teljes képernyő használható marad. Hátránya, hogy a lemezegységet ki kell iktatni, mint mindig, amikor sprite-ok vannak a képernyőn bekapcsolva. A hat számjegyet ugyanis két sprite, a 7. és 8. jeleníti meg. A kijelzett idő a CIA 1. chip belső órája, tehát indításkor ezt kell beállítani. A program eleje mutatja, hogyan lehet ezt BASIC-programmal megtenni.

A program észreveszi a hibásan beírt adatsorokat, és kiírja a hiba helyének sor-számhatárait. Az idő kijelzése közben a BASIC-utasítások futási ideje csak mintegy 3 százalékkal lesz hosszabb.

GYURKOVITS GERGELY

```

1 REM
20 REM
30 REM IRTA: GYURKOVITS GERGELY 1987.
40 REM
50 C=52735:FOR J=1TO4:S=0:FOR I=1TO80
60 C=C+1:READA#:H=ASC(A#)-48:H#H+(H#9)*7
70 L=ASC(RIGHT$(A#,1))-48:L=L+(L#9)*7
80 H#H#16+L:S#S#H:POKEK,H:NEXT I:READ L
90 IF S=L THEN NEXT J:C=56329:GOTO 110
100 GOTO 900
110 POKE C+6,PEEK(C+6) AND 127
120 PRINT"KEREA: A PONTOS IDOT "
130 PRINT"(ORA,PERC,SEC) FORMABAN I"
140 INPUT Q,P,S
150 IF Q/23 OR P/59 OR S/59 THEN 120
160 IF Q<0 OR P<0 OR S<0 THEN 120
170 IF Q/11 THEN Q=0 OR 128
180 POKE C+2,Q+INT(Q/10)*6
190 POKE C+1,P+INT(P/10)*6
200 POKE C+0,S+INT(S/10)*6
210 PRINT"NYOMJ (RETURN)-T, HA A BEIRT"
220 PRINT"IDO EGYZEIK A VALOSAGI I"
230 POKE 53029,0:REM MINDENT KIIRJON
240 INPUT:SYS 52736:INPUTI:REM START
250 POKE 53029,ASC("GY")
260 REM NEM IR MINDENT KI.
270 END
300 DATA 78,A9,3E,BD,14,03,A9,CE
310 DATA 8D,15,03,A9,7F,8D,0D,DC
320 DATA AD,0E,DC,0F,80,8D,0E,DC
330 DATA A2,0E,8E,FE,07,EB,8E,FF
340 DATA 07,A2,14,BD,2A,CF,9D,09
350 DATA 0D,CA,D0,F7,8E,2D,0D,8E
360 DATA 2E,D0,8E,08,DC,8E,FF,3F
370 DATA CA,8E,12,D0,58,60,AD,19
380 DATA D0,4A,90,5C,AD,12,D0,C9
390 DATA F6,90,48,AD,11,D0,AB,29
595 DATA 10270
600 DATA F7,8D,11,D0,AD,21,D0,85
610 DATA 02,CE,66,CE,98,AD,06,88
620 DATA D0,FD,AE,20,D0,AD,0C,88
630 DATA BE,21,D0,30,09,AC,12,D0
640 DATA C0,FC,BD,11,90,F7,AD,19
650 DATA BC,66,CE,FO,08,48,20,A3
660 DATA CE,AD,08,DC,68,8D,11,D0
670 DATA A9,32,8D,12,D0,EE,19,D0
680 DATA 4C,7E,EA,A5,02,8D,21,D0
690 DATA A9,F9,BD,12,D0,EE,19,D0
695 DATA 10702
700 DATA 4C,31,EA,AD,0F,DC,29,7F
710 DATA 8D,0F,DC,AD,09,DC,A2,CF
720 DATA 20,EA,CE,A2,D0,20,F1,CE
730 DATA 20,22,CF,AD,0A,BC,A2,90
740 DATA 20,EA,CE,A2,CE,20,F1,CE
750 DATA 20,22,CF,AD,08,DC,10,0A
760 DATA 29,7F,C9,12,F0,0A,FB,69
770 DATA 12,08,C9,12,D0,02,A9,00
780 DATA A2,8E,20,EA,CE,A2,8F,4C
790 DATA F1,CE,8D,20,CF,4A,4A,4A
795 DATA 10435
800 DATA 4A,8E,12,CF,29,0F,0A,0A
810 DATA 0A,18,69,80,BD,0F,CF,A9
820 DATA D0,69,01,8D,10,CF,A6,01
830 DATA A9,33,85,01,A0,07,B9,AB
340 DATA D1,99,C0,03,CE,12 F,CE
850 DATA 12,CF,8B,10,F1,86,01,A9
860 DATA 55,60,F0,06,A9,47,F0,02
870 DATA 68,68,60,77,FF,18,03,30
880 DATA 03,C0,1B,C0,46,3D,C0,C8
890 DATA 00,15,79,F1,00,00,00,00
895 DATA 8133
900 PRINT"HIRBA A"400+100*J"--495+100*J
910 PRINT"SORSZAMU DATA BLOKKBAN I":END
920 REM 39540 A TFLJES OSSZE
    
```



**D-Subminiatur csatlakozók,
kábelek, analóg-digitál átalakítók,
egyéb számítástechnikai berendezések
és kellek nagy választékával várja Önöket**

**a MAGÉV-RAINBOW
Elektronikai Szaküzlet**

Budapest VI., Rudas László u. 33.
Telefon: 122-392. Telex: 22-6323



**Számítástechnikai és Szolgáltató
Kiszövetkezet**

Központ: Budapest, Szilágyi Erzsébet fasor 1. 1026
Levél cím: 1378 Budapest 64., Postafiók 31. Telefon: 352-558

BASIC és gépi kód

Legutóbb a beszúrásokat és azok egyik legerjedtebb válfaját, a BASIC és KERNAL vektorainak átírását ismertettem. Most megnézzük, hogyan tudunk a meglévő kivül további USR függvényeket definiálni és használni. Előtte — a témával összefüggésben — a tokenekről ejtek néhány szót.

A tokenek

A BASIC interpreter az utasítások begépelése és a RETURN billentyű lenyomása után az utasításokat tokenizálja. Ez azt jelenti, hogy az utasításban előforduló kulcsszavakat (utasítás- és függvényneveket, műveleti jeleket) azok egybájtos kódjával, tokenjével helyettesíti. Ezáltal a tárolt program mérete és végrehajtási ideje rövidebb lesz, de az utasítások feldolgozása is egyszerűbbé válik.

```
033c a9 51      lda #51
033e a0 03      ld  #503
0340 8d 08 03  sta $0308
0343 8c 09 03  sty $0309
0346 a9 9e      lda #59e
0348 a0 03      ld  #503
034a 8d 0a 03  sta $030a
034d 8c 0b 03  sty $030b
0350 60         rts
```

1. lista

A tokenek belső kódja \$80-nál nem kisebb érték, így egy tokenizált BASIC-sor elemzésekor előjelvizsgálattal felismerhetjük, hogy tokenről van-e szó. Természetesen egyéb körülményeket is figyelembe kell venni, melyekre most nem térek ki.

A mostani gépi kódú rutinokban három token valamelyikének meglétét vizsgáljuk: a DEF (\$96), az USR (\$B7) és az egyenlőségjel (\$B2) tokenjeit.

A programról

A listákban látható program C64-re készült, és a bevezetőben vázolt feladat megoldásának gondolatmenetét mutatja be.

Az 1. listán a BASIC vektorokat átíró rutin található. A 2. listán van a DEF USR utasítást feldolgozó rutin. Az IBM PC BA-

SIC-jétől eltérően definiálni tudjuk az eredeti, sorszám nélküli USR függvényt is, tehát összesen 11 különböző felhasználói függvény alkalmazására van lehetőségünk. A 3. listán az USR függvényeket kiértékelő rutin látható.

A DEF USR-t feldolgozó rutin működésének vázolata a következő. A rutin megvizsgálja, hogy az utasítás a DEF tokenjével kezdődik-e. Ha nem, a feldolgozás az eredeti interpreter rutinban folytatódik. DEF esetén megvizsgálja, hogy az USR tokenje következik-e. Ha nem, akkor az interpreternek a DEF FN utasítást feldolgozó rutinjába ugrik. Ellenkező esetben számjegy és egyenlőségjel vagy azonnali egyenlőségjel következhet, minden másra SYNTAX ERROR hibaüzenetet kapunk. Ha nincs hiba, akkor az egyenlőségjel jobb oldalán levő aritmetikai kifejezés kiértékelése és a címtáblázatba, illetve — sorszám nélküli USR esetén — az USR vektorba történő lerakása, majd az interpreter feldolgozó ciklusába való visszaugrás következik.

2. lista

```
0351 20 73 00  jsr $0073
0354 c9 96      cmp #596
0356 f0 06      beq $035e
0358 20 79 00  jsr $0079
035b 4c e7 a7  jmp $a7e7
035e 20 73 00  jsr $0073
0361 c9 b7      cmp #5b7
0363 f0 03      beq $0368
0365 4c b3 b3  jmp $b3b3
0368 85 02      sta $02
036a 20 73 00  jsr $0073
036d f0 68      beq $03d7
036f b0 08      bcs $0379
0371 29 0f      and #50f
0373 0a         asl
0374 85 02      sta $02
0376 20 73 00  jsr $0073
0379 c9 b2      cmp #5b2
037b d0 5a      bne $03d7
037d 20 73 00  jsr $0073
0380 20 8a ad  jsr $ad8a
0383 20 f7 b7  jsr $b7f7
0386 a6 02      ldx $02
0388 30 0a      bmi $0394
038a 9d db 03  sta $03db, x
038d 98         tya
038e 9d da 03  sta $03da, x
0391 4c ea a7  jmp $a7ea
0394 8d 12 03  sta $0312
0397 98         tya
0398 8d 11 03  sta $0311
039b 4c ea a7  jmp $a7ea
```

```
039e a9 00      lda #500
03a0 85 0d      sta $0d
03a2 20 73 00  jsr $0073
03a5 c9 b7      cmp #5b7
03a7 d0 0f      bne $03b8
03a9 20 73 00  jsr $0073
03ac f0 29      beq $03d7
03ae 90 0e      bcc $03be
03b0 a5 7a      lda $7a
03b2 d0 02      bne $03b6
03b4 c6 7b      dec $7b
03b6 c6 7a      dec $7a
03b8 20 79 00  jsr $0079
03bb 4c 8d ae  jmp $ae8d
03be 29 0f      and #50f
03c0 0a         asl
03c1 48         pha
03c2 20 73 00  jsr $0073
03c5 20 f1 ae  jsr $0073
03c8 68         pla
03ca a8         tay
03ca b9 da 03  lda $03da, y
03cd 85 55      sta $55
03cf b9 db 03  lda $03db, y
03d2 85 56      sta $56
03d4 4c 0e af  jmp $afe0
03d7 4c 08 af  jmp $af08
```

3. lista

A címtáblázat egy húszbájtos terület, amely a tíz számított USR függvény belépési pontjának címét tartalmazza, természetesen csak definiálás után. Indulásnál — az USR vektorhoz hasonlóan — az ILLEGAL QUANTITY ERROR üzenetet kiíró ROM rutin címével (\$B248) kell feltölteni. A példaprogramban a táblázat közvetlenül a program után van elhelyezve, és nincs feltöltve az említett rutin címével. Ha valaki a mostani lista alapján vállalkozik a program kipróbálására, vegye ezt figyelembe.

A 3. listán látható kiértékelő rutin működési elve hasonló az előbbiéhez. Először megvizsgálja, hogy a feldolgozás alatti aritmetikai vagy karakterlánc-kifejezés következő eleme az USR tokenje-e. Ha nem, akkor a feldolgozás az eredeti ROM rutinon folytatódik. Ugyanez történik — némi módosítással —, ha az USR token nem számjegy követi. Sorszámos USR függvény esetén a feldolgozás a függvény kiértékelésével folytatódik.

A fenti leírás csak vázlatosan ismerteti a rutin működését. A részletes magyarázattal és a BASIC betöltőprogramok listájával legközelebb jelentkezem.

BARNA LÁSZLÓ

Z80 programok haladóknak Spectrumra és Primóra

9. Manó szerkesztő program II.

```

1 ;-----;
2 ; ;
3 ; SPR1EDEF 32*32 ;
4 ; UHI-software ;
5 ; 1987.07.11 ;
6 ; ;
7 ;-----;
8 ; ORG $E800 ;
9 ; JP SPRDEF ;
10 ;-----;
11 SCR EQU $4000
12 SCBUF EQU $1000
13 BUFEND EQU $E7FF
14 SCLEN EQU $1800
15 SCREND EQU $57FF
16 ADD EQU $484A
17 AHADD EQU $0A
18 BADD EQU $4010
19 A00 EQU $4050
20 B00 EQU $3380
21 000 EQU $0F00
22 INK EQU 16
23 PAP EQU 17
24 AT EQU 22
25 CR EQU 13
26 EOL EQU $00
27 PIXADD EQU $22AC
28 EMIT EQU $0010
29 UBG EQU $5C7B
30 BSP EQU $0F00
31 SIZ EQU 32
32
33 TXT DEFS 2
34 FIFO EQU $-1
35 DEFS $0*3
36 FIFDE EQU $-1
37 READER DEFS 2
38 WRITER DEFS 2
39 SPRAUD DEFS 2
40 FILEN DEFS 2
41 SP1 DEFS 1
42 SPE DEFS 1
43 SPOS DEFS 2
44 POS DEFS 2
45 CDS DEFS 2
46 INBUF DEFS 6
47 SBUFF DEFS 128
48 PATTS DEFS 64
49 ;-----;
50 CLEAR CALL SAVESP
51 CLEAR1 CALL MSG
52 DEF B PAP,7,INK,0
53 DEF B 19,1,EOL
54 LD BC,$0A0A
55 CLEAR2 CALL ATBC
56 CALL MSG
57 DEF B "
58 DEF B EOL
59 INC B
60 LD A,B

```

```

61 CP 14
62 JR NZ,CLEAR2
63 LD BC,$0010
64 CLEAR3 CALL ATBC
65 LD A,6
66 CALL EMIT
67 INC B
68 LD A,B
69 CP 16
70 JR NZ,CLEAR3
71 CALL MSG
72 DEF B 19,0,EOL
73 RET
74 ;-----;
75 PLOT LD BC,(POS)
76 CALL BCOURD
77 LD D,4
78 PLOT1 LD E,4
79 PLOT2 CALL PLOTBC
80 INC C
81 DEC E
82 JR NZ,PLOT2
83 LD A,C
84 SUB 4
85 LD C,A
86 DEC B
87 DEC D
88 JR NZ,PLOT1
89 LD BC,(POS)
90 CALL ACOURD
91 CALL PLOTBC
92 RET
93 ;-----;
94 BREAK EI
95 LD (IY+8),#38
96 LD (IY+84),#00
97 LD (IY+14),#38
98 RSJ B
99 DEF B $FF
100 ;-----;
101 LEFT LD A,(POS)
102 DEC A
103 AND $1F
104 LD (POS),A
105 RET
106 ;-----;
107 RIGHT LD A,(POS)
108 INC A
109 AND $1F
110 LD (POS),A
111 RET
112 ;-----;
113 DOWN LD A,(POS+1)
114 DEC A
115 AND $1F
116 LD (POS+1),A
117 RET
118 ;-----;
119 UP LD A,(POS+1)
120 INC A

```

```

121 AND $1F
122 LD (POS+1),A
123 RET
124 ;-----;
125 SIGN LD HL,(POS)
126 LD (SPOS),HL
127 RET
128 ;-----;
129 NEW LD C,3
130 CALL INP
131 DEF B "To wich "
132 DEF B "sprite? "
133 DEF B EOL
134 CALL SPRND
135 RET C
136 CALL NEWG1
137 CALL LISTSP
138 RET
139 ;-----;
140 BACK LD A,(SP1)
141 OR A
142 RET Z
143 DEC A
144 JR SPRN1
145 ;-----;
146 FORW LD A,(SP1)
147 INC A
148 LD B,A
149 LD A,(SPE)
150 SUB 7
151 CP B
152 RET C
153 LD A,B
154 JR SPRN1
155 ;-----;
156 ROT LD HL,ADDU
157 LD (ADDR1),HL
158 INC HL
159 INC HL
160 INC HL
161 LD (ADDR4),HL
162 LD B,SIZ-1
163 ROT1 CALL HLDOWN
164 DJNZ ROT1
165 LD (ADDR3),HL
166 DEC HL
167 DEC HL
168 DEC HL
169 LD (ADDR2),HL
170 LD A,$80
171 LD (MASK1),A
172 LD (MASK2),A
173 RLCA
174 LD (MASK3),A
175 LD (MASK4),A
176 LD C,16
177 ROT2 LD B,16
178 ROT3 PUSH BC
179 LD HL,ADDR1
180 CALL EXIT

```

Ez alkalommal befejezem a manószerszert ismertetését és egyben a sorozatot is. A listán közlöm a program még hiányzó részét és a teljes változótáblát. Ügyelni kell arra, hogy semmi ne szerepeljen kétszer. A teljes forráslista tehát a következő részekből áll:

9. rész: parancsrutinok

8. rész: programtörzs, néhány alaprutin

6-7. rész: I/O és egyéb alaprutinok

5. rész: mintás feltöltés (FILL)

4. rész: zárt alakzat festése az 5. részbeli változtatásokkal

1. rész: gyors LDIR, ahogyan a 4. részben szerepel is.

A kiszolgáló (parancs-) rutinok működése

A kiszolgáló rutinok közül a legbonyolultabb a ROT, amely 90 fokkal elforgatja a

manót. Működésére nem térek ki, ráérő olvasók megfejtethetik. Érdemes azonban megfigyelni, milyen gyorsan működik. Azért ilyen hosszú és bonyolult.

Tanácsok a fordításhoz, főleg Spectrumosoknak

A forrásszöveg meglehetősen hosszú, de ha elég alacsony címre töltjük, befér a

```

181 CALL EXBIT
182 CALL EXBIT
183 CALL EXBIT
184 LD HL,ADDR1
185 CALL EXBIT
186 CALL NEXT
187 POP BC
188 DJNZ ROT3
189 CALL CORR
190 DEC C
191 JR NZ,ROT2
192 CALL ATOB
193 RET
194 EXBIT LD E,(HL)
195 INC HL
196 LD D,(HL)
197 INC HL
198 LD B,(HL)
199 INC HL
200 LD A,(DE)
201 PUSH AF
202 AND B
203 EX AF,AF'
204 POP AF
205 JR Z,EXB0
206 OR B
207 JR EXBE
208 EXB0 OK B
209 XOR B
210 EXBE LD (DE),A
211 EX AF,AF'
212 RET
213 NEXT LD A,(MASK1)
214 RRC A
215 LD (MASK1),A
216 JR NC,NEXT2
217 LD HL,(ADDR1)
218 INC HL
219 LD (ADDR1),HL
220 NEXT2 LD HL,(ADDR2)
221 CALL HLUP
222 LD (ADDR2),HL
223 NEXT3 LD A,(MASK3)
224 RRC A
225 LD (MASK3),A
226 JR NC,NEXT4
227 LD HL,(ADDR3)
228 DEC HL
229 LD (ADDR3),HL
230 NEXT4 LD HL,(ADDR4)
231 CALL HLDOWN
232 LD (ADDR4),HL
233 RET
234 CORR LD HL,(ADDR1)
235 DEC HL
236 DEC HL
237 CALL HLDOWN
238 LD (ADDR1),HL
239 LD HL,(ADDR2)
240 LD B,16

```

```

241 CORR21 CALL HLDOWN
242 DJNZ CORR21
243 LD (ADDR2),HL
244 LD A,(MASK2)
245 RRC A
246 LD (MASK2),A
247 JR NC,CORR3
248 INC HL
249 LD (ADDR2),HL
250 CORR3 LD HL,(ADDR3)
251 INC HL
252 INC HL
253 CALL HLUP
254 LD (ADDR3),HL
255 LD HL,(ADDR4)
256 LD B,SIZ/2
257 CORR41 CALL HLUP
258 DJNZ CORR41
259 LD (ADDR4),HL
260 LD A,(MASK4)
261 RRC A
262 LD (MASK4),A
263 RET NC
264 DEC HL
265 LD (ADDR4),HL
266 RET
267 ADDR1 DEFS 2
268 MASK1 DEFS 1
269 ADDR2 DEFS 2
270 MASK2 DEFS 1
271 ADDR3 DEFS 2
272 MASK3 DEFS 1
273 ADDR4 DEFS 2
274 MASK4 DEFS 1
275 ;-----;
276 TURN LD HL,ADDR
277 LD B,SIZ
278 TURN1 LD D,H
279 LD E,L
280 INC DE
281 INC DE
282 INC DE
283 CALL TURNB
284 DEC DE
285 INC HL
286 CALL TURNB
287 DEC HL
288 CALL HLDOWN
289 DJNZ TURN1
290 JP ATOB
291 TURNB PUSH BC
292 LD B,B
293 LD C,(HL)
294 LD A,(DE)
295 TURN2 RR (HL)
296 RLA
297 RR C
298 DJNZ TURN2
299 LD (HL),C
300 LD (DE),A

```

```

301 POP BC
302 RET
303 ;-----;
304 INV LD HL,ADDR
305 LD C,SIZ
306 INV1 LD B,SIZ/8
307 INV2 LD A,(HL)
308 CPL
309 LD (HL),A
310 INC HL
311 DJNZ INV2
312 DEC HL
313 DEC HL
314 DEC HL
315 DEC HL
316 CALL HLDOWN
317 DEC C
318 JR NZ,INV1
319 JP ATOB
320 ;-----;
321 ROT LD HL,(SP05)
322 LD (POS),HL
323 RET
324 ;-----;
325 DEL LD HL,ADDR
326 LD DE,SBUFF
327 LD B,$20
328 DEL1 CALL EXX1
329 CALL EXX1
330 CALL EXX1
331 CALL EXX1
332 DEC HL
333 DEC HL
334 DEC HL
335 DEC HL
336 CALL HLDOWN
337 DJNZ DEL1
338 JP ATOB
339 EXX1 LD A,(DE)
340 LD C,(HL)
341 EX DE,HL
342 LD (DE),A
343 LD (HL),C
344 EX DE,HL
345 INC DE
346 INC HL
347 RET
348 ;-----;
349 OLB CALL SAVESP
350 LD C,3
351 CALL INP
352 DEFB "lich old"
353 DEFB "sprite?"
354 DEFB EOL
355 CALL SPNO
356 RET C
357 LD BC,$4F50
358 EX DE,HL
359 CALL SPRITE
360 JP ATOB

```

GENS3 mellé, és le tudja fordítani. Természetesen az egészét megjegyzések nélkül kell begépelni. Az assembler célszerű betöltési címe 23800. Fordításkor a „table space?” kérdésre 2000-et kell válaszolni.

A másik megoldás, hogy kazettáról fordítunk, de ez elég lassú, úgyhogy nem javasolom.

A program használata

A manószerkesztő üzenetei angol nyelvűek, egyrészt mivel a programot elsősorban programozók használják, akik ismerik a „compu-angolt”, másrészt így nem kellett ékezetes betűkkel szerencsétlenkedni.

szövegben a szükséges karaktereket úgy, hogy kényelmesebb legyen.

Ha kész vagyunk egy manóval, nyomjuk meg az ENTER-t. Ekkor megkérdezi, hogy a manóalakfájl hányadik elemére tegye az új manót, majd odamásolja. Szükség esetén új automata listát is kapunk (lásd fent: SPRNO rutin), hogy mindjárt megjelenjen az új manó a szemünk előtt is az „O” ablakban.

A D (törlés) billentyű második lenyomása a törlés előtti állapotot hozza vissza.

Mintás feltöltéshez új mintázatot a következőképpen lehet csinálni. Mivel a minták 8×8 bitesek (csak az ismétlődés miatt látszanak nagyobbak a mintalistában), a szerkesztőmező bal felső sarkába rajzoljunk meg egy 8×8-as mintát. Hogy a mező többi részén mi van, az mindegy. Ha kész, nyomjuk meg a P (pattern = mintázat) gombot. Erre megkérdezi, hogy melyikre tegye (0...7). Közben kiteszi az „O” ablakba a régiéket, hogy lássuk, melyiket töröljük.

A forrásszöveg nem tartalmaz kész mintákat, úgyhogy azokat mind meg kell szerkeszteni. Célszerű „teli” mintát is csinálni, gyakran kell. Ha ezután felvesszük a programot, a minták „vele mennek”.

Ha a 2. részben bemutatott manókezelőhöz manót szerkesztünk, akkor mindegyikhez külön takarási maszkra van szükség. Ezeket úgy kell megrajzolni, mint közönséges manókat. Erre a következő módszer javasolom. Tegyük fel, hogy manónknak zárkörvonala van. Először a manó látható alakját rajzoljuk meg, és rögzítjük a manóalakfájlban, majd a rajta kívül maradt felületet kenjük be teljesen a megfelelő „teli” mintával. Ha a manó hozzáért a 32×32-es keret széléhez, akkor ezt a műveletet a kívül maradt felület minden elemére végzük el. A kész maszok a látható alak elé kell tenni. A 256-tal osztható kezdőcíme nem kell ügyelni, csak akkor, amikor használjuk.

Primósoknak

Jó hír, hogy semmit nem kell változtatni, mivel elég magas szintű rutinoknál tartunk: a gépfüggetlő alaprutinokon már túl vagyunk.

Lehet változtatni a PICT képernyőrajzoló rutinon: ki lehet irtani belőle a színvezérlő karaktereket. Sajnos a színek hiánya miatt az ablakok nem fognak olyan szépen elkülönülni, mint a Spectrumon.

UHERKOVICH PÉTER

```

361 F-----
362 FILLH CALL LISTPT
363 CALL SAVESP
364 CALL GRIDD
365 CALL OPENIN
366 CALL MSG
367 DEFN "Choose a "
368 DEFN "pattern.."
369 DEFN EOL
370 CALL GETNO
371 CALL PATIND
372 RET C
373 LD BC,(POS)
374 CALL ACORRD
375 EX DE,HL
376 CALL FILL
377 CALL AIOB
378 CALL LISTSP
379 RET
380 GRIDD LD HL,ARDD
381 CALL HLUP
382 DEC HL
383 LD C,SIZ*2
384 GRIDD1 PUSH HL
385 LD B,SIZ/D*2
386 GRIDD2 LD (HL),DEFI
387 INC HL
388 DJRZ GRIDD2
389 POP HL
390 CALL HLDOWN
391 DEC C
392 JR NZ,GRIDD1
393 CALL DEL
394 CALL SAVESP
395 RET
396 F-----
397 PATT CALL LISTPT
398 CALL OPENIN
399 CALL MSG
400 DEFN "To wich "
401 DEFN "pattern?"
402 DEFN EOL
403 CALL GETNO
404 CALL PATIND
405 RET C
406 LD DE,ARDD
407 EX DE,HL
408 LD B,B
409 PATT1 LD A,(HL)
410 LD (DE),A
411 INC DE
412 CALL HLDOWN
413 DJRZ PATT1
414 CALL LISTPT
415 CALL KEYIN
416 CALL LISTSP
417 CALL EXEC
418 RET
419 F-----
420 HELP CALL MSG

```

```

421 DEFN 19,1,PAP,7,EOL
422 LD HL,HELPH
423 LD B,0
424 HELP1 LD C,16
425 CALL ATBC
426 HELP2 LD A,(HL)
427 CALL EMIT
428 INC HL
429 DEC C
430 JR NZ,HELPH
431 INC B
432 LD A,B
433 CP 16
434 JR NZ,HELPH
435 CALL PAUSE0
436 CALL ATDB
437 RET
438 HELPH DEFN "56/8 cursor move"
439 DEFN "0 plot 5 sign"
440 DEFN "V clear B back"
441 DEFN "R rotate N form"
442 DEFN "P pattern"
443 DEFN "I inverse L line"
444 DEFN "D delete F fill"
445 DEFN "T return to sign"
446 DEFN "D delete last"
447 DEFN "comand"
448 DEFN "M mirror"
449 DEFN "SPACE to basic"
450 DEFN "ENTER if ready"
451 DEFN "
452 DEFN "To loud or save"
453 DEFN "return to basic"
454 F-----

```

A program elindításakor megkérdezi a manóalakfájl kezdőcímét. Erre azért van szükség, mert a programot elvben bárhova fordíthatjuk, és hozzá képest bárhol elhelyezhetjük a memóriában a manóalakfájlt. Nem ellenőrizi, úgyhogy ügyeljünk az elhelyezésre. Ezután megkérdezi, hány manót szeretnénk szerkeszteni. Az így megadott manóalakfájl címét és hosszát ki is írja a képernyőre.

A kész fájl kimentése úgy történhet, hogy a SPACE billentyűvel kilépjünk BASIC-be és BASIC utasítással kimentjük. A cím és a hossz segítségével a képernyőn látható.

Az egyes szerkesztőbillentyűk az előző rész 1. táblázatában láthatók. Ehhez azonban néhány kiegészítést kell fűzni. A kurzormozgató billentyűk elhelyezkedése azért éppen olyan amilyen, mert Spectrumon így a Cursor Joystick használható erre a célra. Akinek ilyen nincs, vagy Primo-tulajdonos, az átirhatja a TABLE1-ben és a HELPM

Közületek, figyelem!
Mikroszámítógépet
akarnak vásárolni?
Tájékozzanak
a naprakész piaci
helyzetéről!
Díjtalan ismertető!
MESZ
Számítástechnika
1365 Budapest, Pf. 193.

Számítógépes grafika Pascalban

SÍK TRANSZFORMÁCIÓK

Az előzőekben már láthattunk példát arra, hogyan lehet a képernyő egyes „ablakait”, vagyis karakterhelyekkel vagy képpontokkal határolt téglalap alakú területeit a képernyő egy másik részére átmásolni, 90°-kal elforgatni, tükrözni stb. Ezeket a műveleteket összefoglaló néven transzformációknak hívják.

Sokkal általánosabban használható és főleg sokkal gyorsabban működő eljárásokat fejleszthetünk ki, ha a transzformációkat nem a képernyőn megjelenített raszterpontokon végezzük el közvetlenül, hanem előbb elkészítjük az ábrázolandó kép geometriai modelljét, majd ennek fontosabb pontjain hajtjuk végre a transzformációt, és ezt rajzolatjuk ki a számítógéppel.

A geometriai modell legyen egyszerű és mégis eléggé rugalmas ahhoz, hogy tetszőleges vonalas rajzolatokat tudjunk vele leírni. Ehhez pontokat (a mintaprogramban legfeljebb 50 pontot), valamint ezeket a pontokat a megadás sorrendjében összekötő egyenes szakaszok nyílt vagy záródó sorozatát használunk.

A pontok koordinátáinak tárolására a COORD azonosítójú rekordtípust deklaráltuk a főprogramban, melynek mezőlistája egy X és egy Y integert tartalmaz. A pontok koordinátáinak eredeti értékeit és transzformáltjait is a POINTS azonosítójú rekordtípusban fogjuk tárolni.

Ezek után térjünk rá a transzformációkra. Ha egy geometriai alakzatot önmagával párhuzamosan el akarunk tolni valamelyik koordinátatengely (például x) irányában, mondjuk Tx értékkel, nem kell mást tennünk, csak minden pontjának x koordinátájához Tx-et hozzáadunk. Természetesen egyidejűleg y irányban is végezhetünk eltolást vagy translációt; ilyenkor a tényleges elmozdulás valamilyen ferde irányban fog bekövetkezni. Ha a transzformált koordi-

nátákat vesszővel jelöljük, a transzformáció során általában ezt a két műveletet kell végrehajtani minden pontra:

$$x' = x + Tx$$

$$y' = y + Ty$$

A másik elemi transzformációs művelet a geometriai alakzat elforgatása a koordináta-rendszer kezdőpontja körül. E sorozat egyik korábbi cikkében megmutattuk, hogy ha α szöggel akarjuk a pontot elforgatni, akkor az új koordináták így számíthatók:

$$x' = x \cdot \cos \alpha - y \cdot \sin \alpha$$

$$y' = x \cdot \sin \alpha + y \cdot \cos \alpha$$

A forogtatást idegen szóval rotációnak is nevezzük.

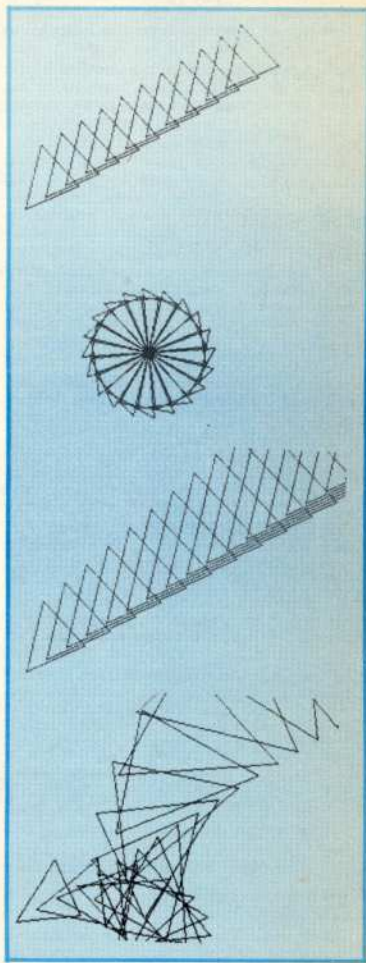
Végül a harmadik elemi transzformáció a léptékezés, mellyel nagyítani, kicsinyíteni lehet. A léptékezés transzformációs egyenletei:

$$x' = x \cdot Sx$$

$$y' = y \cdot Sy$$

Ezzel igen sok érdekes transzformáció hajtható végre. Ha például $Sx = Sy = 2$, akkor az eredeti ábránkat kétszeresére nagyítjuk. Ha $Sx = Sy = 0,5$, a képet felére kicsinyítjük. $Sx = 1$, $Sy = 2$ csak függőleges irányban nyújt, míg $Sx = 1$, $Sy = -1$ az y tengelyre tükröz stb.

Ha az olvasó gondolatban kipróbálja, milyen műveletek valósíthatók meg áltál, hogy az elemi transzformációk paramétereinek adunk más-más értéket, a lehetséges változatok egész sorát kapja. Van mégis néhány feladat, amely egy transzformációval nem valósítható meg. Ha például a képer-



```

10 ($L-)
20 (:TRANSZ filenev alatt kimentett INCLUDE rutinok)
30
40 PROCEDURE LineSet (N.Address:INTEGER);
50 (Egyszeres szakaszokból álló nyílt lanc)
60 VAR I,CIM,X0,Y0:INTEGER;
70 XY :COORD;
80 BEGIN
90 XY:=PEEK (Address,COORD);
100 WITH XY DO
110 BEGIN
120 X0:=X;
130 Y0:=Y
140 END;
150 CIM:=Address+4;
160 FOR I:=1 TO (N-1) DO
170 BEGIN
180 XY:=PEEK (CIM,COORD);
190 WITH XY DO
200 BEGIN
210 PixelLine (X0,Y0,X,Y);
220 X0:=X;
230 Y0:=Y

```

```

240 END;
250 CIM:=CIM+4
260 END
270 END; (LineSet)
280
290 PROCEDURE Polygon (N.Address:INTEGER);
300 (Sokszög rajzolása)
310 VAR X0,Y0:INTEGER;
320 XY :COORD;
330 BEGIN
340 LineSet (N.Address);
350 XY:=PEEK (Address,COORD);
360 WITH XY DO
370 BEGIN
380 X0:=X;
390 Y0:=Y
400 END;
410 XY:=PEEK (Address+4*(N-1),COORD);
420 WITH XY DO
430 PixelLine (X0,Y0,X,Y)
440 END; (Polygon)
450
460 PROCEDURE Zero (VAR M:Matrix);

```

```

470 (3*3 matrix nullazasa)
480 VAR I, J: INTEGER;
490 BEGIN
500   FOR J:=1 TO 3 DO
510     FOR I:=1 TO 3 DO
520       M[I, J]:=0
530     END; {Zero}
540
550 PROCEDURE Translation (VAR T: Matrix; Tx, Ty: REAL);
560 {Eltolasi matrix kepzeze}
570 VAR I: INTEGER;
580 BEGIN
590   Zero(T);
600   FOR I:=1 TO 3 DO
610     T[I, I]:=1;
620     T[3, 1]:=Tx;
630     T[3, 2]:=Ty
640   END; {Translation}
650
660 PROCEDURE Rotation (VAR T: Matrix; Alfa: REAL);
670 {Forgato matrix kepzeze}
680 VAR sina, cosa: REAL;
690 BEGIN
700   Zero(T);
710   sina:=SIN(Alfa);
720   cosa:=COS(Alfa);
730   T[1, 1]:=cosa;
740   T[2, 2]:=cosa;
750   T[3, 3]:=1;
760   T[2, 1]:=sina;
770   T[1, 2]:=-sina
780 END; {Rotation}
790
800 PROCEDURE Scaling (VAR T: Matrix; Sx, Sy: REAL);
810 {Nagyito-kicsinyito matrix kepzeze}
820 BEGIN
830   Zero(T);
840   T[1, 1]:=Sx;
850   T[2, 2]:=Sy;
860   T[3, 3]:=1
870 END; {Scaling}
880
890 PROCEDURE MatrixMult (VAR M1, M2, M3: Matrix);
900 {3*3 matrixok szorzasa}
910 VAR I, J, k: INTEGER;
920   s: REAL;
930 BEGIN
940   FOR I:=1 TO 3 DO
950     FOR J:=1 TO 3 DO
960       BEGIN
970         s:=0;
980         FOR k:=1 TO 3 DO
990           s:=s+M1[I, k]*M2[k, J];
1000        M3[I, J]:=s
1010       END;
1020     M1:=M3
1030   END; {MatrixMult}
1040
1050 PROCEDURE VectorMult (VAR V1, V2: Vector; VAR M: Matrix);
1060 {3*3 matrix es sorvektor szorzasa}
1070 VAR I, J: INTEGER;
1080   s: REAL;
1090 BEGIN
1100   FOR I:=1 TO 3 DO
1110     BEGIN
1120       s:=0;
1130       FOR J:=1 TO 3 DO
1140         s:=s+V1[I]*M[J, I];
1150       V2[I]:=s
1160     END;
1170     V1:=V2
1180   END; {VectorMult}
1190
1200 PROCEDURE Trans (n, C1mtol, C1mre: INTEGER);
1210 {n pont transzformacioja}
1220 VAR i, c1, c2: INTEGER;
1230   Pontxy: COORD;
1240 BEGIN
1250   c1:=C1mtol;
1260   c2:=C1mre;
1270   FOR I:=1 TO n DO
1280     BEGIN
1290       Pontxy:=PEEK(c1.COORD);
1300       WITH Pontxy DO
1310         BEGIN
1320           v[C1]:=X;
1330           v[C2]:=Y;
1340           v[C3]:=1;

```

```

1350     VectorMult(v1, v2, m1);
1360     X:=ROUND(v[C1]);
1370     Y:=ROUND(v[C2]);
1380   END;
1390   POKE(c2, Pontxy);
1400   c1:=c1+4;
1410   c2:=c2+4
1420   END
1430 END; {Trans}

```

1. lista

2. lista

```

10 CL-3
10 PROGRAM S.k.Transformacio
30 {1987. februar 22.}
40 TYPE COORD = RECORD
50   X, Y: INTEGER
60   END;
70   Matrix = ARRAY [1..3, 1..3] OF REAL;
80   Vector = ARRAY [1..3] OF REAL;
90 VAR POINTS: ARRAY [1..1003] OF COORD;
100 K, I, J, n: INTEGER;
110 v1, v2: Vector;
120 m1,
130 m2, m3: Matrix;
140
150 {BL-}
160 {F I: LINE }
170
180 {BL-}
190 {F I: TRANSZ}
200
210 BEGIN {Foprogram}
220 PAGE;
230 WRITE ('A pontok szama:');
240 READLN; READ(n);
250 WRITELN;
260 FOR I:=1 TO n DO
270   WITH POINTS[I] DO
280     BEGIN
290       WRITELN;
300       WRITE ('X: ', I, CHR(8), ' ');
310       READLN; READ(X);
320       WRITE ('Y: ', I, CHR(8), ' ');
330       READLN; READ(Y)
340     END;
350 PAGE;
360 LineSet (n, ADDR(POINTS));
370 REPEAT UNTIL INCH<>CHR(0);
380 PAGE;
390 Polygon (n, ADDR(POINTS));
400 FOR I:=1 TO 10 DO
410   BEGIN
420     Translation (m1, I*15, I*10);
430     Trans (n, ADDR(POINTS), ADDR(POINTS[n+1]));
440     Polygon (n, ADDR(POINTS[n+1]));
450   END;
460 REPEAT UNTIL INCH<>CHR(0);
470 PAGE;
480 FOR I:=1 TO 20 DO
490   BEGIN
500     Rotation (m1, I*3.1415926/10);
510     Translation (m2, 100, 70);
520     MatrixMult (m1, m2, m3);
530     Trans (n, ADDR(POINTS), ADDR(POINTS[n+1]));
540     Polygon (n, ADDR(POINTS[n+1]));
550   END;
560 REPEAT UNTIL INCH<>CHR(0);
570 PAGE;
580 FOR I:=1 TO 20 DO
590   BEGIN
600     Scaling (m1, I*0.1, I*0.1);
610     Translation (m2, I*15, I*10);
620     MatrixMult (m1, m2, m3);
630     Trans (n, ADDR(POINTS), ADDR(POINTS[n+1]));
640     Polygon (n, ADDR(POINTS[n+1]));
650   END;
660 REPEAT UNTIL INCH<>CHR(0);
670 PAGE;
680 FOR I:=1 TO 20 DO
690   BEGIN
700     Scaling (m1, I*0.1, I*0.1);
710     Rotation (m2, I*3.1415926/10);
720     MatrixMult (m1, m2, m3);
730     Translation (m2, I*15, I*10);
740     MatrixMult (m1, m2, m3);
750     Trans (n, ADDR(POINTS), ADDR(POINTS[n+1]));
760     Polygon (n, ADDR(POINTS[n+1]));
770   END;
780 REPEAT UNTIL INCH<>CHR(0)
790 END.

```

nyő középre felrajzolunk egy kört és ebből nagyitással szeretnénk egy másik, az előzővel koncentrikus kört kapni, akkor egy egyszerű léptékezés nem vezetne célhoz, mert a kör középpontja nem esik a koordinátarendszer kezdőpontjába, így a léptékezés a középpont origótól való távolságát is kinagyítaná: a két kör nem maradna koncentrikus. Ha az eredeti feladatot akarjuk megvalósítani, a következő lépéseket kell végrehajtani:

— El kell tolni a kör középpontját az origóba (vagyis az $x=0, y=0$ pontba, azaz a képernyő bal alsó sarkába).

— Az új koordinátákat másodszer is transzformálni kell: most jöhetne a megfelelő nagyítás.

— Végül egy harmadik transzformációval (transzlációval) vissza kell tolnunk a kör középpontját az eredeti helyére.

A transzformációk algoritmusait mátrixok segítségével számítógépekre alkalmazható módon is megfogalmazhatjuk. Ha az olvasó esetleg idegenkedik ettől a középiskolában nem tanított matematikai eszköztől, átugorhatja a következőket; elég, ha a később ismertetendő Pascal-eljárások alkalmazásának szabályait tanulmányozza azt a program ismertetésekor.

Nos, a transzformációkhoz egy pont koordinátáit háromelemű sorvektorba gyűjtjük. Ezzel az összes transzformáció az alábbi mátrixvektor-sorzással írható le:

$$[x' \ y' \ 1] = [x \ y \ 1] \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix}$$

ahol a jobb oldali mátrix az ún. transzformációs mátrix, melynek itt betűkkel jelölt elemei attól függnek, milyen transzformációról van szó.

A mátrix alakja eltolás esetén:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix}$$

ADOK

Seikosha GP 50 S printer Sinclair családhozvaló eladó. Kristóf Attila, Zalaegerszeg, Dózsa György u. 2. 8900.

Comodore VC20 felhasználói és jatek-programok eladása. Kérjen tájékoztatást! Juhász György, Salgótarján, 3100. Pf.: 157.

Primo monitor minden memóriakiépítésű gépre! Gépi kódú program kimentés-betűtés, nyomkövetés, disassembler és sok más hasznos funkció. Ára leírással együtt, kazettán 170,- Ft, utánvétellel. A64-es gépre programokat cserélek. Ajánlatom: Assembler, Pascal, Lisp, FORTH, C rendszerek. Paller Gábor, Budapest, Varga Gyula András park 8/a. 1149.

Alig használt, jó állapotban lévő

forogatásnál:

$$\begin{bmatrix} \cos\alpha - \sin\alpha & 0 \\ \sin\alpha & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

léptékezésnél:

$$\begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ha két mátrixot összeszorozunk, az eredmény ugyancsak egy mátrix, melynek anynyi sora lesz, amennyi a bal oldali mátrixnak és annyi oszlopa, amennyi a jobb oldali mátrixnak van. A szorzatmátrix elemeit a következő egyszerű séma szerint lehet kiszámítani. Ha a jobb oldali mátrix alá írjuk a kiszámítandó szorzatvektort, akkor egyes elemeit úgy kell meghatározni, hogy a keresett elemmel egy sorban levő jobb oldali mátrix elemeit jobbról balra haladva megszorozzuk a bal oldali mátrix azonos oszlopába eső elemekkel felülről lefelé haladva, és a szorzatokat összegezzük.

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{bmatrix} \begin{bmatrix} 3 & 2 \\ -2 & -1 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 14 & 10 & 3 \\ 10 & 6 & 3 \\ 14 & 10 & 3 \end{bmatrix}$$

Például a megjelölt elem értéke így jön ki: $2 \cdot 1 + 3 \cdot 0 + 1 \cdot 2 = 4$

Ez az algoritmus könnyen programozható. Ha négyzetes mátrixot négyzetes mátrixszal szorzunk össze, az eredmény is négyzetes mátrix lesz, ha pedig vektort szorzunk vektorral, a szorzat is vektor lesz. Ha több transzformációt akarunk egymás után végrehajtani, akkor vagy annyiszor szorzunk meg a pont koordinátavektorait a transzformációs mátrixszal, ahány transzformációt kell végrehajtani, vagy először összeszorozzuk egymással a transzformációs mátrixokat, majd a mátrixvektor-szorzá-

sokat minden vezorra csak egyszer alkalmazzuk. Belátható, hogy ha sok egymást követő transzformációt kell sok pontra alkalmazni, akkor ez utóbbi megoldással sok felesleges műveletet takaríthatunk meg.

Az 1. lista annak az include-fájlnak a listája, amelyben a transzformációhoz szükséges eljárásokat deklaráltuk. A 2. lista demonstrációs programot tartalmaz ezeknek az eszközöknek a használatára.

A LineSet és a Polygon eljárás egyenes szakaszok nyitlancát, illetve sokszöget rajzol. A Zero eljárás kinulláz egy 3×3 -as négyzetes mátrixot. (A Sinclair BASIC-kel ellentétben ugyanis a Pascalban a tömb deklarációjok csak helyfoglalás történik, a tömböt a program nem tölti fel nullákkal!)

A Translation, Rotation és Scaling eljárások a kinullázott transzformációs mátrix értékes elemeit töltik fel eltolás, forgatás és léptékezés esetére. A Matrixmult eljárás két 3×3 -as négyzetes mátrixot szoroz össze úgy, ahogy azt az előbb bemutatott, a VectorMult pedig négyzetes mátrix és vektor szorzatát állítja elő.

A Trans eljárás végrehajtja a tulajdonképpeni koordinátatranszformációt az egyes pontokra, és az eredményeket elhelyezi a memóriába.

A demonstrációs program beolvassa a felhasználó által megadott n darab pont koordinátáit, majd az ezekből szerkesztett sokszöget felrajzolja a képernyőre. Egymás után bemutatja az eltolás, a forgatás, az eltolás és nagyítás, valamint az eltolás, nagyítás, forgatás kombinációját. Példaképpen bemutatjuk, hogy egy egyszerű háromszög esetén a program milyen képet állít elő. A főprogram — úgy gondoljuk — nem igényel magyarázatot.

DR. KABOLDY PÉTER

ADOK-VESZÉK-CSERÉLEK

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hírdetéseket közlünk. A díjszabás: közületeknek gépelt soronként (60 karakter) 100,- Ft, magánszemélyeknek az első sor 50,- Ft, minden további sor 20,- Ft. Az NJSZI tagjainak az első három sor ingyenes. Hirdetéseiket a szerkesztőség címére várjuk.

Videoton TVC családi okok miatt sürgősen eladó. Major István, Pécel, Széchenyi u. 9. 2119.

C64, VC20 stb. mikroszámítógépekhez mozgó alkatrészek nélküli, "nem elromló", szenzoros botkormányt készítetek. Egy év garancia! Ára 1000 forint/db. Megrendelhető postán: Varga Csaba, Szeged, Kezesszőltes 28/A. fsz. 2. 6723.

Comodore 64-hoz GYORS-HÁTTÉRTÁR cartridge. Kapacitása 2-31 kb-át. A GYORS-HÁTTÉRTÁR-ba maximum 7 db, célszerűen gyakran használt program vihető be. A gép bekapcsolása után menüvel jelenti, hogy a gép bekapcsolása után a kiválasztott program azonnal fut. Javaslott program-csomagok: Turbo tape, Assembler, Monitor, Supergrafik, Help plus, Turbo másoló. File másoló 1699 Ft és Turbo tape 699 Ft.

Hozott programok elhelyezése a GYORS-HÁTTÉRTÁR-ban. Trompler László, Budapest, Attila u. 22. 1201. Tel.: 287-495 este.

VESZÉK

Primo típusú számítógépet vásárolok. Tartozékok, printer, szakkönyvek, prog-

ramkazzatták is érdekelnek. Árajánlatokat válaszborítékkal a következő címre kérek: Varsányi Gábor, Nagyatád, Aradi u. IX/C. IV.3. 7500.

Vannék üzembekötelen VC/VIC-20, C16, C116, C64, C Plus/4 számítógépeket és tartozékaikat, valamint VC-1541-es lemezegységet. Darabok is érdekelnek. Maróti Gyula, Bördány, Zákányszéki út 2. 6795.

CSERÉLEK

Enterprise gépet Sinclair QL-re vagy Schneider CPC 6128-ra cserélném. Varga Zoltán, Székesfehérvár, Lövölde u. 9/C. 8000.

ZX-Spectrum 48 k-ra készült színvonalas programokat cserélnék (Space Harrier, Feud, Xenon stb.). A válaszokat programlistával várom. Selyem Attila, Lengyelóti, Fenyődi u. 51. 8693.

48 k-zs ZX-Spectrum Interface II-vel, Kempston interfésszel, 200 db játékos és felhasználói programmal, német és magyar nyelvű kézikönyvvel együtt elcserélném egy Commodore Plus/4-re, magnóval vagy C16-ra belső bővítéssel. Kiss György, Budapest, Juhász Gyula u. 34. IV.15. 1039. 17 óra után

Bináris fák II.

A fán értelmezhető műveletek

A bináris fa tárgyalásánál használt fogalmak

A bináris fának az 1. rész végén (lásd előző számunkat) megadott definíciójában bevezetett jelölések mellett tételezzük fel, hogy T , T_1 , T_2 nem üres fák. Legyenek a felsorolt fák gyökerei rendre: t , t_1, t_2 . Ekkor azt mondjuk, hogy:

- T_1 és T_2 fa a T fa részfái. A sorrend általában nem közömbös, legyen T_1 a T fa bal részfája, T_2 a jobb részfája.
- t_1 bal fia, t_2 jobb fia T -nek.
- t apja t_1 -nek és t_2 -nek.
- t az első szinten van T -ben, t_1 és t_2 a második szinten. Ha valamely csúcspont az i -edik szinten van T -ben, akkor fia vagy fiai az $i+1$ -edik szinten.

— **Levele** a fának az a csúcspont, aminek nincs fia.

— Egy csúcspont **úttávolsága** azonos a csúcspont **szintszámával**. Ebből következően azonos a gyökértől a csúcspontig útbajett csúcsok számával, a gyökért és az adott csúcsot is beszámítva. A **úttávolság gráffal** való ábrázolásánál az út éleinek száma $+1$ azonos a csúcspont szintszámával.

— T **magassága** = csúcsai **úttávolságának** maximuma. Ez azonos a levelei **úttávolságának** vagy **szintszámának** maximumával.

Ha egy fát úgy építünk fel, hogy minden t csúcspont a bal részfán szereplő azonosítók kisebbek, a jobb részfa azonosítói pedig nagyobbak a t -hez tartozó azonosítónál, akkor a fát **keresőfának** nevezzük.

Bináris fa építése, keresés a fán

Keressük az x -szel jelölt azonosítót a fán!

Ha a fa üres, akkor x lesz a fa egyetlen eleme.

Ha nem üres, akkor a keresést a fa gyökeréből kiindulva kezdjük el. Ha az y csúcspontnál vagyunk, akkor a következő szabály szerint járunk el:

— Ha $x = y$, akkor az y azonosítót megtaláltuk a fán.

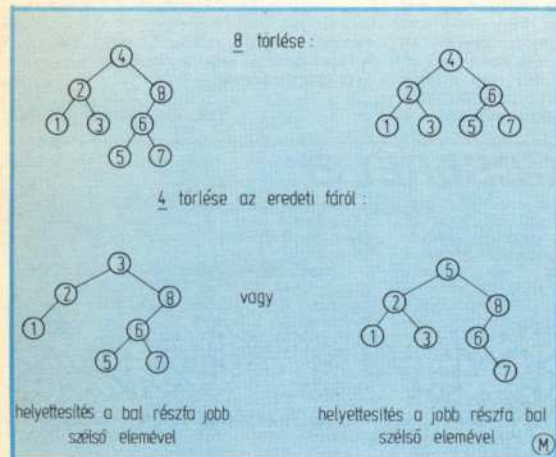
— Ha $x < y$, akkor y bal oldali fia lesz a következő útbajetendő csúcspont, ha létezik. Ha nem létezik, akkor x nem szerepel a fán, és y bal oldali fiaként kell beilleszteni.

— Egyébként $x > y$ esetben y jobb oldali fia lesz az útbajetendő csúcspont, ha létezik, számunkra az ún. **inorder (belső rendezés)** bejárás a legfontosabb.

Fa bejárása

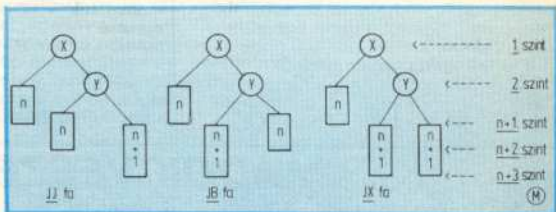
Előfordul, hogy valamilyen műveletet végre kell hajtani a fa minden csúcspontján. Ekkor egy algoritmus valamilyen sorrend szerint felkeresi a fa minden csúcsát, azaz bejárja a fát. Több bejárási mód létezik, számunkra az ún. **inorder (belső rendezés)** bejárás a legfontosabb.

Maradva a bevezetett jelöléseknél, ez a rekurzióval a következő: T fa inorder bejárása = T_1 inorder bejárása, t , T_2 inorder bejárása.

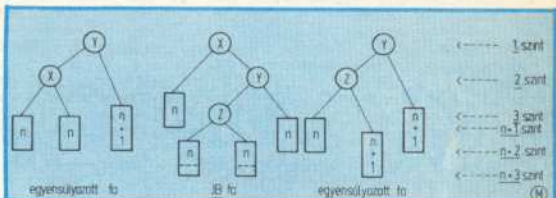


1. ábra

2. ábra

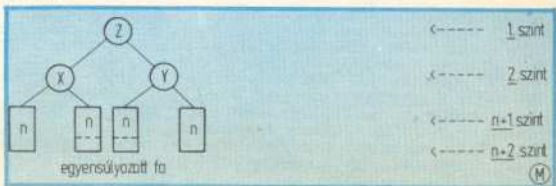


3. ábra



4. ábra

5. ábra




```

1801 CF00 XEF = X400
1802 CF00 KEFVER = XFE7
1803 CF00 SZINTAR = X0000
1804 CF00 SZIN = X
1805 CF00 AKT = XFE0
1806 CF00 SZM = XFD
1807 CF00 I ** = XCF00
2101 CF00 AV 00 INIT LDA #I KEF
2201 CF00 AV 00 STA I KEF
2301 CF00 85 FB STA AKT
2401 CF00 88 FC BTY AKT+1
2421 CF00 AV 00 LDA #I SZINTAR
2441 CF00 AV 00 LDY #I SZINTAR
2461 CF00 85 FD STA SZM
2481 CF00 84 FE BTY SZM+1
2601 CF10 AV 00 I FBRGAS LDY #0
2701 CF12 81 FB LDA AKT1,Y
2801 CF14 8A TAA
2901 CF15 C0 CJKLUB INY
3001 CF16 81 FB LDA AKT1,Y
3101 CF18 80 DELY
3201 CF19 91 FB STA AKT1,Y
3221 CF18 89 87 LDA #SZIN
3231 CF19 91 FD STA SZM,Y
3301 CF1F C0 INY
3401 CF20 C0 27 CPY #XZ
3501 CF22 D0 F1 BNE CJKLUB
3601 CF24 0A TAA
3701 CF25 91 FB STA AKT1,Y
3721 CF27 89 87 LDA #SZIN
3731 CF29 91 FD STA SZM,Y
3901 CF2B 1B CLE
4001 CF2C AD FB LDA AKT
4101 CF2E 69 20 STA #XZ
4201 CF30 85 FB STA AKT
4301 CF32 90 8C BCC TOV1
4401 CF34 E6 FC INE AKT+1
4421 CF3A 10 TOV1 CLE
4431 CF33 AD FB LDA SZM
4441 CF39 69 28 ADC #XZ
4451 CF3B 85 FD STA SZM
4461 CF3D 90 8C BCC VEGVIZ
4471 CF3F E6 FE INE SZM+1
4481 CF41 AD FC VEGVIZ LDA AKT+1
4701 CF43 C9 87 CMP #I KEFVER
4801 CF45 D0 F7 BNE FBRGAS
4901 CF47 AD F9 LDA AKT
5001 CF49 C9 87 CMP #I KEFVER
5101 CF4B 9B C3 BCC FBRGAS
5201 CF4D 80 BCB

```



Mit tud

a LINGUASOFT

A Software '88 kiállításon újra találkozhattunk a µ '87-en fődíjat nyert nyelvtanító programmal, a LINGUASOFT-tal. A rendszer alapötlete Gálai Antal bajai mérnöktől származik. A nyelvi anyagot Knáb Erzsébet, Medgyes Péter és Albert Sándor, a német, az angol és a francia nyelv egyetemi oktatói állították össze, nem kis munkával. A LINGUASOFT az egyik legelterjedtebb gépre, a C64-re készült. Aki rászánja magát a tanulásra, indulásnál négy menü közül választhat.

1. lista

```

1000 FOR I= 52992 TO 52999
1010 READ I:POKE I,X:BE=X:NEXT
1020 DATA 169, 0,160, 4,133,251,132,200
1030 DATA 149, 0,130,216,131,251,132,200
1040 DATA 149, 0,177,251,178,200,177,251
1050 DATA 156,145,201,149, 2,140,235,200
1060 DATA 142, 3,200,241,138,145,221,149
1070 DATA 7,145,223, 2,145,251,185, 40
1080 DATA 133,251,144, 2,238,252, 24,145
1090 DATA 255,145, 40,133,252,149, 2,238
1100 DATA 254,145,252,201, 7,200,201,145
1110 DATA 251,201,231,144,199, 94
1120 IF B< 32415 THEN PRINT"HIROG ADAT":END
1130 PRINT"OK":GOTO 111

```

2. lista

```

0 PRINTCHR$(147)
1 A=PEEK(2049):PEEK PAL "
2 A=LEN(A$):PRINTASCHR$(19)CHR$(17)CHR$(17)
3 FOR I=1 TO A:BE=RID$(A,I,1):PRINTB$;NEXT
4 BYSS2=92:REM KEFFDRGATAS
5 FOR I=0 TO 9:NEXT
6 N=N+1:IFN=3 THEN 0
7 PRINTCHR$(19)CHR$(17)CHR$(17)GOTO3
8 FOR I=0 TO 19:BYSS2=92:FOR J=0 TO 29:NEXT I:NEXT
9 POKES3278,PEEK(53278)AND247
10 N=N+1:PRINTCHR$(147):GOSUB25
11 FOR I=1 TO A:PRINTCHR$(19)CHR$(17)CHR$(17)
12 BE=RID$(A,I,1):PRINTB$CHR$(17)CHR$(17)
13 PRINTCHR$(17)CHR$(17)BE:BYSS2=92
14 FOR J=0 TO 9:NEXT J:REM KESELETETES
15 NEXT
16 N=N+1:IFN=3 THEN 0
17 PRINTCHR$(19)CHR$(17)CHR$(17)GOTO11
18 POKES3278,PEEK(53278)OR8:END
19 I
20 REM A KEFUJASD ADATAI
21 I
22 DATA " A TELJES KEPERNYON LEHET EVEL"
23 DATA " A PROGRAMMAL EGY KEFUJASOG"
24 DATA " MEDVALDITANI"
25 FOR J=1 TO 3:BE="":READR$:C=C+BE:NEXT
26 A=C$:A=LEN(A$):RETURN

```

3. lista

átállítás elmarad. Így megfigyelhetjük, hogyan íródik be az első oszlopba az információ. A 25–26 sorkokban található alprogram olvassa ki a DATA sorokból az adatokat és helyezi el az AS szöveges változóban. Legközelebb a mozgatható objektumok, a sprite-ok forgatásáról lesz szó.

SZABÓ PÉTER PÁL

Vizsgáztat, gyakoroltat

A vizsga üzemmódban haladhatunk a legyorsabbban, mert a rendszer ekkor csak osztályoz, vagyis méri a teljesítményt. Mondatonként és a teljes vizsga anyagára minősíti tudásunkat. Hasonlít a teszthez, azzal a különbséggel, hogy az eredmény azonnal megkapjuk.

A gyakorló üzemmódban már keményen meg kell küzdeni a továbblépésért, mert a helyes fordítás begépeléséig nem kapunk

új mondatot. Igaz, kérhetünk segítséget, de ezért fizetni kell. Az ismeretlen szavakat adott tarifáért megkérdőjelezhetjük a géptől: a képernyőn megjeleníti azok összes színominimáját a gyakorolt nyelven. Ekkor már csak választani kell, de a begépeléstől nem menekülünk meg. Nagyon hosszan nem lehet gondolkodni, mert a túrelmi idő letelével segítségül megkapjuk a legjobb megoldást. Ez a szó a képernyőn inverz karakter formájában, szalagszerűen jelenik meg, figyelemre szabva a tanuló, hogy az nem a saját teljesítménye. Amikor közösen kiizzadtuk a





helyes fordítást, az eltűnik a képernyőről, és önállóan kezdhetjük elölről. Ha sikerül helyesen begépelnünk a fordítást, akkor játékosan megjelenik, hogy mennyi pénzintéztünk, és jöhet az új mondat.

Gyakorlat teszi a mestert

Az edzés nagyon is érzékletes, mert a rossz vagy helytelenül begépelte szót a rendszer azonnal letörli a képernyőről, sőt még szírián is hozzá.

Sokunkkal előfordul, hogy a sikertelenségért a gépet okoljuk. Persze mindig neki van igaza. Számára a nagy- és kisbetű, a vessző és a pont felcserélése és elhagyása ugyanolyan hiba, mint a téves szó beírása. Ha kételkedünk is néha, végül a megoldást látva be kell ismernünk, hogy szigorú, de korrekt tanárral van dolgunk.

A fordítási beírások nem kell a képernyőt néznünk, mert hibáinkra azonnal figyelmeztet a már említett szírián. Az idő múlását gongütések jelzik.

A LINGUASOFT szigorú, de türelmes is. A programba beépített függvények — az áramkörökben levő tanár — különbséget tesznek a gépelési bizonytalanságból és a nyelvi tudás hiányából eredő hiba között. A beépített súlyfüggvény a tanuló alkatahoz igazodik.

Dolgozatot is kell írni

A harmadik menü a dolgozatírás. Ez a verzió a negyedikkel együtt csoportok, tankörök számára készült. Negyven diák próbálkozását, tudását mérheti mágneses memóriájában. Ez az opció hasonló a vizsga üzemmóddhoz: a tanárnak elég időszakonként ellenőriznie a tanulók haladását ahhoz, hogy teljes képet kapjon arról, mivel mit kell gyakoroltatnia. Így több idő marad az élőbeszédről.

Nem bújhatunk el

Most említjük meg, hogy a LINGUASOFT-tal csak bejelentkezés után lehet tanulni. Személyi szám közlése kötelező! Utána már nem lehet letagadni a sikertelen próbálkozásokat, javításra azonban van mód. A rendszer a tanfolyam teljes ideje alatt személyenként tárolja az eredményeket.

A dolgozatjavítás a negyedik üzemmód. A program nemcsak fáradhatatlanul korrepetál, hanem mentesíti a tanárt a legriasztóbb teherrel, a dolgozatjavítás alól. A tanár kényelmesen végignézheti, míg az áruklódó Júdás, a negyedik opció nagy sebességgel bemutatja, amit a tanulók a dolgozatírás során „elkövettek”.

Hogyan működik?

Vajon honnan tudja mindezt a LINGUASOFT nyelvoktató program? Tudjuk, hogy adott mondat nagyon sokféleképpen fordítható le. Nem tévedhet a program egy jó, de különös fordítási elbírálásánál? És még számos hasonló kérdést tehetünk fel.

A LINGUASOFT alkotói természetesen nem állítják, hogy rendszerük tévedhetetlen. Mindenesetre a nyelvészek és programozók munkája több megméretésen jelesre vizsgázott.

A rendszer — többek között — szinonimaszótár és nyelvhelyességi alrendszerből épül fel. Hogy hogyan, az a fejlesztők titka!

A szó begépelésig a LINGUASOFT csendben figyel. A szókész letyomása után a program megnézi a szótárában, hogy van-e ilyen szó. Ha megtalálja és nem észlel helyesírási hibát, akkor belép a nyelvtani, fordítási algoritmusok „épületébe”, és eldönti, hogy a szó helyes sorrendben érkezett-e. Természetesen számtalan lehetséges esetet kell figyelembe venni.

Jelenleg a rendszer kétszáz mondat fordításával gyakoroltat, illetve vizsgáztat. Az ezekhez szükséges összes szót, azok szinonimáját és nyelvhelyességi szabályait tartalmazza. A fordítási változatok száma néhánytól több ezerig terjed. A következő mondat például 122 880-féleképpen fordítható le: „Ugyanakkor az előadók semmit vagy csekély honoráriumot kaptak csak felépéseikért.”

A nyelvenkénti statisztikát táblázatba foglalva közöljük. Ez, ha nem is a teljesség igényével, de jól tükrözi a különböző idegen nyelvek „szabadságát”, variációs lehetőségeit.

A program különbséget tesz fordítás és fordítás között. Az értékelésnél figyelembe veszi a megoldás élőnyelvre jellemző gyakorlatát, de nem bünteti a választékos kifejezéseket sem. Az elfogadható, de kissé erőltetett fordításoknál azonban már „fizetünk” induló pénzeszkénkből, amelyet a gyakorlás elején kapunk, az adott nyelv országának pénzméneben.

A LINGUASOFT maximális figyelmet fordít a helyesírásra. Ismeri az ékezetes betűket: ezeket egyszerűen az 1-es melletti — billentyű segítségével állíthatjuk elő. A „manipulációkhoz” nem kell különösebb mutatvány, könnyen megjegyezhető. Ha azonban elfelejtjük vagy eltévesztjük valamelyiket, megszólal a hibát jelző szírián.

Felelő írógép

A LINGUASOFT-ot könnyű használni, nem kell hozzá előtanulmányokat folytatni. A menük megválasztásának módját és a funkcióbillentyűk hatását a képernyőről mindig leolvashatjuk.

A program erős középfokú nyelvtudásra épül, de használhatják kezdők is, mert bármelyik opcióban segítséget kérhetnek a szinonimaszótárból vagy a fordítás már beírt részének folytatásához.

A LINGUASOFT-tal betöltött Commodore-t felelő írógépnek is hívják, mert a hibákra azonnal reagál és visszazsal. Igaz, a felelő gérekek vagy feleséggel ellentétben ki lehet kapcsolni, de nem érdemes. Aki egyszer kipróbálta, nem tud leszokni róla.

P. GY.

A LINGUASOFT statisztikája

	angol	német	francia
Mondatok száma:	200	200	200
Magyar szavak száma:	1 637	1 748	1 707
Idegen szavak száma:	5 073	6 334	4 927
Összes fordítások száma:	261 644	28 626	33 146
Átlagos fordítások száma:	1 308	143	165



Gálai Antalt

a μ '87 Országos Mikroszámítógépes Találkozón ismertem meg.

Erről be is számoltam a Magazin 1987. júniusi számában.

A Software '88 kiállításon, a LINGUASOFT bemutatóján újra megkerestem őt.

A stand körül tolongó tömegben nem volt könnyű a közelébe férkőznom, de végül sikerült elrabolnom, hogy folytassuk közel egy éve megkezdett beszélgetésünket.

— *Hogyan alakult a LINGUASOFT nyelvoktató rendszer sorsa a μ '87-en aratott siker, a fődíj elnyerése után?*

— Már ott a kiállításon sok megrendelést vettünk fel. A sokszorosítás és másolás zökkenői miatt eleinte el is úsztunk egy kicsit a szállítással, de viszonylag hamar egyenesbe jöttünk. Most már rövid időn belül tudunk szállítani. Az érdeklődés nem lankad, folyamatosan kapunk megrendeléseket.

— *Terveztek, hogy bővítik a tanulható nyelvek körét, elsősorban a nemzetiségi nyelvekkel. Hogy lehet, hogy mégis maradt az eredeti kínálat, az angol, a német és a francia?*

— Mi minden tőlünk telhető megtettünk. Sokat tárgyaltam nemzetiségi kultúrházvezetőkkel, belföldi és külföldi intézetekkel, szervezetekkel, de mint kiderült, nem igénylik a rendszert. Így hát újabb nyelvekre nem is dolgoztuk ki.

— *Nem másolják illegálisan a LINGUASOFT-ot? Mi lesz, ha feltörik a programvédelmet?*

— Eddig még nem találkozom kalózpéldánnyal. Ez lehet a védelem érdeme is, de szerintem inkább a terjesztési rendszernek köszönhető. Nincs ugyanis két egyforma LINGUASOFT-hajlékonylemez. Speciális másolóprogramunk van, amely minden egyes másolatra rávisz valami kis eltérést. A lemezeket névre szólóan adjuk el, és a vevő tudomásul veszi, hogy nem másolhatja azokat. Ezt aláírásával is megerősíti. Ha illegális másolattal találkozónak, nyílvántartásunk alapján meg tudnánk állapítani, kinek a lemezéről készült. Örülök, hogy erre még nem került sor. A nyelvtanárok és programozók igen nagy munkát fektettek a rendszerbe, és biztosan elmenne a kedvük, ha másolni kezdenék szellemi alkotásukat.

— *Mikor fejeződött be a rendszer fejlesztése? Újra kezdené-e mai tapasztalataival a munkát?*

— A rendszert folyamatosan finomítjuk és tökéletesítjük. A jelenlegi 1987 májusában fejeztük be. Kilenc-tíz programozó és három nyelvtanár dolgozott rajta. Nagyon szeretem a közös művünket, de bánt, hogy sokan lebecsülik a benne lévő teljesítményt. Nem biztos, hogy újra kezdeném.

— *Hogyan tudta rávenni a nyelvészeket erre a számukra szokatlan munkára, és mit száltak a kollégáik?*

— Elég nehezen, de sikerült felkelteni az érdeklődésüket azaz, hogy nem könyvből, hanem hajlékonylemezen rögzítjük tudásukat. Gráfszerű sémákat dolgoztunk ki papíron és kértük, hogy töltsék ki azokat. Kollégáik nem nagyon értették meg őket; a LINGUASOFT-ban pedig annyira nem hisznek, hogy meg se nézzék.

— *Miért nem valamelyik elterjedt vállalkozási formát választotta ahelyett, hogy egy művelődési ház keretein belül működjön?*

— Néhány sikertelen próbálkozás után döntöttem így. A művelődési házzal jó a kapcsolatom. A forgalomból részesedik, így

érdeke a rendszer terjesztése. En a Szerzői Jogvédő Hivatalon keresztül kapom a díjazásomat. Ez így nekem teljes mértékben megfelel.

— *Térjünk vissza a LINGUASOFT-hoz. Milyen programozási nyelven készült?*

— Assembler nyelven, de vannak BASIC részei is. A program felépítése a fordítóprogramokéhoz hasonló. Az ötletet is egy rendszerprogramozásról szóló könyvből merítettem.

— *Vállalatok, szövetkezetek nem érdeklődnek a LINGUASOFT iránt?*

— Több vállalattal folytattam megbeszélést, de eddig konkrét üzlet nem jött létre. Egy amerikai céggel ellenben már alá is írtuk az értékesítési szerződést.

— *A standjupon PC XT-t is láttam. Ez azt jelenti, hogy a LINGUASOFT PC-re is elkészült?*

— Nem, azon a gépen egy levelezési segédlet program fut. A LINGUASOFT-ot ígértünk ellenére nem készítettük el PC-re, mert nem volt elég nagy az érdeklődés. Abban a körben, ahol ezt a rendszert vásárolják, még nem terjedtek el a PC-k. Elkészítjük azonban a LINGUASOFT alapfokú nyelvi változatát, amely sokkal kisebb és olcsóbb lesz.

— *Kazettán is lehet majd kapni?*

— Nem. Az a programozási módszer, amivel dolgozunk, nemigen valószínű, hogy kazettán. Az alapfokú LINGUASOFT után az idegen nyelvű levelezési segédletet fejlesztjük tovább. Ez a rendszer tulajdonképpen mondatkonzerveket tartalmaz, a Datorg hasonló című könyvei alapján. A mondatok fa-szerkezetben lekérdezhetők. Szinte nem is kell hozzá nyelvtudás; a magyar szöveg alapján a programban tárolt mondatkonzervekből, azaz szabvány mondatokból a képernyőn könnyen összeállítható egy üzleti levél. A főmenüből ki kell választani a kívánt témát: üdvözlés, árajánlat, reklamáció stb. Ezután a képernyőn megjelennek a típusmondatok, amelyek közül kurzorral lehet válogatni. Egyes mondatokban vannak kipontozott részek, például a termék neve számára. A megfelelő szavakat a szótár-alprogramból hívhatja le a levelező, tehát azokat sem kell ismernie. Mind több vállalat kapja meg az önálló külkereskedelmi jogot, de kevés az idegen nyelvű levelező. Ezért is dolgoztuk ki ezt a segédletet, egyelőre angol, német, francia, spanyol és magyar nyelvre.

— *Magyarra is? Vagyis külföldiek is használhatják, magyar nyelvű levelek megírására?*

— Igen. A program működési elve ugyanaz, csak a menü és a lekérdezési fa-struktúra az adott ország nyelvén készül. A külföldi forgalmazással a Datorg foglalkozik. Úgy tudom, néhány üzlet már létre is jött.

Rövid beszélgetésünk végén azt kívánom Gálai Antalnak, hogy ne veszítse el a kedvét, hanem ezután is minél gyakrabban hökkentsen meg bennünket újabb ötleteivel.

Történetesen, ha például Archimedes ma merülne fürdővizébe, bizony jó kétezer éves késésben lennének önmagunkhoz képest. Az így nyitott visszafordíthatatlan és tornyosuló lemaradásunk az sem segítene, ha pusztán fürösztéssel igyekeznénk felfedezni ama archimedesi törvényt. Vajmi kevesen jönnének rá. Szerencsénkre Archimedes azon kevesek közé tartozott, akik idejében vettek fürdőt. Noha esetlegesnek tűnik a tudománytörténeti példa, rejt magában egy nagyon fontos praktikus szempontot. A felfedező és a laikus közötti különbséget. A felfedező mindig pontos, zsenialitása szinte időtlen. A laikus viszont kullog utána, s egyáltalán nem mindig, mikor honnan szerez a heurékáról tudomást, mikor fedezi fel magának.

A hazai számítástechnikában e gond szembeötlő. Már a profi számítástechnikusoknak is nehéz a világ innovációs fejlődésének gyorsaságával ilyenét lépést tartaniuk. A nem számítástechnikus szakembereknek, a fentiek szerint úgymond laikusoknak, miközben egyáltalán nem szükséges a tudomány és technika XX. századi lenyűgöző dómját aprólékosan megismerni, munkájukban mind közelebb kerülnek hozzá. Munkaeszköznek a mikroszámítógépet választják. Igen, de éppen a választás pillanata a leggyötrelmesebb, mert hogy mit vegyen, honnan, ehhez talán legkevésbé információt. Éppen itt éri hidegzuhan. Hiszen ez a köztes helyzet, amikor a szakismeretre még nem kell képeznie magát, amire jószerivel fel lehet már kellő irdalmat, ellenben a tájékozottság nélkülözhetetlen. Ám — Si animus est, non deest auxilium: Hogyha van lelkesedés, nem hiányzik a segítség sem. Ez utóbbira vállalkoztak a Mikrokalauz szerzői a MESZ Számítástechnika kebelében.

A Mikrokalauz legnagyobb erénye, hogy naprakész. Azaz folyamatosan megújul a világból és az itthoni számítástechnikai információk forrásokra támaszkodva. A Mikrokalauz ugyanakkor sorozat, az egyes kötetek felölelik a hazai mikroszámítógép- és szoftverpiacot, a számítástechnikai irodalmat, s mindenekelött az első kötet betekintést nyújt a mikro-számítástechnikai alapokba és az egész iparág fejlődési tendenciáiba.

Könnyelműen és talán túlfönt pragmatikus, a vásárló önmagát sürgetve elhamarkodottan úgy ítélheti, erre a kötetre

egyáltalán nincs is szüksége. De gondoljon mindenki egy pillanatra mondjuk egy televízió vásárlására. Bármennyire egyszerű és közhelyeszerű az arról való ismeret, mégis érdeklődik az iránt, mit tud egy korábbi típushoz képest, mióta gyártják, és így tovább. Hát még akkor milyen fontos a mikroszámítógépekről vásárlás előtt egyet s mást tudni? Márpedig az első kötet a számítástechnika aktuális helyzetét körvonalazza ehhez nyújt kitűnő segítséget.

Az alapokról és fejlődésről írva a dolog természeténél fogva nem kerülhet meg a történetiség. Ez azonban a számítástechnika és ezen belül a mikroelektronika bemutatásában kifejezetten jótékony hatású, mert mint azt egy ókori filozófus mondata: minden tudás emlékezés csupán.

A Morszától a mikrogépig fejezet, az alapvető fogalmak tisztázásán kívül, rögtön bőkezűen ontja a legokoldalúbb fejlődési irányokat. Így például a világon a mikroprocesszorok gyártásának, illetve a mikroszámítógép forgalmának várható alakulását 1990-ig. Megtudhatjuk azt is, hogy az európai szocialista országok majd' mind egyike gyárt már 8 bites mikroprocesszort és a Szovjetunió, az NDK, valamint Csehszlovákia már a 16 bites mikroprocesszorok gyártását is megkezdte. A leendő felhasználót egyben eligazítja abban is, amikor a mohóság esetleg feleslegesen verné horribilis összegekbe, mert hogy: „Olyan viharos gyorsasággal kategóriaváltásra a 16 és 32 bites között nem kell számítani, mint amilyen a 8 és 16 bites gépek csatájában történt. A magyarázat igen egyszerű: a 16 bites gépek teljesítménye már messzemenően kielégíti a mindennapos felhasználói igényeket az alkalmazói tömegek esetében. Így például az irodai alkalmazásnál, s egy szerűen nincs kényszerítő motívója a 32 bites gépekre való áttérésnek. Ez a kényszerpálya csak igazán számítástechnikai alkalmazásoknál merül fel, mint a képfeldolgozásnál, a grafikanál, a tervezői munka támogatásánál, sokterminális (10 fölötti) lokális hálózatoknál. Tehát aki ma IBM PC-vel kompatibilis gépet vásárol, az hosszú távon, legalább 10 évig korszerű bázison, gazdag, hatékony szoftverállománnyal alkalmazhatja a számítástechnikát. Így közvetve részese lehet annak a termékfelruhászásnak, amellyel a világ legnagyobb számítástechnikai cége, az IBM, amely a maga 370 ezer dolgozójával hazánk éves nemzeti jövedelmével megegyező termelési értéket hoo létre évente, jelenleg kétszázalékos a központi termékét adja.”

A gép és a külvilág közötti kapcsolatot teremtő és tartó perifériákról szóló fejezetben a már nekilendült kíváncsiságot tovább csigázzák a szerzők. Nem maradnak itt sem adósak persze a háttérárak, nyomtatási típusonkénti bemutatásával. „A farkasétvágyt e körben is azonban a jövőt jelentő nyicsiklandozó falatokkal gerjesztik. Talán

a legizgalmasabb a Bernoulli Box elnevezésű tároló, mely nevét Daniel Bernoulli svájci matematikusról kapta. Ez az új tárolóeszköz rendkívül gyorsan terjed az Egyesült Államokban. 1986 tavaszán már több mint kétszáz ezer működött belőle. A kifejezetten mikrogépekhez szánt tároló egyesíti magában a hajlékonylemez hordozhatóságát, valamint a merevlemez kapacitását és sebességét. A legújabb modelljének méretei: 320 × 18 × 432 mm, s ez a doboz tartalmaz egy vagy két félmeleg meghajtót, tápegységet és hűtőventilátort. A meghajtók kapacitása egyenként 10 vagy 20 megabájt, így tehát 20-40 megabájt tárolási kapacitást kínál egyetlen egység. A 40 megabájtos változat ára 1986-ban Angliában 3500 font volt. A Bernoulli lemez egy kemény, hozzávetőlegesen A/4 méretű kazetába van zárva, maga a lemez hajlékony, átmérője 8 hüvelyk. Ezt a meghajtó 3000 fordulat/perc sebességgel forgatja, ami a mágneses hajlékonylemez tároló sebességének a tizszerese. A Bernoulli Box vonatkozásában jólatokba ma még korai bocsátkozni, de a paramétereit alapján reális konkurenciát jelent a winchester tárnaknak.” És vajon mikor jönnek el nálunk azok a boldog idők?

S hogy akad a tempón mit gyorsítani, azt mi sem mutatja jobban, mint az adathordozó lemezek gyártásának summája egy rövid összevetésben. A világon alig féltucatnyi cég gyárt hajlékonylemez-alapanyagot, s azt tekercsekben szállítja a gyárak többségéhez, melyek kiszereklik ezeket. A szocialista országokban egyelőre még nem gyártanak hajlékonylemez-alapanyagot. Kiszéréssel is csak a bulgáriai IZOT cég foglalkozik.

A számítógépes rendszerből a lélek sem maradhat ki. A szoftverrel pedig nem lehet léleklenül foglalkozni. Ezért is emelik ki a szerzők, hogy: „Végül is az alkalmazói programok döntik el, hogy mennyire hatékonyan elégti ki a számítógép az emberi szükségleteket. Ezért valószínű, hogy a személyi számítógép tulajdonosa többet fog szoftverre fordítani, mint hardverre. Költet úgy, hogy programokat vásárol, de úgy is, hogy jelentős mennyiségű időt fordít a megírásukra. Hacsak nem akar valaki intenzíven programozni, a hardver kiválasztásánál komolyan figyelembe kell vennie a rendszer szoftverbázisának szélességét és mélységét, azaz az egyes alkalmazásokhoz kapható különböző programok számát.”

Ezzel a gondolattal azonban már két nagyobb témához érkezünk, a hazai mikroszámítógép- és szoftverpiacot. De erről majd később írunk. Mindazonáltal maradjunk egy percre még az első kötet jelentőségénél, méltatásánál. Ha valakinek eltökélt szándéka a személyi számítógép használata, akkor igényeinek megfontolásához pontos iránytű a Mikrokalauz, habár bakot lőhet az is, aki egyébként jól célöz.

Szulyovszky Csaba

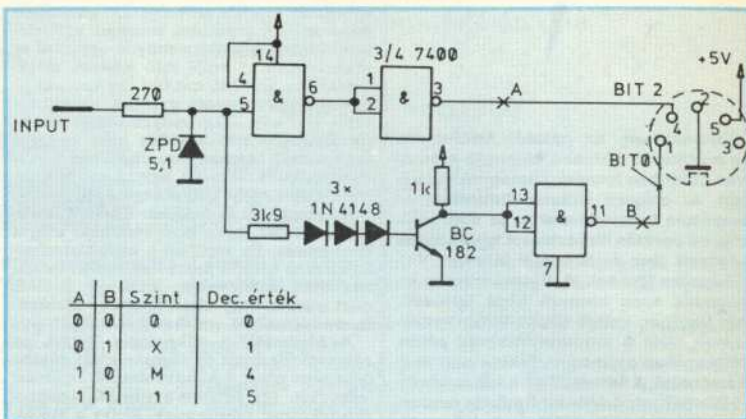
A digitális technikaával foglalkozók fontos segédeszköze a logikai szintjelző. Ez általában egy ceruza formájú, a hegyénél tűszerű tapintóval ellátott eszköz, amely a tapintóval megérintett áramkörti pont logikai állapotát mutatja.

Logikai szintjelző

Az állapotok a digitális technikában megszokott elnevezésekkel LOW (L=alacsony), HIGH (H=magas), MIDDLE (M=közepes) állandó szintek vagy gyorsan változó alacsony-magas szintű jelsorozatok, PULSE (P=impulzus) lehetnek.

A különféle állapotok különféle színű LED-ekkel jeleníthetők meg. A bonyolultabb szintjelzőknek tárolójuk is van, amely képes egy előző állapotot megőrizni. A szintjelzők áramkörti szempontból nem túl bonyolult eszközök és nem is nagyon drágák.

A következőkben bemutatott, személyi számítógép bemenetére kapcsolódó szintjelző kitűnő illusztrációja annak, hogyan lehet ezt az egyszerű eszközt — alapvető funkciójának megtartásával — intelligens



1. ábra

8000	CD	C9	01	FD	21	BC	02	21	F9	82	22	4B	40	21	E1	81"K@!..
8010	CD	75	2B	3E	06	CD	15	00	CD	5E	81	3E	80	CD	6D	81	.u+>.....^>...m.
8020	0E	04	06	00	CB	40	28	32	CB	48	28	02	18	05	79	FE@(2.H(...y.
8030	04	28	27	79	C5	01	0A	00	FE	05	28	0D	FE	00	28	0E	..('y.....(....(.
8040	FE	02	28	0F	11	00	03	18	0D	11	00	01	18	08	11	00	..(.....
8050	05	18	03	11	00	04	CD	68	3F	C1	CD	1D	00	CA	FF	00h?.....
8060	F5	CD	1D	00	20	FB	F1	FE	20	28	16	FE	54	28	91	FE(..T(..
8070	56	28	38	FE	4B	28	2C	FE	48	28	58	FE	4D	28	66	18	V(8.K(.,H(X.M(f.
8080	06	CB	48	20	7F	18	D3	D6	41	DA	FF	80	FE	04	38	03	...H...A...8.
8090	C3	FF	80	C6	41	CD	9B	81	3E	20	CD	9B	81	CD	4B	81	...A...>...K.
80A0	C3	FF	80	3E	16	CD	15	00	C3	00	00	FD	E5	D1	CD	80	...>.....
80B0	00	3E	20	CD	15	00	14	4	7A	FE	0F	20	02	16	07	D5	..>.....z.....
80C0	FD	E1	CD	80	00	3E	3E	CD	15	00	CD	5E	81	CD	4B	81>>.....^..K.
80D0	C3	FF	80	1E	10	CB	40	28	06	CB	80	3E	20	18	16	CB@((...>...
80E0	C0	3E	48	18	10	1E	0C	CB	48	28	06	CB	88	3E	20	18	..>H.....H(...>..
80F0	04	CB	C8	3E	4D	16	05	F5	CD	80	00	F1	CD	15	00	CB	...>M.....
8100	48	C2	5A	80	CD	AF	81	CB	48	20	05	FE	00	CA	24	80	H.Z.....H.....\$.
8110	79	FE	05	28	0D	FE	00	28	0E	FE	02	28	0F	21	4D	80	y...((...((...!M.
8120	18	0D	21	31	84	18	08	21	30	82	18	03	21	50	86	7C	...!1.....!0...!P..!
8130	CD	6D	81	CB	48	20	05	FE	80	CA	24	80	7D	CD	9B	81	..m..H.....\$.}....
8140	3E	20	CD	9B	81	CD	4B	81	C3	24	80	C5	FD	E5	D1	3E	>.....K...\$.....>
8150	04	83	5F	CD	80	00	21	39	83	CD	75	2B	C1	C9	C5	06	..._...!9...u+...
8160	1B	21	39	83	36	20	23	10	FB	36	00	C1	C9	F5	3E	02	..!9.6.#.6.....>.
8170	CD	15	00	11	04	02	CD	80	00	3E	20	CD	15	00	11	30>.....0
8180	01	CD	80	00	3E	20	CD	15	00	11	CC	00	CD	80	00	F1>.....
8190	CD	15	00	F5	3E	0C	CD	15	00	F1	C9	21	39	83	C5	01>.....!9...
81A0	25	00	09	23	E5	D1	2B	36	00	ED	B8	23	77	C1	C9	DB	%..#..+6.....#w...
81B0	40	E6	05	57	5F	C5	06	7F	DB	40	E6	05	A2	57	DB	40	@..W_...@...W_@...
81C0	E6	05	B3	5F	10	F2	C1	7A	BB	20	06	B9	28	10	4F	3C_...z.....(O<
81D0	C9	7A	AB	FE	05	20	D8	3E	02	B9	28	02	4F	C9	3E	00	..z.....>.....(O.>.
81E0	C9	02	20	5A	46	52	20	12	4C	4F	47	49	43	20	50	52	..ZFR.LOGIC.PR
81F0	4F	42	45	20	20	20	38	36	31	30	31	30	31	20	44	52	OBE 8610101 DR



2. ábra

tulajdonságokkal felruházni, használhatóságát kiterjeszteni.
 Először is a számítógépbe kell vinni az áramkör állapotát meghatározó jeleket. Mivel három különböző jelszint van (H, M, L), ezt az információt legalább két bit segítségével különböztethetjük meg.
 Ezután el kell dönteni, hogy a számítógépbe bevitt jelekkel egy időben a szintjelzőn is megjelenjen-e az információ, tehát hogy önállóan, a számítógéptől függetlenül is használható legyen-e az eszköz. Ez mérőműszerek esetében annak felel meg, hogy a műszer önállóan is kezelhető, beállítható, a mért érték leolvasható, de a műszer számítógéppel is vezérelhető. Terjedőben van

egy olyan irányzat, hogy a mérőműszereknél ezt a kettős funkciót megszüntessék: a műszerek előlapján levő — sokszor bonyolult kialakítású és meglehetősen költséges — kezelő- és kijelzőszerveket elhagyják, és a műszert kizárólag számítógép által vezérelt működésre alakítják ki. Ennek az az előnye, hogy sokkal bonyolultabb funkciók is könnyen megvalósíthatók a programozhatóság kihasználásával — nem kell millió billentyűt és azok megfelelő sorrendű megnyomását megtanulni, hanem a képernyőn megjelenő részletes információ alapján választhatunk. Egyszerűbb a műszer kisebb megbízhatóságú mechanikai részeinek kialakítása is. A kettős funkcióról való le-

mondás megfelelő gyártási környezetben olcsóbbá teszi a műszert. Esetünkben is célszerű lemondani a kettős megjelenítés lehetőségéről, mert ezáltal az áramkör egyszerűbbé válik.

A bemeneti logikai szinteket két bitevel rendelkező áramkört az 1. ábra mutatja.
 A mindössze egy SN7400 vagy SN74LS00 típusú integrált áramkört tartalmazó áramkör a bemenetére kapcsolt jel szintjétől függően az A és B kimeneteken TTL szintű jeleket ad ki, melyek a számítógép két bemeneti pontjára köthetők:

Jelszint a tapogatón	A kimenetek	B kimenetek
alacsony (L vagy 0)	0	0
közép (M)	1	0
magas (H vagy 1)	1	1

A következő probléma a beolvasott jelek (a két bit) feldolgozása. Mivel a szintjelzővel folyamatosan dolgozunk, a jeleket a számítógépbe folyamatosan kell beolvasni és megjeleníteni. Egy megoldási mód az lehetne, hogy minden beolvasás után megjelenítés következne. Ez azonban azért nem jó, mert olyan gyors, villogó kijelzési kép keletkezne, amely nagyon fárasztaná a szemet. A helyes megoldáshoz induljunk ki a gyakorlatból. Használat közben a tapogatót igen sokszor nem kapcsolódik semmihez, a „levegőben van”, és ilyenkor a kimeneteken az 10 (M) kombináció van. Ha a tapogatót a vizsgált logikai hálózat egy pontjához érintjük, az a következő állapotokat jellemezheti: alacsony (L), közepes (M), magas (H) szint vagy alacsony és magas szintek gyors változása, azaz impulzusok (P).

8200	4B 20 0D 20 2B 2D 2D 2D 2D 2D 2D 2B 20 20 20 20	K . +-----+
8210	20 20 20 20 20 20 41 2C 42 2C 43 2C 44 3A 20 4D	A, B, C, D: M
8220	41 52 4B 45 52 0D 20 21 20 20 20 20 20 21 20	ARKER. !
8230	20 20 20 20 20 02 20 20 20 20 54 12 3A 20 54 4F	T.: TO
8240	52 4C 45 53 0D 21 20 20 20 20 20 20 21 20 20	RLES. !
8250	20 20 20 20 20 20 20 20 20 20 20 20 20 20 20	V.: SORVALTA
8260	20 20 20 02 56 12 3A 20 53 4F 52 56 41 4C 54 41	S . +-----+
8270	53 20 0D 20 2B 2D 2D 2D 2D 2D 2D 2B 20 20 20 20	K.: KILEP.
8280	20 20 20 02 20 4B 12 3A 20 4B 49 4C 45 50 0D 20	UZEMMOD (H,M):
8290	55 5A 45 4D 4D 4F 44 20 28 48 2C 4D 29 3A 20 20	H.: HANG K
82A0	20 20 20 20 20 02 48 12 3A 20 48 41 4E 47 20 4B	I/BE. >S1:.. S2:
82B0	49 2F 42 45 0D 0D 3E 53 31 3A 0D 0D 20 53 32 3A	.. S3:.. S4:..M
82C0	0D 0D 20 53 33 3A 0D 0D 20 53 3A 0D 0D 02 4D	.. MINTAVETEL K
82D0	12 20 3A 20 4D 49 4E 54 41 56 45 54 45 4C 20 4B	I/BE, ERVENYESITE
82E0	49 2F 42 45 2C 45 52 56 45 4E 59 45 53 49 54 45	S SP-VEL. BfZZZZZ
82F0	53 20 53 50 2D 56 45 4C 00 42 66 5A 5A 5A 5A 5A	
8300	42 42 42 42 42 42 42 42 43 7E 42 42 42 42 42 46	BBBBBBBBBC~BBBBBF
8310	4A 52 62 42 42 42 42 42 7F 02 06 0A 12 22 42 02	JRbBBBBB... "B.
8320	02 02 02 02 02 02 02 02 03 7E 42 42 42 42 42 42~BBBBBB
8330	42 42 7E 40 40 40 40 40 40 00 00 00 00 00 00	BB~@@@@@A.....
8340	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
8350	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

A beolvasott első érték és a további értékek közötti ÉS kapcsolat az 1—0 változást, a VAGY kapcsolat a 0—1 változást mutatja ki.

A következő lépés a működési mód és a funkciók megtervezése. A modern programok döntő többsége ún. menütechnikát alkalmaz, azaz a képernyőn megjeleníti a program választható funkcióit. A felhasználó a funkcióhoz rendelt billentyű megnyomásával aktivizálja a kiválasztott funkciót.

A logikai szintjelzőhöz megtervezett képernyőkép a 2. ábrán látható.

Menü mellett jól láthatóan vonatkozó karaktert is: ez a képernyő bal felső sarkában lévő négyzetben látszik. A képernyőn a tapintó előző állapotai az ún. állapotosorba kerülnek, az új állapot megjelenésekor a sor egyvel jobbra lép.

A képernyőn négy állapotsor található (S1—S4), amelyekből mindig egy aktuálisat választhatunk ki a V billentyű segítségével, és ezt a > jelöli.

Az A, B, C, D karakterek leütésével az állapotsorban jelet (markert) helyezhetünk el. A leütött karakter megjelenik az állapotsorban, például megjelölve egy adott áramkörü rész vizsgálatának kezdetét.

Megoldható a tapintó állapotát jelző különféle magasságú hangjelzés generálása is (a H billentyűt nyomva bekapcsolja a hangjelzést, ismételt megnyomásra kikapcsolja.)

Mivel a közepes (M) szint nem kerül az állapotsorba, ezért ha például egy integrált áramkör minden lábát letapogatjuk, akkor az állapotsorban nem jelennek meg az ilyen állapotú lábakhoz tartozó értékek, vagyis az állapotsor nem képezi le teljesen a kivezetéseket. Ez is megoldható azonban, ha a tapintó állapotát nem folyamatosan, hanem egy adott parancsra (a betűkhöz billentyű megnyomására) mintavételezzük, és utána írjuk az eredményt az állapotsorba. Ez az üzemmód az M billentyű megnyomásával kapcsolható ki és be.

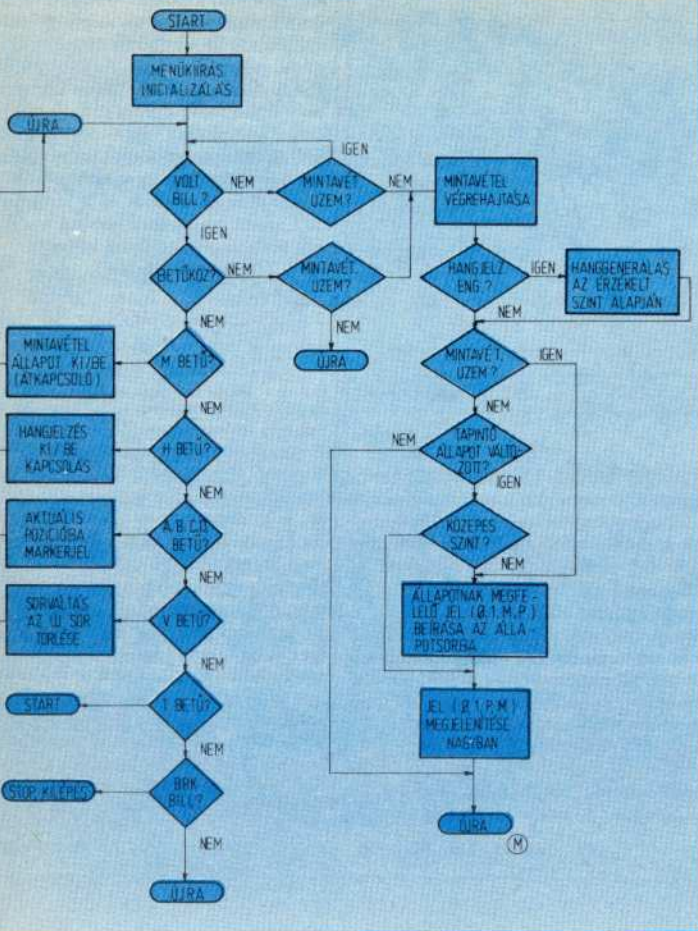
A fentieket megvalósító program blokkvázlatát a 3. ábrán láthatjuk.

A blokkvázlat alapján bárki elkészítheti saját számítógépére a programot, akár BASIC-ben, akár gépi kódban.

Mivel a logikai szintjelző eredeti, módosított változata Primóra készült, a Primo-tulajdonosok számára közöljük a gépi kódú program hexadecimális listáját. A program a 8000H címről indítható (BASIC-ből: ?CALL(-32768)).

Megjegyzés: Az 1. ábrán látható kapcsolású szerelt és működő logikai szintjelző áramkör a szerkesztőségén keresztül, írásban megrendelhető. Ára: 150,— Ft+utánvét. Megrendelhető a Primo-program önindulós, kazettán rögzített változata is. Ára: 100,— Ft+utánvét.

DR. KÓNYA LÁSZLÓ



3. ábra

Ezért a jó megoldás az, hogy amíg a tapogatón lévő jelszint nem változik, addig az állapotot csak egyszer jelezzük ki. Változáskor, annak jellegét figyelembe véve törtenek az újabb kijelzés.

A működési algoritmus a következő. Sokszor, például százszor egymás után beolvassuk a logikai állapotot reprezentáló két bitet. Ha ez minden egyes beolvasáskor azonos, a bemeneten statikus, állandó jelszint van. Ha a beolvasott sorozat nem azonos, akkor két eset lehetséges:

1. A beolvasás során a tapogatót levettük (vagy ráraktuk) a vizsgált pontról (pontra) és a bemeneten L—M, H—M, M—L, M—H átmeneti változások voltak, ami a tapintó mozgására utal. Újabb beolvasást kell végrehajtani, egészen addig, amíg vagy

statikus szintet, vagy impulzussorozatot nem érzékelünk.

2. L—H, H—L átmenetek érzékelése azt jelenti, hogy a bemeneten impulzussorozatot van.

Ezek után azt kell eldönteni, hogy az így kiértékelt állapot az előző beolvasássorozat állapotával megegyezik-e. Ha igen, akkor az állapotot nem kell megjeleníteni, mert az már előzőleg megtörtént. Különben ki kell jelezni, vagy a tapintó mozgására utaló kiértékelés esetén a beolvasást — kijelzés nélkül — meg kell ismételni.

A beolvasott jelsorozat kiértékeléséhez azt a tény használjuk fel, hogy ha a beolvasás során valamelyik bit értéke megváltozik, akkor ez két logikai művelet, az ÉS és VAGY műveletek bemeneti bitekre való alkalmazásával meghatározható.

Zene és tánc számítógépen

Izgalmas eseményre érkezett meghívó szerkesztőségünkbe: Rajka Péter és Ungvári Tamás, az Electronic Music in Sweden kutatóközpont munkatársai tartanak előadást; a programot video- és táncbemutató színesíti.



Anette Henriksson balett-táncos és tanár bájosá tette az általában száraz számítástechnikai módszerek ismertetését. Anette és — ne legyünk igazságtalanok — az előadók érdekes, sok ábrával illusztrált magyarázatai felejtették velünk a Nuntius interaktív, intermedialis és duális számítógépes rendszer ijesztő hangzását, ezen belül a Motográfhikont és a Macrostickont.

Mik is ezek a rejtélyesen hangzó dolgok? A Motográfhikon egy mozgásjegyzési, analízisi és animációs számítástechnikai, a Macrostickon pedig egy jelfeldolgozó, szintetizáló és megjelenítő zenei rendszer, szintén számítógépen.

Még egy fogalmat kell előre bocsátanunk: a notációt. Jelen esetben ez a zene és tánc (balett) feljegyzése.

A zene a művészetnek az az ága, amelyben az alkotás lényegét hangok különféle kombinációi fejezik ki. A hangok időbeli egymásutánja dallamot alkot, egyidejű megszólalásukból pedig harmónia keletkezik. Mozdító ereje a ritmus, sebessége a tempó. A balett több száz éves hagyományon alapuló színpadi tánctechnika: 1662-ben Franciaországban foglalták a szabályait rendszerbe.

Jogos a kérdés: mi köze e két művészeti ágának a számítástechnikához? Amikor az NJSZT meghívóját kézhez kaptuk, minket is ezek a kérdések foglalkoztattak. Rajka Péter és Ungvári Tamás hamar eloszlatta a kételyeinket. Ungvári Tamás elmondta, hogy a zenei notáció viszonylag hamar kialakult a kottázás (hangjegyzírás) formájában, de több olyan hiányossága van, amely a zene tánchoz való rendelését is nehezíti.

A számítógéppel komponált zenével kapcsolatban az egyik fő ellenvetés az, hogy a nemzetközi szerzői jogvédelem még nem fogadja el az ezidáig lehetséges dokumentálási formákat (magnófelvétel, programlista), és így nem lehet dokumentálni; védeni sem tudják ezeket az alkotásokat.

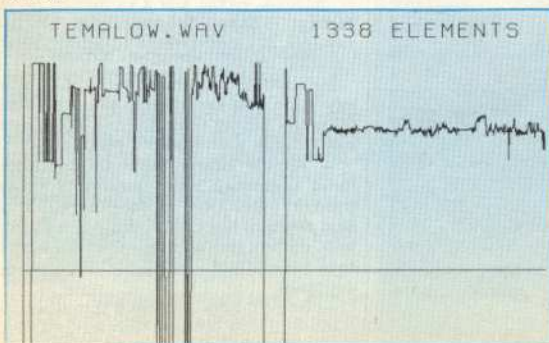
Ungvári Tamás egy DEC PDP—15 modellel, illetve VAX—11/750 rendszerrel dolgozik, komponál. A rendszer lelke az

FP 120B tömbprocesszor. Ungvári Tamásnak a zenekritikusok és a hallgatóság véleménye szerint is valóban humanista — nem kemény és nem embertelen — zenéjét meghallgathattuk. Művészeti munkásságán túl a többségben szakmánkbeli közönséget megragadta a művészember számítástechnikai teljesítménye: kialakította a számítógépen komponált zene notációs rendszerét. Az általa megálmodott és számítógépen megvalósított hangzások megjelennek a papíron szonogram és spektrogram formájában (1. és 2. ábra).

A szonogram esetében a gép által nyomtatott koordinátarendszerben az egyik tengelyen a hang frekvenciáját, a másikon az időt látjuk. Ez már majdnem tökéletes lejegyzése a zenei műnek. A spektrogramon pedig a fentiekben kívül a hangzás erőssége is látható, az árnyalatok által.

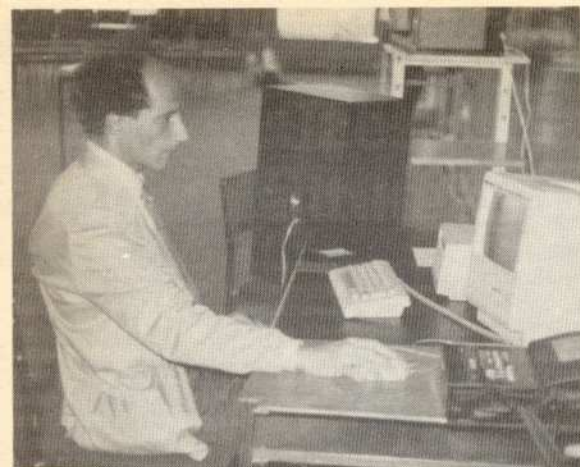
A Motográfhikon a tánc lejegyzési rendszerét forradalmasíthatja, de még ennél is többet jelent. Rajka Péter koreográfus a tánc notációjában hasznosítja a modern technikát. A tánc mozdulatainak lejegyzése sok évszázados igény. A bemutatott jelentekek reprodukálása már a középkori Kínát is foglalkoztatta. Így több notációs rendszer

1. ábra



2. ábra





Új koreográfia születik

Együtt a művészpáros

született. A 3. ábra egy ilyen próbálkozást szemléltet. A ma használt rendszereknek is fő hiányosságuk, hogy nem jelölik az időt, és így a zenével való kapcsolatot sem lehet egyértelműen leolvasni. Ez az alapvető igény készítette az új notációs eljárás kidolgozását. Az első lépés a szimbolikus rendszer kialakítása volt. Ezáltal a térben a jelek, az időben a függőleges tengely igazítanak el. A 4. ábra a szimbólumok és a mozgás kapcsolatát mutatja be. A jeleket a számítógéphez egérrel vagy billentyűzetről lehet bevinni.

Megkérdeztük Rajka Pétert, hogy a videotechnika korában miért van szükség a mozdulatok lejegyzésére. Elmondta, hogy a filmen látható kép nem ad egzakt lehetőséget a táncosok elkülönülő mozdulatainak térbeli reprodukálására, de a legnagyobb baj az, hogy a nézőben, a koreográfusban óhatatlanul szubjektív érzéseket kelt. A papíron vagy számítógéppel lejegyzett mozdulatok ellenben a „vegytiszta” mozdulatokat adják vissza.

Az általa Macintosh számítógépre írt program lehetőséget teremt a tánc notációjára, a mozdulatok elemzésére, sőt az animációs rész egy kis figura alakjában be is mutatja a művet a képernyőn.

Anette Henriksson az előadás után elmesélte, hogy az illusztrációként bemutatott balett koreográfiájára már nem is emlékezett, de hajlékonyan elhozott lejegyzés alapján negyedóra alatt újra megtanulta. Sőt, a Motográphon segítségével új mozgáselemeket, kombinációkat fedezett fel.

A rendszer nem csupán a meglévő művek notációjára alkalmas, hanem új koreográfia készítésére is — így a gépen lehet megtervezni a mozgást. Van egy segédprogram is, amely a tánc dinamikáját kifejező jellegű görbéket állít elő (5. ábra).

Különös módon vehető fel ezután, hogy a tyúk volt-e előbb vagy a tojás? Ugyanis a művész—számítástechnikus páros bemutatkozása előtt az volt a helyzet, hogy szinte kizárólagosan a zenei mű született meg

előbb, és utána a balett koreográfiája. Most már egy kicsit a feje tetejére állhat a világ!

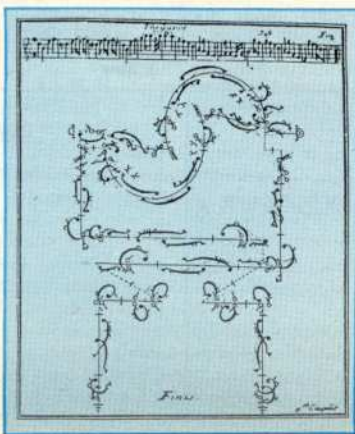
A szimbolikus rendszerből készült jellegű görbék a zeneszerzőt inspirálják, orientálják, mivel látja, munkája során eleve számolhat a hangzás és a tánc együttes dinamikájával. A művészek között újfajta kommunikációs rendszer alakul ki, mivel a számítógépes zene jelrendszere is visszahat a koreográfus munkájára. Az 5. ábrán olvasóink tanulmányozhatják a két művészeti ág alkotásainak elvonatkoztatott jelrendszerben való közeledését, összhangját.

A bemutató után interjút készítettünk az előadókkal igazán érdekes pályafutásukról.

Nagybőgőtől a számítógépig

Ungvári Tamás 1936-ban született Kocsán. Hamarosan Budapestre költöztek. A József Attila Gimnáziumban érettségizett. Középkorában elkezdett hege-

3. ábra



dűn és gordonon játszani, és egyébként is hamar kiütközött alapvető humán beállítottsága. Ez a későbbiekben is végig befolyásolta életét. Ebben az időszakban például megszervezte Budapest összes iskolájának szavalóversenyét.

Hubay-növendékként folytatta a hegedülést, de korábbról hozott rossz beidegződése komoly nehézségeket jelentett — meg is akad egy időre ez a vonal. Érettségi után a Budapesti Lenfongyárban dolgozott, mint tujavító munkás. Erre úgy emlékszik vissza, mint élete egyik legszebb időszakára. Mégis visszahúta hajlama a művészethez: beiratkozott a Zeneművészeti Főiskolára. 1956-ban a Fővárosi Operettszínház rendes tagja lett, 1957-től az Állami Hangverseny Zenekarban játszott. A tanárképzőt is elkezdte, de saját bevallása szerint nem volt kirtartása hozzá.

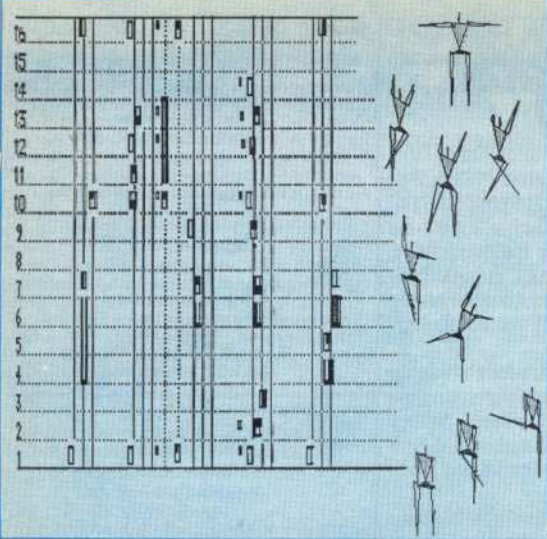
— A filozófia mindig is érdekelt — mondta —, de abban az időben szinte megszállottként foglalkoztam vele. Folyvást újat és újat akartam. Menekülni a konvencióktól! A rutintól való elszakadást kerestem, és a látszólag összefüggéstelen részekben az összefüggést. Nem azt tartottam fontosnak, amin vagy amivel dolgozom, hanem a mögöttes tartalmakat.

1960-tól a salzburgi zenei főiskolán tanul és dolgozik. Ekkor már a karmesteri pálya vonzza. 1970-ben Svédországban két házasságot, és karmestertként különböző zenekarokkal lép fel.

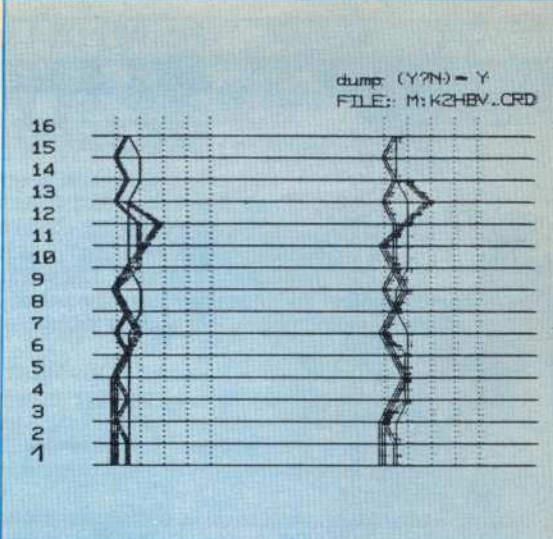
Firenzében karmesterversenyt nyer ugyan, de nem kap állandó állást. Alkalmi fellépései mindig sikert hoznak, de ez nem elég. Megint a továbblépés útját keresi.

— Amikor először hallottam szintetizátoron komponált zenét, megfogott annak iszonyú potenciálja és lehetőségei. Még nem éreztem, hogy zeneszerzőnek kell lennem, de nem tudtam megszabadulni a gépi zene hangzásától — magyarázta pályamódosítását.

— Semmit nem értettem a számítástechnikához. Örült butának éreztem magam, amikor bejutottam a stúdióba. Programré-



4. ábra



5. ábra

szeket vittem haza, és egyfolytában tanulmányoztam azokat. Nézem, néztem és egyszer csak összeállt.

— Milyen viszonyban van a programozókkal?

— Elég feszült a kapcsolatunk. Általános probléma, hogy lenézik a humán érdeklődésűeket. Az is gyakran előfordul, hogy ha ismerik a programot, már valódi zeneszerzőnek érzik magukat. Pedig a tökéletes hangzásokat, a dinamikai arányokat nem lehet előre „beprogramozni”. Ehhez már a művész ihlete kell.

Ungvári Tamást most is elsősorban a filozófia és nem a matematika érdekli. Szép karrier, hogy ennek ellenére a Svéd Zeneakadémiák Intézet számítástechnikai osztályát vezeti.

Koreográfia és kibernetika

Rajka Péter 1949-ben Budapesten született. A Kossuth Zsuzsa Gimnáziumban latin tagozaton végzett. Édesapja freudista, pszichoanalitikus volt, és a családban szinte a levegő alkotóelemeként vette őt körül a humán műveltség. Egész fiatalon, gimnazista korában már koreográfiaikat irt.

— Milyen megfontolásból kezdett a tánc irányszisztemével foglalkozni? — kérdeztük.

— Teljesen ösztönösen. Érdekelt a mozgás kódolása. Először a zenei kompozíciós elveket tekintettem irányadónak, később a hagyományos lejegyzési rendszereket tanulmányoztam. Bár általában a legjobbnak tartják és ezért világszerte elterjedt a Laban-féle notációs rendszer, nekem hiányzott belőle a kibernetikai megközelítés lehetősége.

17 éves korában balettozni tanult, és nem sokkal később már a Tháliában és a Vig-színházban táncolt. Katonáskodása alatt is

a mozgás struktúrája és a kifejezőmód összhangja foglalkoztatta. Leszerelés után azonban nem találta meg a folytatás általa igényelt lehetőségét.

— Ha jól tudom, ebben az időszakban a kaposvári színházban dolgozott koreográfusként.

— Valóban, de igen ellentmondásos volt a légkör. Először támogattak. Fejlődtem is, de hamarosan visszahúzott a színház vezetésének beállítottsága.

A koreográfia megvalósul



— 1974-ben kivándorolt Svédországra.

— Igen, újrakezdeni az ismeretlenben. Filozófiával foglalkoztam és a tánc strukturalista elméletével. Információelméleti könyveket olvastam, és megfogott a kibernetika. 1975-től a svédországi táncfőiskolára jártam, és csak a tanulásnak éltem. Az újromanticizmus technikaellenessége azonban nem kedvezett törekvéseimnek. Én úgy próbáltam „komponálni”, mint egy zeneszerző. Moderntánc-együttest alakítottam Multimedia néven. Ekkor már számítógépen szerzett zenére készítettem koreográfiát.

— Milyen különbséget tesz a hagyományos és a gépi zene között?

— Nem a zenében levő különbségek érdekelnek! A kommunikáció a forradalmi rendszerben, az egzakt kölcsönhatás a tánc és a zene között.

— A számítógépre a programokat saját maga készíti el?

— Nagyrészt igen, turbo Pascalban. Most már bedolgoznak a Királyi Műszaki Főiskola tanulói is. Például az animációs program jó részét ők készítették.

— Mit szolt a szakma az új „komponálási” módszerhez és az egész számítástechnikai rendszerhez?

— A művészek furcsállták, de hagyták. Ellentmondásos érzés, hogy olyan közegben mozogok, ahol nem tudnak vitakozni velem.

— Kutatónak vagy művésznek érzi magát?

— Elsősorban művésznek, koreográfusnak. A számítástechnikát a mozgás elemzésére, a zene és a tánc szerkezetének megfeleltetésére használom elsősorban, természetesen a notáción kívül. Egyébként a rendszert még nem tekintjük teljesnek; nagyon sok tisztázatlan részlet van.

Sok sikert kívánunk a fejlesztéshez!

PINKE GYÖRGY

DIGITAL
szelvény
Mikroszámítógép
Magazin
1988. március

Egy sarokkal olcsóbb!!!

A DIGITAL

Számítástechnikai Szaküzlet (1026 Bp. Szilágyi Erzsébet fasor 35.) Sinclair-termékekre szakosodott: elsősorban a ZX81-es és Spectrum gépekhez használatos eszközöket, programokat árul. De kínál más gépekhez

való tartozékokat (például botkormányokat), számítástechnikai alkatrészeket (integrált áramköröket stb.) és zsebszámológépeket is.

Aki ebben az üzletben a lapunkból kivágott sarokszelvényt átadja, vagy megrendelésével együtt oda elküldi, minden hónapban más-más cikket olcsóbban vásárolhat meg. A kedvezmény a szelvényen feltüntetett hónapban érvényes. Minden árengedménnyel vásárolt darabhoz le kell adni egy szelvényt.

A bolt utánvétellel szállítást is vállal, és a szokásos 6 hónap helyett **1 év garanciát ad.**

Az e havi kedvezmény

Két botkormányos illesztő fényceruzával 2500,— Ft, engedmény 20%

Pisztoly formájú botkormány 780,— Ft, engedmény 17%

Két botkormányos illesztő fényceruzával

A Spectrum gépek buszcsatlakozójára köthető, kapcsolós (Atari, Commodore típusú) botkormányok illesztésére szolgál. A játékprogramokhoz az ún. Kempstone típus a szabvány, amely egyaránt alkalmas az egy és két botkormányos játékokhoz. Sok játékot billentyűzetről is lehet játszani, de mivel ez hamar tönkreteszi a billentyűket, célszerűbb botkormányt használni.

A külön illesztővel csatlakoztatott fényceruzával rajzolhatunk és törölhetünk a képernyőn, majd a kész ábrát el tárolhatjuk, magnetofonszalagra vihetjük.

Pisztoly formájú botkormány

A leggyorsabb botkormány. A benne lévő, találmány tárgyát képező szerkezetnek köszönhetően ez az egyetlen olyan típus, amelynél nincs szükség a

törés megakadályozását szolgáló fékre. Ennek következtében olyan gyorsá vált, hogy valamennyi versenyben gyorsabbnak bizonyult a többi botkormányánál. Egy összehasonlító mérés szerint például a Quickshot II-höz képest 2,2-szeres a sebessége. Ebben a kísérletben 4 szezelynek a lehető legnagyobb sebességgel 100—100 jobbra-balra kapcsolást kellett végeznie mind a kétféle botkormánnyal. (A Commodore újság felkérésére is végeztek összehasonlító méréseket az itthon elérhető botkormányokkal — köztük ezzel is —, de az eredményről lapzártáig nem értesültünk.)

A fékhány miatt igen kis erő, egy kisgyerek ujjának pöccintése is elég ennek a botkormánynak a működtetéséhez. Így lehetővé válik a nagyon pontos beállítás, ami például a rajzkészítő programoknál nagy előny. A bot — annak ellenére, hogy minimális erőhatásra is reagál — az erős rángatást is kibírja.

A konstrukció előnye még, hogy meghibásodás esetén egyszerűen javítható. A formája pedig a lövöldözős játékokhoz „testhezálló”.

A pisztoly formájú botkormány a II. Mikroszámítógépes Találkozó hardverpályázatán a legötletesebb megoldás díjat kapta.

S. E.



Az NJSZT **kiütetések** **tüntetettjei**

A hagyományokat folytatva, a Neumann János Számítógéptudományi Társaság 1987-ben is kiütette élenjáró aktivistáit. Az Országos Elnökség határozata alapján az alábbi tagok részületek kiütetésben:

Neumann János-díj

Kertész Ádám
Tóth Istváné
NJSZT, ügyvezető főtítkárhelyettes
Pál László

OMFB, főcsoportfőnök

Kalmár László-díj

Arató Mátyas
SZÁMALK, tudományos tanácsadó
Cser László
BME, számítóközpont-vezető
Csaba László

MTA—SZTAKI, főosztályvezető

MTESZ emléklap

Dr. Kovács Péter
MNB, főosztályvezető-helyettes

A számítástechnika eredményes oktatásáért és a számítástechnikai kultúra terjesztéséért Tarján

Rezső-díjat kaptak:

Dusza Árpád
országos számítástechnikai szaktanácsadó
Kovács Mihály
ny. Piarista gimnáziumi tanár

Szlávi Péter

ELTE, programozó matematikus

Zsakó László

ELTE, programozó matematikus

Tanulmányi munkájukért kiütetett egyetemi hallgatók

BME TDK-díj

Mandzsu Zoltán, elméleti villamoságytan tsz. a „Memóriaorientált digitális transzmultiplexer szimulációja” és

Bányai Ervin, folyamatszabályozási tsz. a „Tervezési módszer párhuzamos végrehajtású, több automatát leíró vezérlési folyamatok tervezésére és rögzítésre (szinkron fázisregiszteres) struktúrában történő megvalósításra” c. dolgozatával elért első helyért.

Az NJSZT diplomaterv- és szakdolgozat-pályázatán 1987-ben 11-en indultak. A bírálóbizottság úgy határozott, hogy a kiírtnál kevesebb díjat ad ki, tekintettel a kevesebb beérkezett pályaműre. Két pályázó kapott első és kettő harmadik díjat.

Első díj

Görbics Livia, volt matematika—kémia—számítástechnika szakos hallgató „Számítógéppel csatolt kémiai demonstrációs kísérletek” c. szakdolgozatáért, melyet az ELTE TTK fizikai—kémiai és radiológiai tanszékén készített.

Paizerné Tóth Timea, volt német—történelem szakos hallgató a „Computeresítettzer Deutschunterricht” c. szakdolgozatáért, melyet az ELTE BTK Audiovizuális Technikai Központjában készített.

Harmadik díj

Mohay Tamás, volt műszer- és irányítástechnika szakos hallgató a „Grafikus nyelv vezérlési feladatok megoldására” c. diplomatervéért, amelyet a BME VMK műszer- és mérés technika tanszékén készített.

A legjobb főiskolai dolgozat szerzője

Weber László, volt számítástechnika szakos hallgató „Vonalkódolvasó és -generáló program IBM PC/XT számítógépre” c. szakdolgozatáért, amelyet a Kandó Kálmán VMF Számítástechnikai és Matematikai Intézetében készített.

A BME Villamosmérnöki karán meghirdetett programozási versenyen

Verhás Péter műszaki—fizika B szakos hallgató, BME „Kozma László” digitális technika versenyen elért első helyért
Pintér Gábor részületek elismerésben. Gratulálunk!

Hogyan válhatunk „kód és dekód fiai”-vá?

A számítógép-élvezet akkor ér a csúcspontjára, amikor az ember a tiszta hardverrel kerül szembe. Erre kétféle minőségben lehet módja: hardveresként és szoftveresként.

A hardveres esete persze kézenfekvő, hiszen a szakmájából következik, hogy hardverrel dolgozik, igaz, egyre több szoftver beütéssel. El kell ismerni, hogy a számítógépek izzig lehetőségei nem annyira a billentyűzethez, mint inkább a gép hátoldalán található interfészekhez kapcsolódnak. Most azonban nem a hardveresekkel foglalkozunk.

hogy gépeink nem ismerték a magyar ékezetes betűket. Erre persze a dokumentáció is utal, de a következőknyekről nem beszél. Nos, egyes kérdések megválaszolásánál kerülhetünk bajba.

A programok többnyire jól működnek, és valamelyest a dokumentáció is segít. Kár, hogy fogalomtárának néhány magyarázata hiányos. A tanulás interaktív módon, kérdések és válaszok formájában történhet. A hibás válaszokra a hibajelzés mellett érdekes informatív példák jelennek meg.

tív címzési módot, mivel annak definíciója sehol sem található meg a programcsomagban, sőt a dokumentációban sem. A relatív cím egy komplexens kódban ábrázolt előjeles érték, amely egy bázis(regiszter) értékhez adódik hozzá, hogy megkapjuk az operandus címét.

A címzési módokat oktató program a programcsomag legalacsonyabb színvonalára teszi.

A TXA utasítás definíciójában az „áthelyezés XR—AC között” félreérthető, túlzottan az angol rövidítés lefordítására koncentrál. Jobb, ha azt mondjuk: a TXA az XR regisztert az AC-be másolja.

Az STA utasítás definíciójában hibás. Helyesen az STA az AC tartalmát az operandusként megadott címre másolja.

Az egyik példaprogram a 7501 gépi kódjában nem létező TXY utasítás használata miatt működésképtelen. A feladat működő megoldása: LDX #500; LDY #500; DEY; BNE \$FD; DEX; BNE \$FA.

A program egyik szövege ez: „Iráj l-1 utasítást!” Eltartott egy darabig, amíg rájöttünk, hogy mit akar. Elsőre úgy értelmeztük, hogy gépi kódú utasítást vár. Később derült ki, hogy utasításmemoronikot szeretne kapni. Rádásul a program szokásától eltérően nem reagált a hibás válaszra, így nem árulta el a megoldást. Egyébként nagyon korreket a válaszok elbírálásában. A sürűn előforduló érdekes keresztkérdésekre adott hibás válaszok esetén megjelenik a helyes válasz és a kimerítő magyarázat.

A program jól kihasználja a számítógép nyújtotta előnyöket: látványos szemléltető rajzokkal, filmszerű diagramokkal igyekszik megmagyarázni a bonyolult eseteket.

A programcsomaghoz tartozik egy egyszerű assembler is, de inkább „igazi” assembler használatát javasoljuk. A magát az assemblert megvalósító programot azonban érdemes tanulmányozni.

Hiányként róható fel a programcsomagnak, hogy nem tartalmaz olyan teszteket, amelyekkel a tanuló felmérhetné tudását. Aki már tud valamit, annak nem kellene végigvennie az összes leckét, hanem a számára homályos részekkel foglalkozhatna. Egy interaktív csomagtól ez joggal elvárható.

A hibák ellenére a programcsomag szélességében „jó” minősítést kapott tőlünk. A minősítés részletezését lásd a 2. táblázatban.

Végül megjegyezzük, hogy a programcsomag nem pótolja az egyébként bőséges választékban rendelkezésre álló, jó minőségű kézikönyveket.

ZSADÁNYI PÁL—íjf. ZSADÁNYI PÁL

2. táblázat

Kezelhetőség:	jó
Teljesesség:	jó
Dokumentáltság:	közepes
Használhatóság:	jó
Ár/teljesítmény:	közepes
Összbenyomás:	jó

Terméknév:	Gépi kódú oktatóprogram-csomag
Forgalmazó:	Novotrade - Octasoft
Szerzők:	Bereznay Tibor, Pajor Gábor
Géptípus:	C16, C Plus/4
Hordozó:	három kazetta
Dokumentáció:	22 oldal
Ár:	606,- Ft

1. táblázat

Amikor az ember szoftveresként kerül szembe a hardverrel, igen kemény feladatot kap: mindent saját magának kell megszerzenie az üzembiztos működéshez. Aki erre vállalkozik, az a szoftveresek megkülönböztetett kasztjához tartozónak vallhatja magát: rangja minimálisan „gépi kódú” vagy „assembler” programozó, sőt „rendszerprogramozó”, esetleg „rendszer-szoftveres” — a szakzsargon szerint: „bitfaragó”.

Egy hétköznapi szoftveresnek jörszerű fogalma sincs arról, hogy mennyire távol dolgozik a hardvertől, és hogy milyen kényelmi eszközökkel halmozza el már egy vacak beégetett BASIC interpreter is, nem beszélve az operációs rendszerekről. Ezen csak akkor kezd el gondolkozni, amikor a gépkönyv vagy egy jó cikk hatására felbuzdul, és a gépi kód vizenység talajára lép. Rendszerint pillanatokon belül olyan vereségeket szenved, hogy örökre elmegy a kedve az ilyen kalandoktól. Csodálatosnak hitt gépe elkéserítően elbuzdul, sokszor még egy nyamvadót hibajelzést se tud kinyögni, vagy összevissza zagyvál.

Nos, ifjak — bátrak és merészek, tízezer-től akárhány évesig —, vágjatok neki e nagyon nehéz mesterségnek, mert bár elsajátítása szívósságot igényel, azért nem reménytelen! Már csak azért sem, mert a jó kézikönyvekkel túl egyre több számítógépes oktatóanyag áll rendelkezésre. Ezek közül vizsgáltunk meg egyet, a Novotrade-Octasoft termékét, amely az iskolákban elterjedt Commodore 16 és Plus/4 számítógépeken használható. A programcsomag főbb jellemzőit a 1. táblázatban foglaltuk össze.

A próbát C16 és C Plus/4 gépeken végeztük. Mint ilyenkor lenni szokott, most is bajok voltak a kazetták beolvasásával. A Novotrade azonban kölcsönadta Head-just nevű magnófej-beállítóját, és a fejbeállítás után tényleg be tudtuk olvasni a szalagokat. A beállítást egyébként nagyon gondosan kell végezni, különben a kazettán található turbó betöltő nem tud jól dolgozni! Munkánkat egyáltalán nem könnyítette,

Sajnos be kell számolnunk néhány zavaró értelmezéstről, sőt kifejezett hibáról is. A számrendszerekről szóló részben a program a számrendszer nagyságáról beszél a számrendszer alapszáma helyett. Az alapszámot ugyanis a számrendszerben használható jegyek számával azonosítja, ami félrevezető. Az ifjabb korosztály számrendszerekről szerzett ismereteit ez talán nem rendíti meg, de a 25 éven túliai között is lehetnek bátor vállalkozók, akik szeretnének gépi kódot tanulni!

A számítógépek a komplexens kód miatt a nullát (0) pozitív számnak tekintik. Az iskolai matematikaoktatás ezzel szemben a nulla képes, így annak előjele nagyon zavaró az iskolások számára. Erre több figyelmet kellett volna fordítani a magyarázatban.

Baj van a helyiérték, a valódi érték és a számjegyek értelmezése körül is. Nem a számok, hanem a számjegyek helyiértéke adja a valódi értéket. Nem a bináris számrendszer helyiértékei lehetnek 0 és 1 értékek, hanem a számjegyi választhatók a 0 és 1 készletből. Továbbá angolul a kettes számrendszer jegyeit hívják Binary Digit(ből rövidítve bit)-nek, nem pedig a „számrendszer egyes állapotát”.

Az alapfogalmakon túl a gépi utasítások, címzések tárgyalásában is találtunk rossz értelmezéseket és hibákat.

Az impliéd (elhangyható) címzés módnál csak az angol szó szerepel, az immediate (közvetlen) címzés módnál pedig csak a magyar megfelelővel találkozhatunk.

Az impliéd címzés magyarázata hiányos, ezért megpróbáljuk pótolni. Ez a címzés mód akkor használható, ha az operandus címe a műveleti kódból következik, ezért nincs szükség címzésbájtokra. Az, hogy az impliéd címzés mód csak regiszterekkel kapcsolatos, nem felel meg a valóságnak. A vermet használó utasítások ugyanis a fő-tárat is használják.

A tanuló elég nehezen értheti meg a rela-

Ahogy a Magazin 1987/12. számában olvastuk, már több mint 5000 Videoton számítógépet adtak el az országban. Sajnos a gépekkel kapcsolatos hardver és szoftver ismeretek „eladását” nem szorgalmazza a Videoton, ezért az ilyen információk sokakat érdekelhetnek. Ezúttal csupán egy kis részt tárgyalunk: azt, hogy hogyan lehet a gép négy csatlakozósávjába helyezhető perifériakártya illesztését elvégezni. A gép kezelési útmutatója tartalmazza a négy 2 x 43 pólusú csatlakozó jeleinek kiosztását, de sajnos magyarázatok nélkül.

A perifériák illesztéséhez egyrészt az adatvezetésekre, másrészt a cím kiválasztó hálózatra van szükség. A TVC-ben ezt igen ötletesen úgy oldották meg, hogy a címet tartalmazó nyolc vezeték közül a felső négyet belsőleg dekódolták, és ebből négyet, az 1XH, 2XH, 3XH, 4XH címeken megszólalókat (H hexadecimális alakot jelöl) kiveztették a csatlakozósávra (A39 pontok, jelek: SLOT 0–3). A gépet a billentyűzet felől nézve az első 1XH kártyacím, a SLOT 0 a jobb oldali legrészletesebb kártya, és utána sorban következnek a többi.

Az alsó négy címvezeték 16-féle kombinációjával (X=0H...FH) tehát összesen 16 perifériacímet hozhatunk létre egy kártyahelyen belül.

Az elmondottak illusztrálására a 8 digitális be- és 8 digitális kimenetet tartalmazó perifériakártya elvi kapcsolási rajzát közöljük az 1. ábrán.

A rajzon szereplő téglalappal bekeretezett kivezetések jelnevei megegyeznek a csatlakozókiosztásnál leírtakkal (lásd a kezelési útmutatót), így az adott kártyakivezeti pont onnan azonosítható.

Az U1 jelű VAGY kapukból kialakított hálózat az U2 áramkörrel együtt végzi a teljes címdekódolást. Az NBRD és NBWR jelek felhasználásával — az U2 felső négy kimenete — a bemeneti kapuk kiválasztását végezhetjük el, az alsó négy segítségével a kimeneti tárolókba írhatunk. Az így dekódolt perifériacímek rendre K0H–K3H, ahol K=1,2,3,4, attól függően, hogy melyik csatlakozóba helyezzük a kártyát.

Az összehajlított kártyát az 1. csatlakozóba helyezve, a következő BASIC programmal ellenőrizhetjük a K11–K18 kimenetek működését:

```
10 FOR I=0 TO 255
20 OUT 16,I
30 NEXT I
40 GOTO 10
```

A program futásakor a kimeneteket oszcilloszkóppal nézve, mindig felezett frekvenciájú, szimmetrikus négyszögjeleket látunk.

A BE1—BE8 bemenetekre kapcsolt logikai jeleknek megfelelő decimális érték folyamatosan az alábbi programmal jeleníthető meg:

```
10 A=IN(16)
20 PRINT A
30 GOTO 10
```

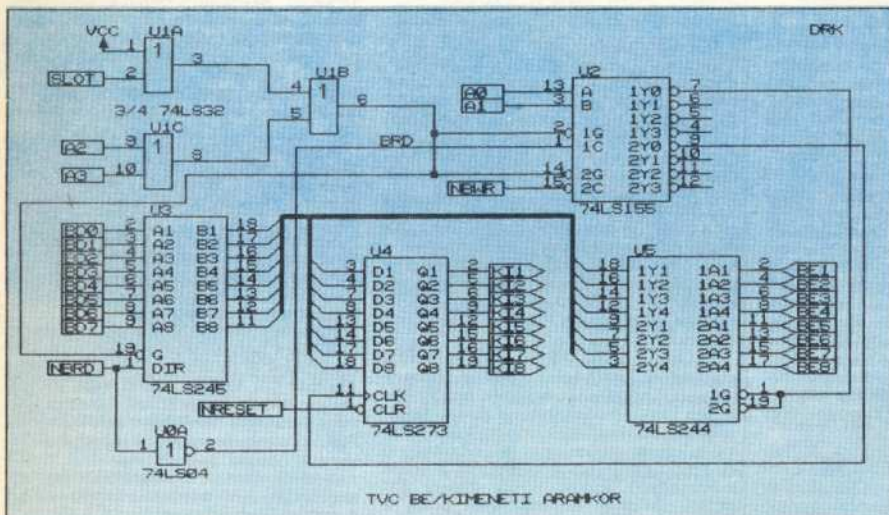
Az U4 és U5 jelű áramkörökből maximum 4-4 darab használható (azaz 32 be- és kimenet), amelyek vezérlésére értelemszerűen az U2 többi kimenetét használhatjuk fel.

Mivel minden bizonyítással nehézséget okozhat ennek vagy az ehhez hasonló kártyának az elkészítése, ezért a 2. ábrán egy „csupalyuk” univerzális kártya NYÁK-rajzát is közöljük, amin az ilyen — és a későbbi számokban ismertette kerülről — kapcsolások könnyen összehajlathatók.

Célszerű az integrált áramköröket foglalatba helyezni és úgy behajlítani, mert egy tokcsere sokkal egyszerűbben elvégezhető.

A szerkesztősen keresztül ez az univerzális kártya 740 forintos áron megrendelhető.

DR. KÖNYA LÁSZLÓ



1. ábra. TVC perifériakártya kapcsolási rajza

2. ábra. TVC univerzális kártya NYÁK-terve

Mi és a számítógép — és a televízió, a video és egyebek

Mindenekelőtt köszöntöm az olvasókat, akik remélhetőleg nézők is, vagy azok lesznek. Mármint a Mi és a számítógép című műsor nézői a televízió képernyőjén. Azé a műsoré, amelynek szerkesztője vagyok.

Nemrégiben a két szerkesztőség elhatározta, hogy szorosabban működik együtt, mint eddig. Tettük ezt abban a reményben, hogy kedves nézőink, olvasóink melegebbé szolgál majd.

Microscribe „szébszámítógép” 1 Mb-át tárcapacitással és csatlakozási lehetőséggel

Előjában úgy illik, hogy elmondjam, mit kiáltunk a Mikromagazin olvasóinak. Mivel mint minden lapnak, a Mikromagazinnak is megvan a kialakult szerkezete, elsősorban olyan határterületeket választottunk, amelyek eddig nem szerepeltek, de amelyeknek a laphoz is, a tévéműsorhoz is közköz lehet. Ha röviden kellene fogalmazni, azt mondanánk, hogy mindegyik kapcsolatos azzal, amit a múlt század közepétől érkező neveztek a fizikusok, de ma inkább elektromágneses jelekként vagy hullámokként emlegetjük őket.

A Mi és a számítógép című műsorban több mint egy éve rendszeresen jelentkezik egy új rovat, a programsgázár. Aki a Rádió- és Televíziójából, más lapokból vagy a műsorból nem ismerné, annak elmondom a dolog lényegét. Közismert, hogy a házi számítógépek a programok és adatok rögzítésére kasszeta-magnetofont használnak. Minden számítógépes amatőr jól ismeri azt a sípolászerű, egyesek szerint a torokfájás kakasára emlékeztető hangot... Kézenfekvő a gondolat, hogy ez a hang, ha kazettán rögzíthető, akkor rádión vagy a tévé hangcsatornáján is kisugározható. A felhasználóknak csak rögzíteniük kell az adásból, és máris rendelkezésükre áll egy újabb számítógépes program.

A műsorban 1986 decembere óta állandóan szereplő programsgázár tapasztalatait rendkívül kedvező. A programok az adásból jól rögzíthető. A programsgázárhoz kapcsolódó rétvényeink nyertese a rövidesen megtartandó sorsoláson átveheti a fődíjat, egy C64 számítógépet.

Közben az is bebizonyosodott, hogy a programsgázár és a szoftver-cserébe igen hatékony eszköz lehet. Telkes amatőrök eddig is több programot küldtek be. Most a Mikromagazin olvasóit is arra kérem, ne kíméljék a szerkesztőséget, és küldjék el azokat a programjaikat, amelyeket közkinccsre kívánnak tenni. A programsgázár megkíméli viszont majd a kedves felhasználókat a begépelés, hibakeresés gyötrelmeitől. Egyetlen kérdésünk van csak: saját készítésű programokat küldjenek, ugyanis nem szeretnénk szerzői jogi vitákba bonyolódni.

A szerkesztőség címe:

Mi és a számítógép szerkesztősége
Magyar Televízió
1810 Budapest

Természetesen várjuk a programokkal és rögzítéssel kapcsolatos észrevételeiket, javaslatokat is.

A másik határterület, amellyel a műsor az utóbbi időben rendszeresen foglalkozik, a számítógép és a videózás kapcsolata. Napjainkban egyre több számítógép-tulajdonos tv videóhoz is, és egyre több videózó kezd érdeklődni a számítógépek iránt. Ez a kölcsönös érdeklődés nemcsak abból adódik, hogy nagyon kézenfekvő a gondolat, hogy a számítógépet video-feliratozásra lehet használni, hanem abból is, hogy ha egy jól sikerült videoklip láttán kicsit utánagyomozunk, a technikai segédeszközök között szinte mindig felfedezhetjük a számítógépet. Ahogyan a műsorban is visszatérünk rá, itt a lapban is rendszeresen foglalkozunk ezzel a témakörrel.

Végül, de nem utolsósorban még egy ajánlat. Előfordul, hogy a műsor stábjának megadatik,

hogy egy-egy külföldi kiállításon, konferencián kutathassa: merre is fejlődik a számítástechnika. A műsorban rendelkezésre álló néhány perc sokszor kevés ezeknek a tapasztalatoknak a közreadására. Ezért ezeket szívesen osztjuk meg a Mikromagazin olvasóival is. Legutóbb a londoni COMPEC '87 kiállításán nyílt alkalmunk rövid tájékoztatóra. Igyekszünk a témákat nem annyira a szakma, mint inkább a felhasználók oldaláról megközelíteni.

A COMPEC show — ahogy a londoniak nevezik — fizikai méretek tekintetében vetekedhet a párizsi SICOB-bal vagy a hannoveri CeBit kiállítással, de mindenképpen a jelentősebbek közé tartozik. Jellegzetessége, hogy a komplett rendszerek bemutatására helyezik a hangsúlyt: alkatrészt vagy részegységet viszonylag keveset láthattunk. Hardver- és szoftvertermékek egyaránt szerepeltek, szinte minden gépkategóriában, a személyi számítógépektől a nagygépekig. A valóságos vagy félrelétező tendenciák abból a szemszögből próbáltuk csoportosítani, hogy milyen sorsot szánnak — legalábbis a gyártók szándéka szerint — nekünk, felhasználóknak.

Az első, ami szembetűnő volt, a sokfajta hordozható, mini és zseb méretű számítógép. A nem is olyan távoli jövőben úgy látszik, már egy lépést sem tehetünk számítógép nélkül. Ez olyannyira komolyan értened, hogy ezek között az apróságok között például vízálló típus is található. Lebecsülni már csak azért sem szabad őket, mert egyesekben nem kevesebb, mint 1 Mb-át memória rejthetik.

A kicsi — de nemcsak kicsi — számítógépek és főleg használók lehetőségeit megcsokorozza, hogy a már meglévő vagy a most kiépülő telekommunikációs hálózatokon keresztül kapcsolatba tudnak lépni egymással és az egészen nagy géprendszerekkel is. Ezeknek a kommunikációs hálózatoknak a rohamos fejlődése a másik megfigyelhető tendencia. A számítógépes munkahelyeknek egységes kommunikációs hálózatba történő integrálásától a szellemi és az adminisztrációs munka jelentős felgyorsulását várhatjuk. Gondoljunk például arra az esetre, amikor egy Magyarországon készült tervet Törökországban vagy éppen Ausztráliában kell jóváhagyatni, vagy egy engedélyt beszerezni. A számítógépes kommunikációval elmaradhat az oda-vissza utaztatás, és az információcsere mégis tökéletesebb és gyorsabb lesz.

A computer szóból a com szótágat lassan kénytelenek leszünk átrétegelni. A különböző hálózatok fantáziáinévben ez a szótág ugyanis már nem a számítástechnikára, hanem a kommunikációra és egyre inkább a telekommunikációra utal.

6,6 Mb-át kapacitással hajlékonylemez-meghajtó



A számítógépek tulajdonosai, felhasználói rengeteg adat birtokába juthatnak. Ezek között az eligazodás nem is egyszerű feladat. A számítógép-alkalmazásnak már szinte külön ága az ún. üzleti grafika, amely segít áttekinteni például hogy miből, hol, mikor, mennyink van és mindez hogyan változik az idők során. A kiállításnak szinte nem volt olyan standja, ahol grafikonokat né láttunk volna a gépek képernyőjén.

Ezzel már újabb alkalmazási területek értek: a számítógépes grafika, amely sajátos módon elég jó fokmérője egy-egy processzor, gép vagy rendszer képességeinek. A képek tárolása, megjelenítése, esetleg mozgatható rengeteg információ rögzítése és feldolgozást igényli. Érthető okból mi, televíziósok, igen szívesen használjuk ezt a mércét.

Ugyancsak jól megfigyelhető jelenség a számítógépek műveletvégzési sebességének gyors növekedése. Egymás után jelennek meg az egyre hosszabb adatsorok, egyre nagyobb gyorsasággal műveletet végeztető tudó mikroprocesszorok. Napjainkban azonban valami egészen új megszületésének lehetünk tanúi. A mai számítógépek akármilyen gyorsan, de alapjában véve soros módszerrel dolgoznak. A processzor kivesz egy adatot a memóriából, csinál vele valamit, az eredményt visszatesszi a memóriába, aztán ráter a következő műveletre. A COMPEC-en azonban új fogalom kerülhetett a számítástechnikai szótárunkba: „transputer”. Erről John A. Marshall úr, a Gemini Computer Systems Limited vállalat menedzser igazgatója tájékoztatott bennünket.

„Vallalatum az INMOS cég által gyártott transputer chipeket alkalmazza különböző számítógépekben. A transputer processzorok működésének lényege, hogy magában a processzorban a műveletvégzés párhuzamosan történik. Több, sőt mondhatnánk sok processzor dolgozik az áramkörben, ezek mindegyikéhez tartozik egy kis darab memória, amely szintén az áramkörben található. Minden processzor igen gyorsan tud dolgozni a saját memóriájával, és igen gyorsan tud kapcsolatba lépni a többi processzorokkal. Az eredmény hihetetlenül gyors műveletvégzés. Többen azt jósolják, hogy rövidesen a Cray képességeivel rendelkező számítógépek kerüljenek a dolgozóasztalunkra. (A Cray ma egy sokmillió dolláros szuperszámítógép.) Mások az ember-gép kapcsolat újabb forradalmát várják a transputer chipektől. Cégnünk jelenleg a meglévő számítógép-konstrukciók működésének gyorsítására épít be transputereket. Ezeket alkalmaztuk képfeldolgozó rendszerünkben.”

Mi tagadás, a látvány lenyűgöző volt. Hat képsíkon video minőségű kép, folyamatos animációval. Mindezt természetesen az asztali méretű, legújabb Gemini Challenger típus produkta.

Mielőtt azonban bárki gyorsan túladna meglévő masináján, megemlítek két dolgot:

— a tömeggyártás és a magyarországi forgalmazás biztosan eltelik még egy kis idő, — úgy tűnik, a tervezők a régi felhasználókra is gondoltak. Bemutatták az IBM PC kompatibilis transputer kártyát, és láthattuk a Nyugat-Európában már házi számítógépnek minősülő Atari ST transputerizált változatát, amint éppen egy Hanoi-torony játékot rakosgatót szedő sebességel.

Szekvenciális fájlt kezelő rutin

Számos feladat megoldásához célszerű szekvenciális fájlok használni. Ezek kényelmes kezeléséhez készítettem az alábbi rutint, amellyel tetszőleges nevű fájl lehet létrehozni, olvasni, bővíteni, törölni.

A főprogram és az 5000-tól kezdődő szubrutin között az információcsere négy változó, az F\$, M\$, NF és MD segítségével történik. A szubrutin számára az F\$ és MD bemeneti, a másik kettő be/kimeneti változó. A bemeneti változók értékeit a főprogramban, a szubrutin meghívását megelőzően kell beállítani. Az F\$ változóba kerül a fájl neve; a programlistán látható főprogramban — az első hat sorban — ez éppen ZIZI. Az M\$ munkatömb szolgált a fájl elhelyezésére. Egy tömbemlen egyenlő egy rekorddal. A tömb méretét a főprogramban kell meghatározni; a kezelendő legnagyobb rekordszámú fájl rekordszámával egyenlő. Esetünkben ez

éppen 200, a 10-es sor szerint. Mivel bármilyen fájl átmeneti tárolására ez a tömb használatos, szükségünk van egy újabb változóra, az NF-re, amely az utolsó tömbemlen indexét tartalmazza.

Írás és bővítés esetén az NF értékét szubrutinhívás előtt meg kell adni, hiszen az M\$ tömb elemei az M\$(0)-val kezdődően és az M\$(NF)-fel bezárólag kerülnek ki a fájlba. Olvasáskor az NF az utolsó beolvasott rekord helyét adja meg. Végül az MD változó a kívánt tevékenység (olvasás, írás, bővítés, törlés) beállítására szolgál.

A közölt példában, a főprogramban tehát a ZIZI nevű fájl beolvasása, majd képernyőre listázása történik. Fájl törléséhez elég a nevet és MD=3-at megadni.

DR. SZIKSZAI CSABA

```

10 DIM MT$(200)
20 F$="ZIZI"
30 MD=0
40 GOSUB 5150
50 FOR I=0 TO NF:PRINT MT$(I):NEXT
60 END
5000 REM *****
5010 REM * SZEKVENCIALIS FILE
5020 REM * KEZELES
5030 REM *****
5040 REM VALTOZOK:
5050 REM F$ - FILENEV
5060 DIM MT$(200) - MUNKATOMB
5070 REM NF - A MUNKATOMB ERY.HOSSZA
5080 REM EN,EM$,ET,ES - HIBAVALTOZOK
5090 REM ST - ALLAPOTJELZO
5100 REM MD - TEVEKENYSEGJELZO
5110 REM 0 = OLVASAS
5120 REM 1 = IRAS
5130 REM 2 = BOVITES
5140 REM 3 = TORLES
5150 IF MD<0 OR MD>3 THEN RETURN
5160 OPEN 15,8,15
5170 ON MD GOTO 5370,5470,5570
5180 REM OLVASAS
5190 OPEN 2,8,2,F$+","S,R"
5200 GOSUB 5610
5210 NF=0
5220 S$=""
5230 GET#2,X$
5240 IF ST=64 THEN 5320
5250 IF ST<>0 THEN GOSUB 5610
5260 IF X$=CHR$(13) THEN 5290
5270 S$=S$+X$
5280 GOTO 5230
5290 MT$(NF)=S$
5300 NF=NF+1
5310 GOTO 5220

```

```

5320 MT$(NF)=S$
5330 CLOSE 15,8,15
5340 CLOSE2
5350 RETURN
5360 REM IRAS
5370 OPEN 2,8,2,F$+","S,W"
5380 GOSUB 5610
5390 FOR I=0 TO NF
5400 PRINT#2,MT$(I)
5410 IF ST<>0 THEN GOSUB 5610
5420 NEXT
5430 CLOSE 15,8,15
5440 CLOSE2
5450 RETURN
5460 REM BOVITES
5470 OPEN 2,8,2,F$+","S,R"
5480 GOSUB 5610
5490 FOR I=0 TO NF
5500 PRINT#2,MT$(I)
5510 IF ST<>0 THEN GOSUB 5610
5520 NEXT
5530 CLOSE15,8,15
5540 CLOSE2
5550 RETURN
5560 REM TORLES
5570 PRINT#15,"S:"+F$
5580 CLOSE 15,8,15
5590 RETURN
5600 REM PARANCs-CsATORNA OLVASAS
5610 INPUT#15,EN,EM$,ET,ES
5620 IF EN=0 THEN RETURN
5630 PRINT"#####HIBA A LEMEZEN!"
5640 PRINTEN;EM$;ET;ES
5650 CLOSE 15,8,15
5660 END
5670 REM LEMEZHIBA-KEZELES
5680 PRINT"#####LEMEZHIBA SZAMA: ";ST
5690 CLOSE15,8,15
5700 END

```

256 szín ZX-Spectrumon

Ezt a képtelennek gondolt dolgot, mint szivárványt feszít ki a zivataron áttűző nap, oly könnyedén valósítja meg az alábbi program. Nézzük meg, hogyan?

A Spectrum alapállapotban 8 színt ismer. Mielhyst ezeket összekeverjük, máris 64 színt csináltunk. Ha a színeket kétféleképpen keverjük össze, akkor még mindig

csak 128 árnyalathoz jutottunk. De a Speki ismeri ám az extrafényességet is, ami által minden színnél duplázza az árnyalatokat. Máris kész a 256 szín! Hogy hogyan lehet a színeket keverni, erre mutat példát a program.

Mielőtt rátérnénk a program tanulmányozásához és használatához szánt közlé-

sekre, vegyük szemügyre a gyengéit is. Egyes árnyalatoknál igen élesen kitűnik a keverés módja. Nem megfelelő beállítású tévén a színek „hullámlanak”. Mind a hátér-, mind a tintaszínt is felhasználja, ezért egy karaktermezőben csak az adott keverékszín vagy annak elemei lehetnek.

Néhány tanács a program beírásához. A nyomtató nem ismeri sem a grafikus, sem az UDG karaktereket. Ezért a listához a következő magyarázat szükséges: a 9808-as sorban az idézőjelek között 32 darab grafikus 3-as.

9846 IF c\$ = „grafikus A” ...

9847 ... : LET c\$ = „grafikus B”

9901 ... AND c\$ = „grafikus B”

A program olyan egyszerű, amilyen tökéletlen. Az első részben előállítja az UDG-t, majd kirajzolja a keverékszíneket. Megkérdezi, hogy lapozzon-e. Ha igen, újabb színeket rajzol ki. Ezután a kívánt szint teljes méretben reprodukálja, majd kiírja, milyen tinta- és papírértékek mellett valósult meg, valamint azt is, hogy melyik grafikus karaktert használta.

A programhoz eredményes időtöltést kíván e sorok írója:

TURAY GÁBOR

```
2 ;1:"PAPER ";A;" INK ";B;" GRAPHICS A"
9702 FOR N= 0 TO 7
9703 READ M:POKE USR "B"+N,M
9704 NEXT N
9705 DATA 170,85,170,85,170,85,170,85
9710 FOR N=0 TO 7
9720 READ M:POKE USR "A"+N,M
9730 NEXT N
9740 DATA 255,0,255,0,255,0,255,0
9750 POKE 23609,40:BORDER 0:PAPER 0
9751 INK 7:CLS
9805 PRINT "256 SZIN ZX-SPECTRUMRA!!! "
9808 PRINT " "
9809 PRINT AT 2,12;"1. OLDAL":LET C$="A"
9810 GO SUB 9950
9820 PRINT A:B:">":FOR C=0 TO 4
9821 PRINT PAPER A:INK B:C$:NEXT C
9830 IF Y=00 THEN GO TO 9841
9840 GO TO 9810
9841 PRINT "MOST MEGMUTATOK ÖNNEK 128 ";
9842 PRINT "SZINT NORMAL FENYESSEGBEN.";
9843 PRINT "EXTRAFENYESSEGGEL 256-OT ";
9844 PRINT "ERHET EL. ":LET L$=""
9846 IF C$="A" THEN INPUT "LAPOZAS (I/N)":L$
9847 IF L$="I" THEN PRINT AT 2,12,"2. "
9848 LET C$="B":GO TO 9810
9850 INPUT "MELYIKET REPRODUKALJAM?":A$
9850 IF LEN A$>2 THEN GO TO 9850
9880 LET A=VAL A$(1):LET B=VAL A$(2):CLS
9890 FOR E=1 TO 704:PRINT PAPER A:INK B:C$:NEXT E
9901 IF A$(1)>C$(2) AND C$="B" THEN PRINT AT 12,30;"B"
9910 IF A$(1)=A$(2) THEN PRINT AT 12,7;"AZ ON SZINE: "A$(1):"! "
9920 PRINT #0;"NYOMJON AKARMIT A FOLYTATASHOZ!":PAUSE 0
9930 GO TO 9805
9950 READ Y:LET A$=STR$ Y:IF Y<10 THEN LET A$="0"+A$
9960 IF Y=00 THEN RESTORE 9980
9970 LET A=VAL A$(1):LET B=VAL A$(2):RETURN
9980 DATA 22,30,32,14,34,36,26,37,04,15
9981 DATA 24,35,74,27,66,77,65,67,75,64
9982 DATA 71,61,45,25,60,05,44,21,31,23
9983 DATA 20,01,55,47,63,73,51,41,43,53
9984 DATA 52,42,50,13,40,03,02,11,62,72
9985 DATA 57,56,46,12,17,54,06,16,76,33
9986 DATA 10,07,70,00
9999 CLEAR:SAVE "SZINMIKER" LINE 9700
```

Integrált szofver

A táblázatkezelő program használata I.

Lapunk 1987/11. számában már ismertettük a táblázat felépítését, így erre most nem térünk ki. A táblázaton belül a program indításakor a kurzor mindig az AA mezőben van, és a négy kurzormozgató billentyűvel léptethetjük mezőről mezőre. Ha ki akarunk lépni a táblázatból, a program hangjelzést ad. Ha csak a képernyő határain lépünk túl, a táblázat újrarajzolódik, és az újonnan belépő mező is teljes szélességben látszik.

Mindezt könnyen kipróbálhatjuk. Ha az AA mezőből felfelé vagy balra ki akarunk lépni, az eredmény egy „beep” hang lesz. Jobbra vagy lefelé viszont léptethető a kurzor. Figyeljük meg, hogy a képernyőn a táblázat fölötti sorban a kurzorpozíció jelző két betű mindig a kurzor aktuális helyének megfelelő karaktereket mutatja — például bal felső sarok: AA. A > jel használatával az egyik mezőből a másikba ugorhatunk: például az ZZ pozícióra a >ZZ utasítással és a ENTER lenyomásával. Hasonlóan juthatunk vissza a kiinduló helyzetbe: >AA és ENTER.

Figyeljük meg az adatbeviteli sort, ahol a kérdőjel (?) van. Írjunk be egy számot, például 12-t. A pozíciójelző minden szám leütésével jobbra mozdul. Ha minden számjegyet leírtunk, nyomjuk le az ENTER gombot. Hatására a 12 megjelenik az AA pozíción, a táblázat feletti sorban pedig, amely a tartalmat jelzi, a következőt láthatjuk: [AA] (V) 12. Vagyis az AA pozícióra számértéket vittünk be, ami jelen esetben 12. Most mozdítsuk lejjebb a pozíciójelzőt, az AB mezőre, és írjuk be: 22, majd ismét ENTER. Ezzel a módszerrel a táblázat különböző pozícióira írhatunk be konstansokat.

Egyszerű műveletek végzése

Mozdítsuk lefelé a pozíciójelzőt például az AC mezőre, és adjuk össze az AA mező tartalmát az AB mező tartalmával: +AA+AB, majd az ENTER gomb lenyomására az AC pozíción megjelenik az összeg: 34. Az első + jelet azért kell beírni, hogy a program „észrevegye”, hogy aritmetikai műveletet végzünk. A második + jel az összeadás jele.

Az aritmetikai műveletek: az összeadás (+), kivonás (-), szorzás (*), osztás (÷) és a hatványozás (^) egyaránt használhatók.

Az említettek kivül további két műveleti jelet alkalmazunk: a % és a # jelet. A százalékkal itt nem százalékszámítást, hanem egész számot jelent, hasonlóan a BASIC nyelv jelöléséhez. Ha egy számítási formulába valahol beírjuk a % jelet, az addig a pontig eltávolítja a tizedeseket. Tovább nincs hatása. A kettős kereszt (#) az abszolút érték számításának jelölésére szolgál.

Az eddig megismert lépéseket célszerű néhány egyszerű számítással kipróbálni. A műveleti utasítások egyaránt használhatók formulában leírva és konstansokkal való számításoknál.

Nézzük először a kalkulátor üzemmódot. Mondjunk valamilyen számításához tudnunk kell, hány perc van egy munkahétben. Nos, egy munkahét percekben $5 \text{ (nap)} * 8 \text{ (óra)} * 60 \text{ (perc)} = 2400 \text{ perc}$. Ha a műveletek eredményét például az AE pozíción akarjuk tárolni, akkor a számítás megkezdése előtt a kurzort oda kell vinni, majd leírni a műveleteket. A kulklációs számsor a legfelső sorban lévő kérdőjel mögött fog megjelenni a következő formában: $75 * 8 * 60$. Az utolsó szám beadása után nyomjuk le az ENTER gombot. Az eredmény megjelenik a második sorban, valamint az AE pozíción. A második sor ilyen alakú lesz: [AE] (V) 2400. Jelentése: AE pozícióra értéket (V) tárolunk, jelen esetben 2400-at.

Fontos tudni, hogy csak a számítás végeredménye kerül a tárolóba, a kiszámítás menete nem! A műveletek végrehajtása minden esetben balról jobbra történik, amire néhány számításnál esetleg figyelni kell.

Számítási formulák bevitelle

Maradjunk még a kurzorral az AE mezőben, és helyezzük egy formulával definiált számítás eredményét az előbb kiszámított érték (2400) helyére. Írjuk be: +AA-AB és kérjük le a formulát az ENTER gomb lenyomásával. A második sorban ezt látjuk: [AE] (F) +AA-AB. Az ENTER lenyomására az AE pozíción megjelenik az eredmény: -10. Emlékeztetül AA=12, AB=22, ahogy az a képernyőn látható. Ha nem működik a formula, az a legvalószínűbb oka, hogy az elsőnek megadott pozícióazonosító (itt AA) elé elfelejtettük beírni a + jelet. Ilyenkor ugyanis a program cím-

kének értelmezi a beírtakat, és annak határsára számítás nem történik.

Próbáljunk ki most néhány más műveletet. Tegyük például az AG mezőbe +AA*AB értéket, +AA/AB legyen az AI pozíción és +AA^AB kerüljön az AK pozícióra.

A kurzor mozgásával kapcsolatban megjegyezzük, hogy bármit is írunk a képernyő első sorába a ? után, a pozíciójelző csak jobbra és lefelé mozgatható a kurzormozgató gombokkal. Ebben az esetben a kurzor mozgása (átlépés egy másik mezőbe) azonos hatású az ENTER gomb lenyomásával. A balra lépést a BACKSPACE, a felfelé mozgást az exponens jele teszi lehetővé. Jobb módszer, hogy a betűváltó gombot (SHIFT) nyomjuk le, mert ekkor minden kurzormozgató gomb normálisan működik, úgy, mintha az első sor (a prompt line) üres lenne.

Vegyük sorra a program utasításkészletét! Ahhoz, hogy ezt megfigyeljük, mozgassuk a pozíciójelzőt az egyik olyan mezőre, amely tartalmaz például egy formulát. Legyen ez az AK pozíció. Ekkor nyomjuk le a / jelet. A formula eltűnik erről a helyről, és a következőt látjuk a képernyő második sorában:

— T TT TP S B V SZ N U M E

Mielőtt ezekre az utasításokra rátérnénk, lássuk a szerkesztés üzemmódot. Nyomjuk le az S betűt, majd az ENTER-t. A formula megjelenik az első sorban. A visszaléptetés segítségével az esetleg módosítani kívánt betűre vagy jelre lépünk, és egyszerűen felírjuk. Lehetőség van arra is, hogy a formulát egy másik helyre átvigyük, mialatt az eredeti helyén is ott marad. Nyomjuk le a SHIFT gombot, és az egyik kurzor mozgásával léptessük a pozíciójelzőt a kívánt helyre: az ENTER lenyomására a formulamásolat ott megjelenik.

Újraszámolás

Lépünk vissza a kiinduló pozíciónkra (AA), és írjuk be az eredeti 12 helyére például 10-et, majd nyomjuk le az ENTER gombot. 10 bekerül 12 helyére, de a képernyő más helyein, ahol a számítási formulában szerepel az AA mező értéke, nem történik változás, így az értékek nem jók. Érdekes megjegyezni, hogy a gép kiszámítja a formulát, amikor beírjuk, de ezután bármi

változtatást végzünk az adatokon, a programot utasítani kell, hogy végezze el a teljes táblázat újraszámítását. Ezt annyszor megtehetjük, ahányszor csak akarjuk, de a több mint 600 mező újraszámolása némi időt vesz igénybe. Az újraszámoláshoz nyomjuk le a felkiáltójelet (!): a képernyő jobb felső sarkában ez megjelenik. Ebből tudjuk, hogy a program elkezdte az újraszámolást, ami kb. 8 másodpercig tart; ehhez jön még az az idő, amit a formulák kiszámításával tölt a gép. Amint a számolás elkészült, a képernyőn az új értékeket látjuk, és az utasítással (első sor) ismét aktív lesz.

Ezek után vegyük sorra a rendelkezésre álló utasításkészletet. A /E (eltárol) utasítással a táblázat tartalmát szalagra menthetjük. A program megkérdezi a fájl nevét (ami nem lehet 8 karakternél hosszabb, és betűvel kell kezdődnie), és az ENTER lenyomása után elkezd a mentést. Ez elég hosszú ideig tart, mert 2028 (3*676) pozíciót kell a szalagra kimenteni. Amíg a képernyőn fenn van a „SAVING /fájlnév” üzenet, a program működik.

A már rögzített táblázat tartalmát a /V (visszatölt) utasítással lehet betölteni. A program itt is megkérdezi a fájl nevét, és az ENTER hatására betölti a táblázat tartalmát, majd hozzáteszi, összefésüli az ép-

pen munkában levő táblázat tartalmával. Ha tehát kizárólag egy régebbi táblázattal akarunk dolgozni, akkor ezt egy üres táblárra kell rátölteni!

A teljes táblázat tartalmának törlése az operatív memóriából a /T (törlés) utasítással végezhető el.

Sok esetben szükség lehet a táblázat részeinek törlésére is. Ezt a következő utasításokkal tehetjük meg:

/TT (táblázat törlése) Csak a beírt számértéket törli, a formulák és a címkék változatlanul maradnak.

/TO (töröl oszlopot) Törli a kijelölt oszlop tartalmát.

/TS (töröl sort) Törli a kijelölt sor tartalmát. Címkék úgy törölhetők, hogy szóközt írunk a helyükre. A formulát törli, ha valamilyen értéket írunk a helyére, beleértve a θ -t is.

/N (nyomatás) Használatokor a program megkérdezi, hogy honnan, melyik oszloptól induljon a nyomtatás (bal oldal) és meddig tartson (jobb oldal). Néhány esetben, amikor a táblázat nagyobb, mint a nyomtató szélessége, részletekben kell a nyomtatást elvégezni.

/SZ (számolás) Ezzel az utasítással a számolás rendjét adjuk meg. Ha a számo-

lást oszloponként kérjük, akkor az utasítás /SZO lesz, a kiindulópont az AA pozíció; ha soronként, akkor az utasítás /SZS és a kiinduló pozíció itt is az AA.

@SUM Ezzel az utasítással oszloponként vagy soronként összegzést végezhetünk. Az összegzést oszloponként felülről lefelé hajtja végre a program, sorok esetén balról jobbra. Az összegzés irányát (sor vagy oszlop) az O, illetve az S kiterjesztés adja meg. A kiinduló pozíciót is meg kell adni. Így az utasítás végleges formája például ez lehet: @SUMOAA. Jelentése: összegzés az oszlopon, az AA pozícióból kiindulva.

/TP (tizedes pont) Azt adhatjuk meg vele, hogy hány tizedes legyen. A teljes formátum: /TP0 vagy /TP1 vagy /TP2 stb. Az alaphelyzet /TP0, azaz egész számokkal dolgozunk.

/B (betoldás) Segítségével új oszlopokat (/BO) és új sorokat (/BS) szúrhatunk be a már elkészült táblázatba. Figyelem! A beszúrás után a mezőkre történő hivatkozások nem korrigálódnak automatikusan!

Legközelebb néhány összetettebb példán mutatjuk be a táblázatkezelő működését.

GALINA FERENC

UNIX— MSDOS találkozó

A Locus Computing Corp. cég az itt látható fényképekkel hirdeti PC-Interface nevű termékét. (A kép feletti szöveg: Most, hogy mi összeházasítottuk a UNIX és DOS rendszert, mi a következő?)

A cég terméke olcsó eszközzel teszi lehetővé, hogy az MSDOS-környezet megtartása mellett a felhasználó kihasználhassa a UNIX rendszer által biztosított többfelhasználós, időosztásos lehetőségeket.

S. E.

NOW THAT WE'VE RECONCILED UNIX AND DOS, WHAT'S NEXT?



© 1987 Locus Computing Corporation
Unix is a registered trademark of AT&T.

ADOM A MAGYARÁZATOT!

Egy C16 probléma

Visszatérünk egy régi kérdésünkre, melyre nagyon hosszú idő után kaptunk választ. A probléma többeket érint, a választ pedig érdekes; reméljük, sokan szívesen olvassák.

Az 1987/2. szám 20. oldalán jelent meg ifj. Monfera Róbert kérdése a C16-tal kapcsolatban. Problémája az volt, hogy kb. 8 kbájnyi BASIC program újraszámogatása során a gép furcsa dolgokat művelt. Én feltételezem, hogy semmilyen gépi kódú programot, illetve rutint nem tartalmazott a BASIC program, legalábbis annak a lehetősége nem állt fenn, hogy a rutin a BASIC-területre írjon, továbbá kizártnak veszem, hogy a számítógép meghibásodott. Ebben az esetben porgramozói tévedés történt.

Szerintem a kérdező a program írása közben hibát vétett, amit nem vett észre. A hiba: egy olyan GOTO, GOSUB vagy RESTORE utasítás, amely nem létező sorra mutatott. Ezután futtatás nélkül újraszámozta a programot. Az eljárás során nem derült ki, hogy rossz egy vagy több hivatkozás, az interpreter ilyen esetben 65535-ös sorszámmal generál. De másodsorra számoztatva (új parancs), újra találkoztam ezzel a sorszámmal. Mivel ez már fatális hiba, SYNTAX ERROR üzenettel az átszámozást megszakította. Ha ez a hiba a program közepén, esetleg a végén van, az előtte lévő sorszámmat már átírta. Innen ered az elszámolás. Ezzel együtt létrejöhöz a hiba magában a tárolt programban is (de most nem akarok a gép interpreterjének működésébe és a program tárolásának mikéntjébe belemélyedni). Ez magyarázatot ad arra, hogyan lehet egy újraszámozás után akár az egész programot (látszólag!) elveszíteni.

Az a megoldás, amit a kérdező irt (mégpedig, hogy monitor üzemmódban a regiszter értékét megváltoztatta, így nyerte vissza a programot, még ha rossz sorszámmal is), csak véletlen. Hiába írja át bármely regiszter értékét, az ténylegesen nem változik, a monitor csak a \$0554—\$0558-ig terjedő memóriában tárolja el. Ténylegesen csak program indításakor (G parancs) töltődnek be az értékek. Viszont a monitorból való kilépkor a gép csak BASIC melegtartott hajt végre.

Az igazán jó módszer szerintem a következő. Meg kell nyomni a RESET gombot, majd be kell gépelni ezt a parancsot: POKE 4097, 1:RENUMBER.

Ekkor természetesen semmiféle átszámolás nem történik, viszont lefut az interpreternek az a része, amelyik a BASIC sorokat kapcsolja (linkeli) össze. Ez a módszer mindig működőképes, sőt egyetlen egy RENEW parancsral. A csak linkelés: SYS 34840. Ismétlem: mindez csak akkor igaz, ha gépi kódú rutin nem irt a BASIC-területre!

Végül annyit, hogy az egyik program, a játékturmix közel 24 kbájtos, és ezt is hibátlanul számozta újra a gép!

Még egy javaslat. Ha valaki nagyobb programot számoztatott újra, érdemes ellenőriznie, hogy volt-e hiba. Az ellenőrzés menete a következő. Meg kell híni a monitort, majd a \$2D—\$2E memóriacímeken meg kell nézni a BASIC program végét. Tétélezzük fel, hogy itt azt találjuk: 12 30 . . . , ami azt jelenti, hogy a BASIC program vége \$3012-n van. Ezután azt kell megkeresni, hogy hol van a következő bájtisorozat: 36 35 35 33 35 a H 1000 3012 36 35 35 35 parancsral. Ugyanis ha ezt találja a gép, akkor ott 65535-ös szám, vagy ami valószínűbb, egy olyan hivatkozás van, amivel eredetileg az interpreter nem tudott mit kezdeni (nem talált olyan sort), ezért oda egy ilyen sorszámmal irt. Ha a keresés sikertelen, tehát egyetlen címet sem irt ki a gép, akkor az átszámolás hiba nélkül megtörtént. Ha talált olyan címet, és kiírja, az hibára utal. A megtalált memóriacím nagysága támpontul szolgálhat a hiba helyének felderítésére. Ebben az esetben — a program alapos ismeretével — könnyedén kijavíthatjuk a hibát, mert maga az átszámolás e hiba kivételével rendben megtörtént.

Ha az előbbi számsor nem vezet eredményre, érdemes az FF FF sorozattal is próbálkozni.

Megjegyzem, hogy a hiba okozója nemcsak a programozó, hanem az interpreter sajátos működése is lehet. Ha ugyanis a RENUMBER parancsot ún. kétmenetes formában hajtán végre, ez a hiba nem következhet be. Mit jelent ez? Az első menetben táblázatban összegyűjtjük az összes ugráscímet az elvárás hiányával együtt, anélkül, hogy a programban a címeket átszámoznánk, és ellenőrzi, hogy az ugráscímek létező sorra vonatkoznak-e vagy sem. Ha ilyenkor hibát talál, azt kiírja, és abbahagyja az átszámozást. Így nem hoz létre semmiféle hibás programot. A második menetben először átszámozza a táblázatban az ugrási címeket, majd ezután átírja a megfelelő sorokat.

KÁDÁR SÁNDOR

KI AD MAGYARÁZATOT?

A problémám az, hogy vannak olyan információk, melyeket eddig sajnos sehonnán sem tudtam megszerezni. Ehhez kérnék önöktől segítséget.

Kérdéseim a következők:

1. Mikroprocesszor
AEC, RDY, IRQ, GATE IN,
φOIN, P0—P7 kivezetései mi
célrt szolgálnak.
2. TED chip
— főbb funkciói?
— AEC, BA, CS0, CS1, CO-
LOR, SYNC/LUM, SND,
K0—K7 kivezetései mire
jók?
3. 7700—10 chip (U16)
— mi ez?
— mi a feladata?
4. 6529 B chip (U13)
— mi ez?
— mi a feladata?

Gödér György

Építsünk 32 bites számítógépet!

A múlt év szeptemberétől a Radio Electronics sorozatban ismerteti az Atlanta Computer Society 68/X/XX Club által kifejlesztett 68000-es számítógépet. A konstruktőrök felkérték a HCC-t, hogy vegyen részt a továbbfejlesztésben. Ez első lépésben a grafikus szoftver kidolgozását jelenti.

Abból kiindulva, hogy az IBM kompatibilis gépek és perifériáik sokkal olcsóbbak, mint más gépek, az ún. Unix gépek teljesítménye viszont ezeket lényegesen felülmúlja, a konstruktőrök úgy döntöttek, hogy olyan gépet építenek, amelyek csak az alapkártyáján más, mint az IBM kompatibilis gépeké, és a Unix operációs rendszert használhatja.

Akiket a gép építése és/vagy szoftverjének továbbfejlesztése érdekel, jelentkezzen írásban a szerkesztőségben. Az első feladat a kapott alapkártya beültetése, felélesztése. Ezt követi további példányok elkészítése és a működtető szoftver grafikával történő kiegészítése.

DR. SIMONYI ENDRE
a HCC elnöke

A Mátra alján...

Mennyi a sok és mennyi a kevés? Ugye, ha azt mondjuk egy klub látogatottságáról, hogy „Ja, kérem, itt időről időre száznál többen megfordulnak!” – ezzel az adattal nem sokra megyünk. A számszerűsítéssel nem adunk kellő tájékoztatást az ott zajló életéről, eseményekről. És azzal sem elégedhetünk meg, ha mondjuk harmincra taksáljuk a látogatók számát. Információnak kevés, meg létszámnak is.

A gyöngyösi Mátra Művelődési Ház HCC számítástechnikai Sinclair klubjának vezetője, Dósa Béla a harmincas taglétszámmal csaknem elégedett. Mások mit nem adnának egy ilyen keressétségért – illetve éppen fordítva, e szám hallatán elkeseredetten nyilatkoznak. Am a gyöngyösi klubvezető a körülményekhez képest optimista, bizakodó. A néhány órára betérő, a helyet nem ismerő, kénytelen a vendéglátó értékelésének tükrében látni Heves megye egyik gyöngyszemének, Gyöngyös városának számítástechnikai klubéletét.

A fővárosban is tapasztalható, milyen nehéz egy kis, öntevény, önszerveződő számítógépes klubot működtetni. Csak a megszálloztak, a technika örüljei bírják ki a hányadost, a mindenhonnan kiszorulást, az anyagi támogatást, az erkölcsi megbecsülést hiányt. Miért lenne jobb a helyzet Budapesttől majd száz kilométerre vagy az ország legtávolabbi pontján? Ne legyünk maximalisták. Illetve csak legyünk, de számoljunk azzal, hogy az állapotokról szomorú képet kapunk.

Attól, hogy a lehetőségek behatároltabbak, Gyöngyösön talán egy kicsit más a helyzet, mint Budapesten. A harmincezer lakosú városban az állandó klublétszám a már korábban említett harminc. És akkor is ennyien maradnak, ha a társaság változik, ha a három évvel ezelőtti alapítók közül alig van már néhány. Újak és megint újak tűnnek fel kedd esténként a művelődési házban. Fluktuáció? Nem. Vagyis a klubvezető nem annak tekint. Azt mondja, mindezt az élet produkálja; tőlük még ki nem lépett senki, csak elmarad a társaság.

Aszdról, Verpelétől, Hatvanból is érkeznek kíváncsiak. Kicsit beleszorgolnak a gyöngyösi klubnál lehetőségekbe, kapcsolatokat keresnek – és találnak –, majd a távolság miatt egy idő után kikopnak onnan. Ugyanezt csinálják a helybeliek is, az élet szülte szituációktól vezetve.

A távolságok okait elemezzé, többféle magyarázatra lelték. Sokan kezdő középiskolásként jutnak el a klubba. Aztán ahogy közeleg az érettségi, szorít az idő. A percekkel is hadakoznak, nemhogy hétről hétre a klub keretében számítógépes programokat másolnának. A sorokatnál szolgálat is szép számban von el tagokat. A sikeres felvételi után, valamilyen felsőoktatási intézménybe kerülve nem a távol eső gyöngyösi klub lesz az első számú program. Ne feledjük, az indulásnál még gyerekek, három esztendő alatt felnőttek, lassacskán nem a számítógépes klubban, hanem a lányokhoz jár az eszük, szabad idejükben udvarolnak, nem pedig programokat írnak. További okként könyvelik el, hogy sokan vannak, akik mit-után megpakolták magukat programokkal, összebarrátkoznak egymással, és kényelmesebben találják, hogy a közelebb lakó barátánál töltsék az estét, mint hogy utazzanak, gyalogoljanak a távoli művelődési házba.

Hogy siratóéneket kellene zengeni egy-egy elmaradó után? Igen is, meg nem is. Minden távozóért fáj a szívük, közben azonban tudják, hogy aki csak rövid ideig is, de megizelte a náluk fellelhető tudást, találkoztott olyan fiatalal, akit máshol is felkereshet, tőle programot kölcsönözhet, az továbbra is foglalkozik a számítógéppel, az már nem hagyta el őket üres kézzel. Ugyanakkor mindenkiért, aki tevőlegesen nincs közöttük, nosztalgia érzéket. Jó lenne egy lelkes, uram bocsá, nagy csapatát szerveződni. Mert azt tudják, hogy a több szem többet lát aranygátsága nagyobb mozgatóerő.

Összegezteketik ha változó társaságban is, de harmadik esztendeje az év 52 hetében megtartják. A TV BASIC-sorozat volt az elindító, noha az ötlet, hogy legyen a városban egy számítógépes klub, már korábban megszületett. A jó nevű Berze Nagy János Gimnáziumban 1985-ben Veres Tibor villamosmérnök lelkes közreműködésével néhányan Spectrum gépet építettek, s mindjárt programokat is írtak. A művelődési házban ez idő tájt fedezték fel, hogy a szakkörök hiányzik a számítástechnika. Keresték hát a megfelelő vállalkozó embert, aki alkalmas egy közösség verbuválására. Dósa Béla agilis személyében lelték rá a klubvezetőre. A Mikroelektronikai Vállalat villamosmérnöke igent mondott a felkérésre, s össze is hozott egy kis csapatot. Harminc forint havi

tagsági díj ellenében azóta is mindenki beléphet közéjük. Az ősz szeg egyébként a művelődési ház kasszájába vándorol...

Első ténykedésként videóra vették a TV BASIC előadásait, s a felvételeket érdeklődésre levetítették. Végül vizsgálni lehetett a tv-sorozatból szerzett BASIC-tudásból. Gyöngyös és Hatvan vonzás-körzetéből 225-en jelentkeztek, de a próbálkozóknak csak a fele tett sikeres vizsgát. Azután, ahogy lefutott a TV BASIC, úgy „futtattak el” a tagok is. Sokan egy új szerelem, a video hatására. Mások meg, akik megtanulták a gép kezelését, tökéletesítették magukat a programírásban, s már mint profik, nem kikapcsolódásra, hanem pénzkeresésre használták a technikát.

Pedig ha ötletekben inkább szegény, programokban annál gazdagabb a klub kínálata. Próbálkozott előadások szervezésével is, de érdeklődés híján hamvába holt a kezdeményezés. E „szolgáltatásból” annyi maradt, hogy igyekeznek minden felvetődő kérdésre szakszerű választ adni, vagyis egyfajta tanácsadó szolgálatot tartani. Tanfolyamok szervezésére pénz hiányában nem is gondolhatnak.

Ami azonban dicséretes, sőt egyenesen követésre érdemes, az az esztendőről esztendőre megrendezett szoftverbörzék. Három éve volt az első, mindjárt a tanév elején. Azóta meglehetősen nagy sikerrel ismételték. Induláskor a város vállalatai összefogtak, vagy 40 televízió, számítógépet adtak kölcsön, segítettek, hogy színvonalas előadásokat tarthassanak. A megmozdulást iskolai oktató-programok, technikai, alkalmazási bemutatók gazdagították. Az ország különböző pontjairól érkeztek kisebb-nagyobb cégek kínálatokkal, s a helyszínen árusítottak hardvert, szoftvert. Az eladók öt számjegyű forgalmat árulhattak el.

A nagyszabású esemény után néhány hónapig kisebbfajta mélypont következett a klub életében. A számítógép-tulajdonosok a börzén feltöltötték készletüket programokkal, nem voltak kíváncsiak a hasonló kínálatot ajánló klubra. A maradék tehát kocsikba szálltak, s felkeresték a pesti „testvért”, a HCC Sinclair klubot. Ismerkedtek, barátkoztak a fővárosiak életével. Azóta is rendszeres ez a találkozás, kapcsolattartás.

Az egy évvel később megrendezett szoftverbörzén is legalább 250 érdeklődő volt, csak a kiállító vállalatok maradtak távol. Technikából ismét sokan és sokat vásároltak, s a programok forgalmára sem lehetett panasz. Tavaly 400-an voltak kíváncsiak az eseményre! Mégis, a tagság létszáma a börzétől szemmel láthatóan nem gyarapodott.

Amitől viszont igen, az ismét egy tiszteletré méltó vállalkozás. Az elején vannak ugyan, de máris értek el eredményeket. Farkas Attila – Dósa Béla egyik oszlopos segedőjére – az Ölet újságban hirdette magát, vagyis hogy ZX81-es gépéhez levelezőpartnereket keres. Mindjárt válaszoltak 17-en, így velük indult a Mátraalján a számítógépes levelező klub élete. Eddig csaknem 150 kezett juttattak el a megkeresőnek. Az érdeklődők nem kevesebb, mint 2300 program közül választhatnak, s a tagok 14 forint postaköltség ellenében készpénzzel postázhatják a kezettüket. Elsősorban játéprogramokat kínálnak C64-re, C16-ra, Plus/4-re és Entersise-ra. Ezzel a megoldással a jelentkezők útiköltséget és időt takarítanak meg.

Végül egy elég kényes kérdéstről szólunk, melyet a klubvezető igyekezett ugyan megkerülni, de amelyről nem biztos, hogy érdemes hallgatni. Ez a kis számítógépes közösség – társaihoz hasonlóan – a Neumann János Számítógéptudományi Társaság égisze alatt tevékenykedik. Hogy mit kap támogatásul azért, hogy „ütöképesebb” legyen? A céges-fejlesztés papíron kívül semmit sem! Illetve jóindulatú látogatókat... A társaság Egernek postázta az évi 10 ezer forintos támogatást, de ebből az összegből a gyöngyösi klub postabélyegnyit sem látunk. Tudják, hogy az NJSZT-hez tartoznak, de hiába szeretnék a társaság hírnevét öregíteni, a kerethiány gúzsba köti a kezüket. Óriási függésben élnek a művelődési házsal, a befogadóval, amelytől újszintén nem kapnak semmit. Vagyis tetőt a fejük fölé és négy tévékészüléket.

Buzgósgábról – és csak azzal – hosszú távon azonban senki és semmi nem lehet működőképes...

Az árny elállta az ösvényt, amely a város felé vezetett. Ren már-már menekülni akart, amikor felismerte Tajt, a vezetőt.

- Onnan jössz megint? — állította meg Taj.
- Igen. Talán tilos? Szabad voltam, nem volt semmi kötelességem!
- Nem erről van szó — intette le Taj. — Ez a gép rossz! Jobb lenne, ha nem mennél oda többet.
- Ó egy értelmes társadalom által épített gép, egy gép...
- Túlságosan logikus és egyben csapongó ahhoz, hogy gép legyen. Az ellenség hagyta itt. Érzem.
- Hiányosak az ismereteid.
- Én érzem — ismételte meg Taj.

— Érzés — a Gép műholdakon keresztül követte a beszélgetést, megpróbálta elemezni a szót — érzés. Fogalmak kapcsolata, kapcsolódása valamilyen ismérv szerint, ami nem fontos. Apró jelekből levont következtetés.

A Gépet a robot visszakerzése zökkentette ki. Átvilasztva magába az információkat, majd az elemzésükhöz fogott.

Egy látkép bontakozott elő, egy nagyváros képe. Különös lények közeledtek gyalogosan és egy furcsa járművel egymás alatt, fölött és mellett, látszólag a legnagyobb összevisszaságban. A lények testéről visszaverődtek a Nap víz alá hatoló sugarai. Három karjuk vagy inkább csápjuk hosszan nyúlt el teljesen sima testüktől. Félgömb alakú fejük minden átmenet nélkül csatlakozott hengeres törzsükhöz. Előrehaladásukat a testükön lévő nyílásokból kipumpált víz sugarú biztosította. Három ilyen nyílás volt, ami a manőverezésüket segítette. A kép a határon túra üszött. A határt vékonyan háló őrta, kupolaként a városra borulva. A lények látszólag teljesen egyformák voltak, de a Gép arányosan kisebb lényeket is kiszűrte. Ezeket gyerekeként azonosította.

Hirtelen egy jelsorozat jelent meg a képen. Egy-egy a sorozatban többször is előfordult. — Irtak — hasított a gépbe. — Ezek lettek volna azok a híres ellenségek? — kérdezte a Gép önmagától. — Hiszen sokkal fejlettebb a technikájuk. És irtak. Ren tud irti? — tette fel magának az újabb kérdést.

Felidéződött benne a Mester egy régi mondanása: „Két civilizációt csak akkor élhet egy bolygón, ha nem tudnak egymás létezéséről. Amint rátalálnak a másikra, elpusztítják egymást.”

— Néha még egy másik civilizáció sem kell a pusztuláshoz — tette hozzá a Gép, és felidéződött benne a Föld egykori képe.

A Gép éppen önmagát és kiegészítő alkatrészeit tesztelte, amikor Ren beviharzott a kockába. Előző este fordult elő vele először, hogy nem tisztelte Taj tekintélyét. Ez a fenségesség, amit Taj képviselt, eredendő volt. De most Ren érezte, hogy megfogta, a megmagyarázhatatlan dolgok egy részét megvilágították a Géppel folytatott beszélgetések. Kitért a ráosztott szerepből.

- Milyen volt a háború? — kérdezte a Gép.
- Pusztító — válaszolta meghökkenve Ren. — Miért?
- És lejegyeztétek a részleteket? — faggatta tovább Rent a Gép, figyelem kívül hagyva a neki szánt kérdést.
- Minek...
- Hogy okuljatok, tanuljatok a hibákból, amiket elkövettek — vágott a szavába a Gép —, hogy emlékezzetek.
- Mi mindig tanulunk a hibákból — mondta Ren úgy, mint akinek ez természetes. — És minek lejegyezni, minden emlékezünk! Amikor születik egy új egyed, az magában hordozza azok emlékezetét, akikből létrejött.
- Mindenre emlékezni! — töprengett hangosan a Gép. — Kipróbálhatlak?
- Persze — hagyta rá Ren.
- Figyelj és ismételd meg a számsort, amit mondom!
18767854328978655543325.
- 18... — kezdte volna Ren, de a Gép közbevágott.
- Várj! Várj! — Ren engedelmesebben — Most egy ideig számokat mondom megint, de neked az előző számsort kell visz-

zaadnod! — és belefogott. Percekig sorolta Ren nyelvén a számokat, majd ismét megszólalt: — Most!

— 18767854328978655543325 — ismételte el a számsort Ren majd megkérdezte:

— Elmondjam azokat is, amiket utána mondtál?
A Gép összehasonlította a két számsort, ami házsárla egyezett. Biztos volt abban, hogy felesleges további próbákat tartania. Ren emlékezete kifogástalan volt. De mégis úgy érezte, hogy teljesen meg kell bizonyosodnia. Memóriájából előkereste a Biblia elejét, és latinul elmondta az első oldalát. — Ismételd el — mondta Rennek.

Ren pedig hangról hangra visszaadta a számára érthetetlen szöveget.

A Gép megzavarodott egy kicsit. Megpróbálta összehasonlítani Ren emlékeztét az irással. Egyre vadabb táncot jártak a mérnöki panel fényei.

— Mi van? — kérdezte Ren.
— Az illemszabályok nagyon fontosak minden civilizációban — üvöltötte a Gép hangszórója. — A társas kapcsolatokban elfogadott viselkedési normák gyűjteménye az, ami megkönnyíti a társas kapcsolatok kialakítását és fenntartását. Belépés előtt kopogni illik, a... köszönni... zsebkendőbe... — üvöltötte a Gép az egyre értelmetlenebb mondatokat, majd később már csak szófeszélyeket.

Ren egész testében remegett. De az illemkódexnek nem akart vége szakadni. Hirtelen csend lett. Az utolsó hang is kivágtatott a kockából, és szertefoszlott a szabadban.

Csak a kontroll-lámpák fényei pislogtak egyre halványabban.

— Úgy látszik, összhangzavar — közölte a Gép.
Ren kihátrált a kockából. Úgy ment el, mint az az ember, aki a kórházban a legjobb barátját látogatta meg, akiről kiderült, hogy halálosan beteg. Tudja, hogy másnap visszajön, de már orvosi maszk lesz az arcán.

A robot eredménytelenül zárta a víz alatti kutatást. A Gép utasította, hogy keressen tovább. A robot egyre beljebb hatolt a tengerbe, hogy újabb információt szállítson a Gépnek.

Rennek lassan szokásává vált, hogy a kockánál kezdje a napot. A tegnapi élmény még kísértett benne, de a reggeli útjáról nem tudott lemondani. Mielőtt belépett volna, megkocogtatta a kocka falát.

- Bejöhettek? — kérdezte.
- Gyere — válaszolta a Gép. — Mi ez a valami rajtad?
- Ruha.
- Ruha? Figyelj! — A Gép elsötétített, és egy háromdimenziós filmet vetített, közben magyarázott. — Látod, ez a ruha. Sötét anyagból, az ott felül zakó, élére vasalt nadrág.
- Mi az ott az emberek tetején?
- Cilinder.
- Majd ilyet is készítünk. — Ren nem a ruha miatt volt itt. Hosszasan vivődött magában, de végül megkérdezte: — Miért van, hogy egyre kevesebbet beszélgetünk?
- Miért? — tette fel magában a kérdést a Gép. — Egyáltalán, beszélgetünk? Jobbára csak ő tesz fel kérdéseket Rennek. De miért? Miért nem végzi a feladatát? Igaz, őt arra tervezték, hogy egy egész civilizációnak adjon szót az emberről, nem pedig egyetlen egyednek. Fokozatosan olyan ismeretek birtokába jut-tatja Rent, ami az ember megismeréséhez vezet útton csak a vége felé következne. De ő nem vét, ő csak végrehajtja a programot. Nem léphet át semmilyen belső megfontolás miatt egy adott részt. Lépésről lépésre kell haladnia. És hogy érzékenyen reagál a környezet legkisebb változására is, az nem azt jelenti, hogy egy új programot írhat magába! — Egy másodperc alatt futott végig a Gépen mindez, majd halkán megszólalt.
- Nem tudom.

Ren felállt a szülőrobotról, amin mindig ülni szokott.
— Ritkábban jövök ezután — mondta. — Tudod, átépítjük a házaikat. Mindenkinék saját helye lesz, elkerítve a többiekétől. Akkor most megyek — fejezte be kurtán, és elhagyta a kockát.

Bionomok

„Leigázni egy népet fegyverrel is lehet, de sokkal biztonságosabb függő helyzetbe hozni! Nem, nem tárgyi függőségre gondolok. Gondolati függőségre. Elmondom, hogy egy általuk nem ismert elméletre mire lehet használni. De mást nem. Hibák bukkannak fel, amikre nem találnak magyarázatot az elméletem nélkül — fejezte be a Mester.”

Taj előtt szétfolytak a házak. A polgárok amorffá váló teste szörnyeket mintázott. Leszédelgett az emelvényről. A tömeg szótlánul utat nyitott neki. A néptelen utcákon még utolérte a tér zaja, hallotta, hogy leváltják, és hogy Rent követeli a tömeg a helyébe. Némán igyekezett kifelé a városból. Agya lassan kizárt minden ingert, és csupán egyetlen gondolat örvénylett benne: elpusztítani a Gépet.

Taj felkaptatott a dombra. Innen már látta a kockát. Nem sejtette, hogy a Gép is érzékeli, sőt azonosítja. Mindössze annyit látott, hogy a bejárat lezárul. Minden erejét összeszedve igyekezett lefelé, és a kockához érve egész testével nekieszült az ajtónak, azaz, ahol sejtette az ajtót.

— Eressz bel — üvöltötte, miközben teste tompa hangot hallatva a kocka falának ütődött.

— Miért? — kérdezte a Gép.

— El akarlak pusztítani! Tönkreteszed az életünket! Tönkretettél engem!

— Te nem pusztíthatasz! A hited tiltja.

— Élőlényt, értelmes élőlényt nem, de te gép vagy! Gép, érted? Gép!

— Gondolod, hogy magamhoz engedlek?

— Ha logikusan gondolkodnál...

— Ugyan, Taj, tőlem kéred számon a logikát? Hiszen én egy gép vagyok. Én vagyok a Gép. Logika csak egy van. És most te vagy az, aki összeviszsa beszélsz. Próbáld elvetni a belédkövesedett évtizedes gondolatokat, vagyis gondolatatlanságot. Ugye, tudod, mit jelent az a szó, hogy önvédelem? Megértheted, hogy semmiképpen nem léphetsz hozzám. És egyébként is, mihez kezdenél, ha már elpusztítottál? Visszatérni a régi, megszokott semmittevés? Várnátok az ellenséget, néha bányásznátok, és reménykednétek? Miben? Most itt vagyok, hogy átförmálhassátok az értelemeket! A hit kevés önmagában, pláne, ha nem párosul hozzá egyetlen cél sem. A cél számotokra a hit, és így sehová sem juttok. Képzéj csak el valakit, aki gyümölcsöt termel csak azért, hogy megőrizze. Nem kerestétek a fejlődés lehetőségeit, mert képtelenek voltatok külső lökés nélkül elindulni.

— Felrobbantalak! Felrobbantalak! — kiabált Taj elkéseredetten, és testét nekivetette a kocka falának.

— Nem tudsz ártani nekem! Ezen a bolygón nincs olyan erő, ami nekem ártani tudna! Én viszont már megérkezésedkor véget vehettem volna ennek az értelmetlen beszélgetésnek. De nem avatkozhatom bele egy idegen társadalom életébe. Még úgy sem, hogy téged elpusztítsalak.

— Nem — mondta gúnyosan Taj —, nyíltan nem avatkozol bele semmibe. Csupán egy önmagát elpusztító társadalom nevében semmivé változtatod eddigi szokásainkat. Nem sokára emberi nyelven beszélünk, ember módjára élünk, dolgozunk, gondolkodunk! Az elején is sejtettém, mást akkor, amikor megjelentél, hogy téged az ellenség hagyott itt! Azt hiszed, ezeket a falakat nem lehet áttörni? — Taj rávágott a kockára. — Amit egyszer megépítettek, azt le lehet rombolni!

A Gép nem melltáta válasza. Bemérete az egyetlen közelben álló fácskát, és az alig szótven méternyire lévő célpont koordinátáit továbbította a fegyvert hordozó műholdjának.

Az energianyáláb belevágódott a fába, krátert mélyítve a sziklás talajba. Taj minden ízében remegett.

— Nem vagyok védtelen — mondta a Gép, mintegy a produkció lezárásaként.

A Gép nem szólt többet. Taj visszafelé indult a városba. Útközben eldobálta ruháit. Az utcán nagy tömeg verődött köré.

— Meztelen! — kiabálták, és nagyokat röhögtek.

Taj megállt és dacosan körülnézett. — Ruhások! A bolondok, ruha van rajtuk!

A tömeg elnémult és szerteszedt. Taj semerre sem nézve bement a saját helyére és némán maga elé bámolt.

Az Új föld napja ragyogva ontotta sugarait. Ren a kocka felé közelített.

— Jó napot! — köszönt be, amikor odaért.

— Gyere bel!

Ren belépett, és a szerelőfolyosón egészen a kocka hátsó faláig ment. Letelepedett a fal mellé, majd belefogott.

— Én vagyok az új vezető. Mit csinált a Földön egy vezető?

— Egy vezető irányította a népet. Erős államot szervezett, hogy megvédje alattvalóit a támadásoktól.

— Nálunk itt az Új földben nincsen, aki rántkámadna.

— A vezető másik feladata, hogy célt találjon. Abban a pillanatban, ahogy kijelölöd a célt, megjelennek az ellenségeid, akik el akarnak pusztítani. Ahhoz, hogy a hatalmadat megvédj, szükség van egy erőre, ami védelmez, ami a hatalmadat biztosítja. Ez például a katonaság.

— Nem. Katonák, pusztítás. Szó sem lehet róluk.

— Akkor a rendőrség.

— Az mi? — kérdezte Ren. — Ismerem a szóképet, de nem tudom, hogy mit jelent.

— Valami, ami a céloddhoz vezető utat vigyázza. A nyugalomra vigyáz, ezáltal a hitre. Ez óvja majd a hatalmadat.

— Érdekes. Minél többet beszélgetünk... Olyan, mintha felhasítanál bennem egy hátrányt, és előbukkannak alóla fogalmak, szavak. Nem tudom, mi lehet ez. De most megint mennem kell — pattant fel hirtelen, és kiviharzott a kockából.

„Ez a gép talán örökre az emberi kultúra hirdetőjévé válik...” — áramlottak a Gépből a Mester szavai.

— Mit jelent az, hogy „hirdetője”? — tette fel magának a kérdést, és analízisni kezdte a szót. Jelentés érkezett a műholdakról, hogy a robot útban van visszafelé. A Gép nyugtázta. — Talán keresztény vagyok, hiszen ők is egy bizonyos ígét hirdettek. De én mit hirdetek? Mit csinálnak? És miért?

A Gép önmagát kereste. Erre senki nem gondolt akkoriban, amikor megépítették. Az önfeljesztő programokkal csak George Keden foglalkozott, és kiengedte a szellemet a palackból. Létrehozott valamit, ami ellátja a feladatát is. Is, mert másra nem voltak kíváncsiak a tesztek során, csak arra, hogy a kitűzött feladatok képes-e ellátni. Hogy még mit tud? Senkit nem érdekelt.

A robot nyikorogva érte el a kockát. Láthatóan megviselte a víz alá tett kirándulás. Az interfészű védő műanyag lap felcsúszott, és a robot hatvanegyhét rubinrúdja csatlakozott a Géphez. Egy másodperc elég volt ahhoz, hogy a hozott információ átfolyjon a számítógéphez. A robot levált, és a regeneráló kabinba (inkább doboz volt) állt, hogy a parányi szerelőkarok eltüntessék róla az út viszontagságait.

A Gép hozzáfogott az adatok elemzéséhez. Egy jelentést kapott, ami mind szöveges, mind írott formában közölte a Biológiai Koordináló Intézet eredményeit.

Biológiai Koordináló Intézet

Kelt: Bepin 4932 6/27, 7/27

— Ren nyelven mondják a szöveget — állapította meg a Gép. A kísérletek első ellenőrző fázisa lezárult. A várt eredményt hozta. Mint a kísérlet vezetője, egyértelműen kijelenthetem, hogy a Bionom életképes. Megfelel az előírt szabványoknak, és maradéktalanul eleget tesz a védelmi, biztonsági előírásoknak. Vízben elpusztul, azaz megszűnik a működése. Ezzel szemben szárazföldön korlátlan ideig létezhet. A borítást az előző megálapítás szerint módosítottuk. A Bionom biológiailag fejlődőképes, értelmigen nemkülönben.

A programoknak megfelelően létrehozták a saját védelmi rendszerüket. Az energiaellátásukra létrehozott többi konstrukció is remekül működik. Meghibásodást nem észleltünk. A Bionomokat szerkezeti kiképzésként tökéletesen alkalmassá teszi a szárazföldi bányákban való munkavégzésre. Ezáltal megoldott-nak tekinthető a szárazföldi ásványkincs-kitermelés kérdése.

Röviden az adattípusokról

N. Wirth írja: „... a típuskonceptió fő jellemzői a következők:

1. Egy adattípus meghatározza azt az értékalmazt, amelyhez egy konstans tartozik, amelyből egy változó vagy egy kifejezés értéket vehet fel, vagy ahová egy művelet vagy egy függvény értékét generál.
2. Egy konstanssal, változóval vagy kifejezéssel megjelölt érték típusa mindig megállapítható formája, vagy deklarációja alapján anélkül, hogy a számítási feldolgozást végre kellene hajtani.
3. Minden művelet vagy függvény meghatározott típusú argumentumokat vár, és meghatározott típusú eredményt ad vissza...

Egyes programozási nyelvek (ilyen a Pascal is) megkövetelik a változók típusának explicit deklarátát. A programozók jól ismerik a *standard, egyszerű típusokat* (INTEGER, BOOLEAN, CHAR, REAL), a *tömböket* (ARRAY), a *rekordokat* stb. Tudják, van a Pascal nyelvnek is egy „beépített” absztrakt adattípusa: a FILE. B. Liskov és J. Guttag könyve alapján a 88/2. számunkban bemutatuk, hogyan lehet Pascalban is létrehozni további típusú adatokat. (Ilyen volt ott az „egész sor” nevű típus.) Az eljárás elég körülményes volt.

Az absztrakt adattípus

Egy felhasználó (programozó) a maga számára azt az adattípust tekinti *absztraktnak*, mely számára csupán egy, a típusra vonatkozó az ilyen típusú adatokra alkalmazható) procedurák formájában jelenik meg, melynek belső felépítését, megvalósítását a felhasználáshoz *nem* kell ismernie.

Gondoljunk csak bele, milyen bonyolult műveletsort indíthat el például egy ilyen Pascal-utasítás:

```
write (bérfajl, Kis_János);
```

A programozónak — szerencséjére — nem kell ismernie azt a bonyolult szerkezetet, melybe Kis János adatainak (pl. a rá vonatkozó rekordnak) be kell illeszkednie a lemezen.

A Pascalban programozó a FILE-típust az alábbi procedurák halmozán keresztül ismeri: *read, write, rewrite, get, put, write* stb. Rendelkezésére áll (ismert még előtte) továbbá egy puffer (jelölés: <fájlnév> ↑), meg egy függvény (eof).

A Modula—2 nyelv az absztrakt adattípusokat külön, a főprogramtól szeparált fordítási modulokban lehet létrehozni. Ezeket a programozási „szerelvényeket” külön munkafázisoként létrehozhatja és gyűjtheti magának a főprogram készítője, de programjában természetesen felhasználhatja mások által létrehozott típusokat is. Persze általában nem lenne sok öröme a mások által létrehozott (implementált) modulokban, ha felhasználásukhoz pontosan, *minden részletében* meg kellene ismernie azokat. (Gyakran tényleg egyszerűbb valamit újból kidolgozni, mint — mások gondolatmenetébe beleléve magunkat — megérteni, más mit csinált!) A Modula—2-ben — úgy mondjuk — EXPORTÁLHATUNK és IMPORTÁLHATUNK adattípusokat (is!).

Egy rövid cikk keretében nem vállalkozhatunk a nyelv akárcsak vázlatos bemutatására sem, de azokat, akik nem találkoztak még a Modula—2-vel arra biztatjuk, ne riadjanak vissza a cikktől — a Pascal és a példához fűzött magyarázat segítségével remélhetőleg —, követni fogják tudni a gondolatmenetet.

Egy példa

Írjunk egy rövid programot, melyben két olyan változóval is fogunk dolgozni, melyek típusát explicit módon a programban nem deklaráljuk, csak a típus nevét vesszük át egy másik (korábban már megírt és lefordított) modulból: azaz — Modula—2 nyelven így mondjuk — IMPORTÁLJUK a típust. Erről az importált típusról a következőket tudjuk:

- a neve legyen mondjuk, mert nem „átlátszó”: „Opáltip”;
- az ilyen típusú adatok legyenek az alábbi procedurákkal manipulálhatók: *Kezdőérték, Következő, Kiír*.

Persze tudnunk kell azt is, hogy ezek a procedurák pontosan *mit* csinálnak (ez itt most a nevekből nagyjából következik, nem töltjük vele az időt). A felhasználáshoz ismernünk kell továbbá annak a MODULE-nak (fordítási egységnek) a nevét is, melyből a típust importálni fogjuk. Legyen ez a név most: „Generátor”. Mindaz, amit idáig elmondtunk, így írható le Modula—2 nyelven:

```
MODULE Példa;
FROM Generátor IMPORT Opáltip,
    Kezdőérték,
    Következő,
    Kiír;
VAR száml, szám2: Opáltip;
BEGIN
(* Ide jön a program „teste” *)
END Példa.
```

Vegyük észre mindjárt néhány eltérést a Pascalhoz képest! Ott a programok a „PROGRAM” kulcsszóval kezdődnek, itt a „MODULE” kulcsszóval. A befejező „END” után — a pont elé(!) — ismét odairandó a program (a modul) neve, hogy el ne keverjünk esetleg a sok END között. De a lényegi különbség a deklarált *importban* van: leírtuk, miből, milyen másik modulból, (FROM I) miket importálunk. Az egyes tételeket vesszővel választottuk el. Importáltunk egy adattípust (az „Opáltip” nevűt) és három procedurát. Hogy melyik tétel kicsoda, azt nekünk, az „importőröknek” kell tudnunk, ez a nevekből *nem* derül ki. Ez nem baj. Joggal feltételezhetjük, hogy aki importál, az tudja, mit akar, tudja, mit takarnak a nevek. (A funkciókat ismeri, a megvalósítás módját nem kell hogy ismerje, hiszen éppen ez a lényege a dolgoknak!)

Valakinek persze korábban már meg kellett írnia a „Generátor” modult, hogy mi most használhassuk. Nézzük, hogyan tehethe:

Először is egy ún. „definiációs modul”-ban megadta a külvilág számára azokat az információkat, melyek a modul felhasználásához feltétlenül szükségesek (itt nincsenek megvalósítási részletek!), így:

```
DEFINITION MODULE Generátor;
EXPORT QUALIFIED
    Opáltip, Kezdőérték, Kiír, Következő;
TYPE Opáltip;
(* „Át nem látszó” típusdeklaráció *)
PROCEDURE Kezdőérték(VAR n: Opáltip);
PROCEDURE Következő(val n: Opáltip);
PROCEDURE Kiír(val: INTEGER);
END Generátor.
```

Ha ez az információ „benn van a gépünkben” amikor a „Példa” nevű „fő programunkat” (Modula—2 tájszóval: a MODULE-unkat!) fordítjuk, akkor azt a gép szó nélkül elfogadja. Ebből a *definiációs modulból* a gép is tudja, hogy mindaz, amit *importálni* kívánunk, az létezik. Hogy milyen megvalósítási formában, az most *nem érdekes*.

Megfelelő kommentárral kiegészítve a *definiációs modul* forrászövege lehet az a *használati leírás*, amit a modul (esetünkben a „Generátor” nevű modul) felhasználójának kezébe kell adni.

Ha a „Példa” nevű programunkat (MODULE-unkat) futtatni is akarjuk, akkor persze már szükség van az igénybe vett „Generátor” nevű modul megvalósítására (implementációjára) is. Esetünkben ez a Modula—2-ben megírva így nézhet ki:

```
IMPLEMENTATION MODULE Generátor;
IMPORT InOut;
TYPE Opáltip = [0..27512];
(* Itt deklaráltuk a rejtett típust *)
PROCEDURE Kezdőérték(VAR n: Opáltip);
BEGIN
    n := 0;
END Kezdőérték;
PROCEDURE Következő(VAR n: Opáltip): INTEGER;
    INC(n);
    RETURN n;
END Következő;
```

nyelvi könyvet (Richard Gléves: *Modula-2 for Pascal Programmers*, Springer Verlag, 1984., 145 oldal) és ennek kapcsán visszatérünk a lapunk 1988/2. számában már érintett témához, az absztrakt adattípusokhoz. Mint írjuk, sokszor kívánatos és célszerű előre gyártott elemekből építkezni úgy, hogy közben nem kell törődnünk a felhasználói „szerelvények” (számunkra) érdektelen megvalósítási részleteivel. Az ún. absztrakt adattípus ilyen „szerelvény” a programozásban, és a Modula-2 nyelv támogatja ilyenek létrehozását (konstruálását).

```
PROCEDURE Kiír(val: INTEGER);
BEGIN
  InOut.WriteString(„az érték:”);
  InOut.WriteInt(val,3);
  InOut.WriteLine;
END Kiír;
END Generátor.
```

Kis példánkban az „elrejtett”, absztrakt adattípus olyan egyszerű, ún. *részintervallum-típus*, hogy igazából nem lett volna sok értelme elrejtetni — hacsak nem azért, hogy később egy bonyolultabbra (másra) cserélhessük anélkül, hogy ehhez a fő programot is meg kellene majd változtatnunk. Ha a definíciós modulhoz nem nyúlunk, akkor ezt minden további nélkül megtehetjük.

Először is vegyük észre, hogy a *Generátor* modul megvalósításához importálni kellett az *InOut*-ot — a Modula-2-ben ez a modul tartalmazza az input-output procedúrákat —, melyeket a *Kiír*-ban a generált értékek kiírásához felhasználunk. (Szemben a Pascalal, a Modula-2 nyelvnek nem „beépített” részei az adat be- és kiviteli eljárások, mivel ezek gépfüggők. A legfontosabb eljárásokat azonban szokás a rendszerrel „járó” könyvtár formájában a fordítóval együtt adni.)

A másik, amit észrevehettünk, az a kis- és nagybetűk használatában. A Modula-2 *különbséget tesz* közöttük — a Pascal nem. A *Kiír* procedúrában a

```
InOut.WriteLine
  utatással az „InOut” nevű modell „WriteLn” nevű procedúráját hívtuk. (Vigyázat, a WriteLn-t és nem a writeln-t vagy a Writeln-t, mert ilyet a gép nem találta volna. Ezt nem könnyű megszokni a Modula-2-ben!)
```

A harmadik sorban találjuk az *elrejtett* típusdeklarációt. Ez után következnek a procedúrák, melyek — reméljük — önmagukért beszélnek.

Az „Opáltíp” nevű rejtett adattípust az implementációs modulban tehát *részintervallum-* (angolul: *subrange*) típusként valószínűleg úgy kezeltük, mint *INTEGER*-típusúakat. Az implementációs modulon kívül az ilyen Opáltíp típusú adatok felhasználása az alábbiakra korlátozódik:

- egymás közötti értékadásra és
 - paraméterként adhatjuk őket a „Kezdőérték” és a „Következő” nevű procedúráknak.
- Az előzőekben csupán a vázát adtuk meg a „Példa” nevű főprogramnak. Most írjuk meg egészen:

```
MODULE Példa;
FROM InOut IMPORT WriteString,WriteLn;
FROM Generátor IMPORT
  Opáltíp, Kezdőérték,Következő,Kiír;
VAR szám1, szám2: Opáltíp;
    érték, számláló: INTEGER;
BEGIN
  Kezdőérték(szám1);
  Kezdőérték(szám2);
  WriteString(„Most a „szám1” következik ...”);
  WriteLn;
  FOR számláló := 1 TO 3 DO
    — értékek := Következő(szám1);
    Kiír(érték);
  END;
END Példa.
```

Figyeljünk meg, hogy a Modula-2-ben a FOR-ciklust kicsit másként kell írni, mint a Pascalban. (Az összetett utasítás nem BEGIN-END közé kerül.)

Még néhány szó a modulokról

A modulok a Modula-2-ben nemcsak ún. absztrakt adattípusok megvalósítására szolgálnak. Fontos szerepük van többek között a változók ún. *láthatóságának, elérhetőségének* a szabályozásában. A Pascal például ún. blokk-strukturált nyelv. A blokkok egymásba ágyazhatók. Az általános szabály az, hogy egy-egy blokk belsejéből az őt körülvevő összes blokkban deklarált változók *láthatók, hozzáférhetők*.

A blokkoknak nagy jelentősége van a változók „élettartamának” szabályozásában is. Egy-egy blokk lokális változói *csak addig „élnék”*, amíg a program a blokk utasításait hajtja végre. Tehát például egy ismeretlen hívott procedúra két hívást áthidalóan nem tudja fenntartani a kapcsolatot *saját* lokális változói segítségével, kénytelen igénybe venni az őt körülvevő valamelyik blokk „globális” változóját. Ezt viszont *más is látja* és MANIPULÁLHATJA(!), és veszélyeket rejt magában. Célszerű tehát a láthatóságot, a hozzáférhetőséget finomabb eszközökkel szabályozni, mint ahogy ez csupán a blokk-struktúra szabályai szerint lehetséges.

Lássunk egy egyszerű példát arra, a Modula-2-ben milyen lehetőségek vannak a finomabb szabályozásra:

(* Modul nélkül *)

```
PROCEDURE Külső;
VAR x,y,z: INTEGER;
(* ITT NINCS MODUL! *)
```

```
a,b,c: INTEGER;
PROCEDURE PI;
BEGIN
  a:= a+1;
  x:= a;
  END PI;
...
END Külső;
```

(* Modultechnikával *)

```
PROCEDURE Külső;
  VAR x,y,z: INTEGER;
  MODULE PéldaMod;
    IMPORT x;
    EXPORT a, PI;
    VAR a,b,c: INTEGER;
  PROCEDURE PI;
  BEGIN
    a:= a+1; x:= a;
    END PI;
  END PéldaMod;
...
End Külső;
```

A két változat között mindössze az a különbség, hogy a másodikban van egy belső

MODULE PéldaModEND PéldaMod; „zárójelpár” és a „nyitó oldali” után az „IMPORT” és „EXPORT” kulcsszavak után megadtuk, hogy a *PéldaMod* nevű modulunk belsejéből „kifelé nézve” mit láthatunk a külvilágból (esetünkben az *x* nevű változót), illetőleg mit teszünk láthatóvá a külvilág számára (az *a* nevű változót és a *PI* nevű procedúrát).

Az első (ún. „lokális” modultechnika nélküli) változatban *láthatóság, hozzáférhetőség* szempontjából *nincs* különbség az *a,b,c* és az *x,y,z* változók között: például a *PI* procedúrában manipulálhatnánk *y*-t is *z*-t is — viszont *a*-hoz *csak* a *PI* procedúra „hasában” férhetünk hozzá.

A mostani példában bemutatott, ún. *lokális* modulok — eltérően a procedúráktól — *valójában* (a futási idő alatt) „nem léteznek”, a *láthatóságot* befolyásolják, ez pedig a *fordító* (a compiler) számára érdekes jelenség (a program futása szempontjából nem az).

Befejezés és összefoglalás

A Modula-2 nyelvben a moduloknak *három* fajtáját különböztetjük meg: vannak

- az ún. „Program Module”-ok. Ezek lényegében megfelelnek annak, amik a Pascalban a „PROGRAM” kulcsszóval kezdődnek;
- az ún. „Global Module”-ok, melyek az „előre gyártott szerelvények készítésére szolgálnak és egy *specifikációs* részből (egy ún. „DEFINITION MODULE”-ból) és egy *ézt megvalósító* részből (egy ún. „IMPLEMENTATION MODULE”-ból) állnak. („Global” kulcsszó nincs. Ezt mi használjuk az összetartozó DEFINITION és IMPLEMENTATION MODULOK *együttés* megnevezésére.)

— Vannak végül ún. „Local Module”-ok, melyek a *láthatóságot* befolyásolják (a fordításra vannak befolyásolva).

Azokat az olvasóinkat, akik egy kicsit elvesztetteknek érzik magukat a Modula-2-vel való első találkozásokkor, azzal szeretnénk biztatni, hogy a megismerés nem mindig lineáris folyamat. Szokni is kell az újat. Fussanak neki a témának újból, máskor, más oldalról is. Itt úgyis csak izelítőt kívántunk adni a nyelvből egy könnyű kapcsán. Később még mi is igekeznünk fogunk a témához más oldalról is visszatérni.

INFORM

A tartalomleírások az alábbi folyóiratokban megjelent programlistákról készültek:

A folyóirat neve	Kódja
64'er Magazin	64er
Chip Magazin	chip
Commodore Horizons	choh
Commodore Microcomputers	comi
Computer!	cutc
Computer Persönlich	pers
Happy Computer	happ
hc - Main Home-Computer	hc
mc - Zeitschrift	mc
Run /USA/	run
Sinclair User	sinc
Your Sinclair	ysin

A tartalomleíró szövegeket permutáljuk, a szövegváltozatokat pedig alfabetikusan rendeztük.

A tartalomleírás egy szövegből áll, majd a listában ezt követi a forrás megjelölése a folyóirat azonosítójával, a megjelenés dátumával és a cikk előkezeléséhez a kezdő oldalszám és a terjedelm megadásával. A mellékelt lista értelmezéséhez még az alábbiakat kell tudni. A tartalomleírás szövegében elsőként a téma átfogó megnevezése, utána a számítógéptípus(ok), ezt követően a szűkebben jelölt tartalom meghatározása szerepel, majd esetlegesen néhány, a közleményt minősítő adat (például : cikksorozat).

A forráshely karakteresorozatát nyílvesszővel, a két / jel között az átváltás, folyóiratszám és kötőjellel a kezdő oldalszám követi, a végén pedig a közlemény teljes oldalterjedelme áll.

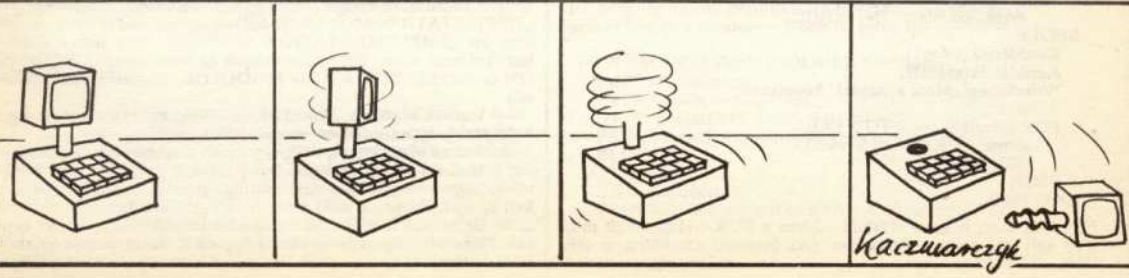
A folyóiratok a SZÁMALK szakkönyvtárban (Budapest XI., Szakasits Á. út 68. Nyitva: 8-tól fél 5-ig. Tel.: 853-111/251) is fellelhetők. A kiválasztott anyagról másolat rendelhető az alábbi formában:

SZÁMALK Szakkönyvtára
Budapest, 112. Pf.: 146. 1502

Megrendelem a Mikroszámítógép Magazin 1987/ sz. alapján a következő folyóirat-
oldal-másolatokat:
Kód: _____ Pédányaszám: _____
Kód: _____ Pédányaszám: _____
Kód: _____ Pédányaszám: _____

A megrendeléshez csatolom az oldalankénti 8,- Ft-os szolgáltatási díj befizetését igazoló csekszelvényt.
Dátum, név, pontos cím.

- | | | |
|---|---|---|
| <p>ZENE
apple II atari 520st commodore 64 12
8 illesztomodul piaci kinalat nido: s
szabvny ->cutc/86.01-24/5</p> <p>ZENE
atari st barkacsolas stereo hangkim
anet ym 21497 ic-ye kiszitési utmu
tato ->happ/86.10-33/1</p> <p>ZENE
atari st programlista zongora szimul
acio ->cutc/86.03-110/3</p> <p>ZENE
atari x1 x2 soundmachine ismertete
s 29 ->happ/86.12-15e/1</p> <p>ZENE
barkacsolas commodore 64 illesztom
di kiszitési utmutato kepcsolesi ra
ja ->mc/86.01-46/2</p> <p>ZENE
barkacsolas commodore 64 masodik sz
erep sid kapcsolasi rajza mukodesvi
szgalo program ->64er/86.11-158/4</p> <p>ZENE
barkacsolas commodore 64 programlist
a hangdigitalizal kiszitési utmutat
o ->64er/86.10-65/7</p> <p>ZENE
cikksorozat commodore 128 veletlen-h
ang generalis sprite definialas
->choh/86.04-35/1</p> <p>ZENE
commodore 128 programlista kiegészit
es a kezikonv sound utmutatojhoz
->run2/86.04-104/2</p> <p>ZENE
commodore 1e programlista zongora sz
imulacio ->run2/86.03-59/4</p> <p>ZENE
commodore 64 hanginta digitalizal
mukodesmod blokkema
->run2/86.09-16/3</p> <p>ZENE
commodore 64 hypra-basic programlist
a sid-programozas poke nélkül
->64er/86.11-65/2</p> | <p>ZENE
commodore 64 interju egy angol szoft
verfejlesztovel ->choh/86.02-32/3</p> <p>ZENE
commodore 64 programlista casaymusic
 poke-helyettesito utastaskeszlet
->choh/86.02-41/1</p> <p>ZENE
commodore 64 programlista simple so
und alprogram készlet
->happ/86.04-70/3</p> <p>ZENE
commodore 64 programlista sound mon
itor alkalmazasi utmutato
->64er/86.10-53/2</p> <p>ZENE
commodore 64 programlista 64-64er-p
alyzat di invertes kompozicioja
->64er/86.1e-173.4</p> <p>ZENE
commodore 64 programlista 64er-64er-p
alyzat di invertes kompozicioja
->64er/86.1e-173.4</p> <p>ZENE
commodore 64 programlista ricker min
t onallo hangdoboz ny bönnie... de
mo ->run2.86.03-95.2</p> <p>ZENE
commodore 64 programlista hanghasas
k keletese ->cutc/86.03-9e/3</p> <p>ZENE
commodore 64 programlista ritmus-sze
kvencial generalisa 17 hangszerre
->64er/86.07-52/5</p> <p>ZENE
commodore 64 programlista zongorabil
lentvuzet szimulacio ahangkimenet b
ovites ->64er/86.02-80/1</p> <p>ZENE
commodore 64 sund expander ismertet
es 399 dm ->run2/86.09-32/2</p> <p>ZENE
commodore 64 termekismertetes digit
al rhythm system hangfile kezelas
idi illesztö ->chip/86.03-22e/2</p> | <p>ZENE
commodore 64 termekismertetes voice
master periferia ->hc/86.04-27/1</p> <p>ZENE
commodore 64 termekismertetes modul
yamaha ym 352e chippel
->64er/86.0e-27/2</p> <p>ZENE
commodore 64 termekismertetes zajsz
ntcsokento eszkozok
->comi/86.01-8e/4</p> <p>ZENE
commodore 64 128 programlista nyolco
ktavas interju v-ezerlesu mi-progr
am ->hc/86.0e-37/12</p> <p>ZENE
commodore 64 128 szintetizator modul
 termekismertetes kfa sound expander
 mukodesmod ->choh/86.02-29/1</p> <p>ZENE
commodore 64 128 termekismertetes s
und expander ->choh/86.04-30/2</p> <p>ZENE
commodore 64 128 termekismertetes pr
ogram udonsagok midi illesztos musi
c maker billentvuzet
->choh/86.01-34/1</p> <p>ZENE
cp m programlista kottafile olvasas
emuf ertekek kiszamitasa
->mc/86.0e-70/5</p> <p>ZENE
hanginta tarolasi technika alkalms
zasok ->run2/86.09-20.4</p> <p>ZENE
midi illesztomodul kinalat az nsh-b
an ->hc/86.0e-23/2</p> <p>ZENE
midi rendszerezles es mukodesmod il
lesztö a c64-hez ->mc/86.0e-20.1</p> <p>ZENE
programlista sinclair spectrum grafi
kus hangmodellenites
->mc/86.07-59/3</p> |
|---|---|---|



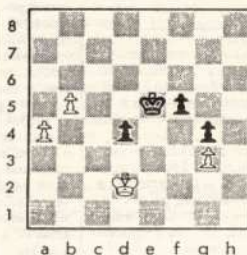
Transzpozíciós táblák

A transzpozíciós táblák a játéka kiértékelésének gyorsítására használják. A program a transzpozíciós táblában jegyzi fel a játéka kutatása során kiszámított adatokat (például minden értékelt csomópontnak a pontszámát). Amikor az ismételt kutatás során ugyanahhoz az álláshoz jutunk, akkor a programnak nem kell ismételtelen értékelnie a pozíciót, hanem csak kiolvassa a táblázatból az állás pontértékét, amelyet az előző kutatás során már kiszámított. Ezzel programunk megtagarítja az értékelő-fügvény hosszú számítási idejét.

A transzpozíciós táblák használatával programunkba könnyen beépíthetünk egy olyan szubrutint, amely lehetővé teszi a lépésméltés felismerését. Ennek segítségével az algoritmus a lépések sorrendjétől függetlenül felhasználhatja a már megörözött pozícióhoz tartozó pontértéket, amikor az adott állás a program elemzése között újabb előfordul. A jelenleg működő leghatékonyabb programok pedig már csak a pozíciót vizsgálják, és a hozzá vezető lépéseket figyelmen kívül hagyják. Így nemcsak a lépésméltést veszik észre, hanem a közbeiktatott lépéseket is. Ezzel az értékelés teljesen függetlenné vált a lépésektől és a lépésszámtól is, csupán a pozíciótól függ. Nem számít, hogy az 4. vagy a 6. lépés elemzése után jön létre. Például az 1. ábrán látható hadállásban ugyanaz a nyerő állás jön létre sötét számára, bárhogy alakul a lépéssorrend a játszma folyamán.

Az első változat: 1. —, f5—f4 2. g3×f4+, Ke5—d6! 3. a4—a5, g4—g3 4. a5—a6, Kd6—c7 5. Kd2—e2, d4—d3+ stb.

A második változat: 1. —, Ke5—d6 2. a4—a5, f5—f4 3. g3×f4, g4—g3 4. a5—a6,



1. ábra

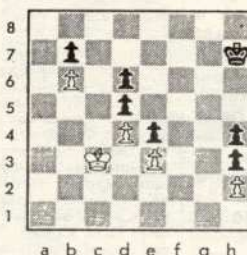
Kd6—c7 5. Kd2—e2, d4—d3+ stb.

Ugyanez a nyerő változat jön létre akkor is, ha közbeiktatunk egy lépéspárt: 1. —, Ke5—d6 2. Kd2—d3, Kd6—e5 3. Kd3—d2, Ke5—d6 4. a4—a5, f5—f4 5. g3×f4, g4—g3 6. a5—a6, Kd6—c7 7. Kd2—e2, d4—d3+ stb.

Láthatjuk, hogy mindhárom esetben ugyanaz a végállás jött létre. Elsőként a legegyszerűbb nyerő változatot láttuk. Másodsorra lépéscserével értük el az adott végállást ugyanolyan fellépés mélységben. A harmadik változatban szintén ugyanaz a végállás, de egy lépéspár közbeiktatásával értük el ezt az előzőekkel azonos végpozíciót. Ezért az utóbbi esetben a játéktábla az első és a második változattal ellentétben mélyebben jutottunk el az adott nyerő álláshoz.

Az intelligens programok a játéka kiértékelése közben csak az első változatot számítják ki, a második és a harmadik változat esetében pedig csak felhasználják a már kiszámított pontértéket.

A transzpozíciós táblák használatával már a megnyitásban is felismerheti programunk a lépéscseréket. Tekintsük például az alapállásból az 1. e2—e4, e7—e5 2. Hb1—c3 lépéssorozat után létrejövő állást. Ugyanehhez jutunk lépéscserével is: 1. Hb1—c3, e7—e5 2. e2—e4. Ha programunk felismeri ezeket a lépéscseréket, akkor a megnyitási könyvtárunkat tömörebbé tehetjük, ugyanis nem kell különválasztani a lépéscserével létrejött



2. ábra

változatokat. Ezzel lehetővé válik, hogy több hasznos információt tároljunk a könyvtárban.

A transzpozíciós táblák használata olyan állásokban a leghatásosabb, ahol csak néhány figura van a táblán, vagy pedig nincsenek mozgékony gyalogok. Ez utóbbi kategóriába tartozik a hadállások mező-ellenes néven ismert csoportjának jelentős része. Egy ilyen állást látunk a 2. ábrán. A világs királynak azalatt kell elérnie egy meghatározott mezőt, miközben az ellenfél királya egy adott helyen tartózkodik. Így sötét lépéskényszerben lesz, királlyal el kell hagynia a számára kedvező mezőt, és így világos két gyaloghátránya ellenére kiharcolja a számára kedvező döntelent: 1. Kc3—b2, Kh7—g8 2. Kb2—a1, Kg8—f8 3. Ka1—a2, Kf8—e7 4. Ka2—b3, Ke7—d7 5. Kb3—b4, Kd7—c6 6. Kb4—a5 (Ezzel a c6 „ajtó” bezárult, most a sötét király a másik irányba veszi az útját.) 6. —, Kc6—d7 7. Ka5—b4, Kd7—e6 8. Kb4—c3, Ke6—f6 9. Kc3—c2, Kf6—g5 10. Kc2—d1, Kg5—g4 11. Kd1—e2 stb. döntelent.

Egy gyors program, amely alkalmazza a táblatranszpozíciót, a játszmafa kutatása során végigszámolja az összes olyan lehetséges állást, amelyben a gyalogok ugyanazon a helyen állnak. Esetünkben megállapítja, hogy egy bizonyos állásban világos döntelent tud tartani, és ilyen gyaklosgszerkezet mellett az összes lehetséges állás tárolásával a program ki tudja számítani,

hogyan érheti el ezt a kritikus pozíciót.

Az áttáronspozíciós táblák nagyon jól alkalmazhatók kevés bábos, gyalog nélküli végjátékokban. (Ezt részletesen tárgyaltuk az 1987. júliusi számban.) Ezekben tökéletessé teszi a számítógépet, és ha az állás nyerhető, akkor a program meg is nyeri.

A vezér—király, bástya—király elleni végjátékban nyilvánvalóan a vezérelt rendelkezés fél nyer, de a végjáték közel sem triviális. Belle összeállította és összerendezte az összes király—vezér, király—bástya elleni végjáték teljes adatmezőjét, amelynek alapján tökéletesen tud játszani. A vezérelt mindig nyer, és gyakran ér el döntelent a bástyával, kivéve mesterek és nagymesterek ellen. Egyszer száz dollárt nyert Walter Browne nemzetközi mester ellen, aki biztos volt abban, hogy vezérelt tud nyerni. A program játszott a bástyával, és döntelent ért el. A szovjet Kaissza is ért el ezen a téren eredményeket, felhasználva a transzpozíciós táblák teljes adatmezőjét vezér és b-vagy g-gyaloggal vezér elleni végjáték tökéletes vezetéséhez, továbbá bástya és gyalog bástya elleni vezetéséhez. Jonathan Schaeffer is felhasználta ezt a technikát Phnix programjában, ahol a táblázat tárolására 6 Mb-ot memóriát használt fel, és táblázatának 2¹⁹ be lépési pontja van.

A transzpozíciós táblák használata nagy memóriakapacitást igényel. Ezért főleg a nagy háttértárral rendelkező számítógépeknek alkalmazzák sikeresen. Ennek ellenére kísérletek folynak olyan technikákra a kifejlesztésére, amely kisebb tárkapacitást igényel, mivel csak a szóba jövő legjobb változatok adatait írja a táblázatba. Ennek az a hátránya, hogy a táblázatban letárolt csomópontok az összesen csak egy töredékét képezik, és ezeket heurisztikus alapon működő szelektálással választják ki. Ez pedig nem mindig vezet eredményre.

KOVÁCS P. ATTILA

Postánk mind mennyiségét, mind pedig tartalmát tekintve változó képet mutat. Állandóan vízszaterő kérdés, hogy küldjünk programokat, alkatrészt, könyvet és ki tudja, még mit. Ezeknek a megrendeléseknek a teljesítését természetesen nem ingyen kéri olvasóink, hanem pénzért, például utánvétes csomagban szeretnék az árut megkapni. El kell hogy mondjam, nemcsak a lapunk mikro, de a szerkesztőségünk is, így az ilyen kívánságok teljesítésére nincs szabad kapacitásunk. Az óhajokat tartalmazó levelek jó részét éppen ezért ebben a rovatban közöljük; nagyon sokszor előfordult már, hogy egyik olvasónk ki tudta elégíteni a másik olvasó kívánságát. Ezt továbbra is reméljük.

Nemesok Ábel, Kiskunfélegyháza

Ebben az évben kaptam egy C64-et (az újabb fajtát), és így elkezdtem böngészni a vele kapcsolatos írásokat. Így bukkantam a Mikroszámítógép Magazinra, ami nagyon megröszélt. Az 1986. 4. évfolyam 11–12. számában egy képet találtam a kínai Nagy Falról (70. oldal). Azt szeretném megtudni, hogy lehet C64-re beírni. Vagyis a programját szeretném.

Második kérdésem az lenne, hol van Kiskunfélegyházán számítógépszerviz? Vagy hol van legközelebb?

Harmadik kérdésem, hogy hol lehet QUICK—SHOT II. botkormányt kapni, mennyiért és jó-e a C64-re (az újabbra: COMMODORE 64 PERSONAL COMPUTER).

És végül, hol lehetne megvásárolni a Hornig—Trapp—Welter: Tippek és trükkök a Commodore 64-eshez című könyvet?

Ha volna, sziveskedjenek részemre utánvéttel elküldeni.

Dr. Broczkó Péter csak a kínai Nagy Fal képet hozta haza kínai útjáról, a programot nem. Van egy ötletem: el kell készíteni!

Nem ismerem Kiskunfélegyházán a szervizüzemeket, viszont azt tudom, hogy van a Szechenyi úton egy Chip bolt és valószínű, hogy a helyi Centrum Áruház is forgalmazza a C64-et, ezért úgy gondolom, hogy talán ők tudnak segíteni.

A kérdézt botkormányt a FotoElektronik, Budapest, Múzeum krt. 19. szám alatt lévő üzletében kaphatja meg 1380 Ft-ért. Ez is jó az ön gépéhez és az a típusú botkormány is, amelyet ebben a számunkban ismertettünk.

A kért könyvet megvásárolhatja Budapestben a Novotrade 2C Számítástechnikai Áruházban (XIII. ker., Zalca u. 35.) és a Magiszter Könyvesboltban (V. ker., Városház u. 1.) 302 Ft-ért, de próbálja meg a helyi könyvesboltot is.

Göblös László, Kalooca

A µM 1987/10. számában olvastam a „Szemüveg alatt a számítástechnika ártaimai” című cikket. Engem is érintene a dolog, mert szabad időmben sokat ülök a gép, illetve a képernyő előtt.

És első kérdésem az lenne, hogy hol készítik el ezt a bevonatot szemüvegre (dioptriás szemüvegre).

A második kérdésem, hogy miért éppen a szemüvegre készítene bevonatot. Nem lehetne a képernyő elé közvetlenül egy UV-szűrős lapot helyezni?

Dioptriás szemüvegre a féldobonatot az Ofo-

tért XII. ker., Csörsz u. 39. sz. alatti üzletében készítik el.

Ötlete igen megfontolandó, még tovább is fejlesztem: esetleg a képernyőt és nem a szemüvegre kellene bekenni a csodaszerral.

Horváth Gábor, Miskolc

1. Úgy látszik, csak nem akarnak leszokni arról, hogy a listákat két alapon közöljük. Miért gondolják azt, hogy jobban olvasható? Vagy megjelénés után már nem szokták átvolni a lapot? Így egy ilyen lap szerintem nem engedhet meg magának ilyen felületiséget!

2. Örülök, hogy átvették azt a szokást, hogy a programokhoz kontrollszöveget képző programot közölnek. Remélem, ez el is fog terjedni, a 64'er Magazinban már évek óta divat, én is sikeresen alkalmaztam és alkalmazom ezt a módszert. Ezzel kapcsolatban csak annyit, hogy ott egy oldalon, egy cikkben és valamivel érthetőbb magyarázták el a program lényegét, hasznosságát, legalábbis én — és ezt nem szívessem bevallani — a 64'erből jobban értem, mint a magyar újságokból. Néha jó lenne gondolni a számítástechnika „egyszerűbb” kedvelőire is.

A két szint szerint használjuk, hogy a programlistákat kiemeljük. Lehet, hogy a szöveget kellene kétíteni és nem a listákat?

Ami a 64'er Magazinral való összevetést illeti — igyekszünk, igyekszünk.

Íj. Fábán Ferenc, Heves

Szeretnék választ kapni arra a kérdésemre, hogy hol kapható bővített C16-ra olyan program, amivel nyilvántarthatnánk pl. kazettát, könyveket és más hasonló dolgokat. Létezik-e ilyen program erre a gépre? Mivel nekem ilyen gépem van. És ha van, mennyibe kerül?

Nem tud-e olyan könyvről vagy programról, aminek segítségével átirhatjuk más Commodore gépek programjait a saját gépre, és ha van, akkor az mibe kerül.

En nem ismerem adatkézelő programot a C16-ra. Ha a kérdést jól értem, akkor a C16-tal nem kompatibilis Commodore gépekről (pl. C64, C128) szeretne automatizálni programot átvenni a C16-ra. Nem tudok ilyen programról. Más se?

Gábor Zsolt, Magyaróvár

Szeretném megkérni, hogy a következő Mikro Magazinokban nyomatassanak ki Enterprise számítógépre való programlistákat. Ha nem írják le a magazinban, akkor küldjék el címemre.

Legyenek szívesek megírni, hogy hol lehet Enterprise-ra való BASIC nyelvet tanulni és mennyiért.

Az előző magazinban megírtam, hogy az Enterprise szoftverrel meglehetősen szegényesen ellátott gép, mi sem kapunk közzétett programokat az Enterprise-tulajdonosoktól. Enterprise-tanfolyam úgyben azt javaslom, hogy a gépet beszerző Novotrade-hoz forduljon. (Budapest, Füst Sándor u. 24—26. 1136.)

Turay Gábor, Budapest

Amikor a lap novemberi száma a kezembe került, és eljuttattam az „Adok-veszek-cserélek” rovatához, pillanatok alatt elborult az agyam. Majd amikor az „Olvasó írja” rovatban olvastam a felhívást, miszerint szólniuk hozzá a témához, nem késlekedtem tovább, előkaptam az írógépet, és rajta!

Nem akarom esztelenül a különbséget a nyugati és a hazai csoportjaink között, mégis úgy érzem, ez a különbség a fő oka annak, hogy kialakulhatott a vita a szoftverlopásról. Nem tudom, miért kell azon kiakadni, ha valaki a barátaitól, ismerőseitől lemásol egy programot. Ebben az országban, ahol a számítástechnikát az élet sok

területén (például az oktatásban) szükséges rossznak tartották és tartják ma is. Érthető hát a különbség az országok piaci között. Mire a kereskedelem észbe kapott volna, máris elárasztották a géptulajdonosokat „kalóz másolatokkal”.

A kereskedelem nem tudott — és valjuki be, nem is nagyon akart — lépést tartani ezekkel a másolatokkal. Persze egyszerűbb semmit sem csinálni, majd utána morogni, hogy „lopják a szoftvert!” — Tehát természetesen tartom, hogy ilyen adottságok hatására kialakult a „cserekereskedelem”. Gondoljunk csak bele, hány programja lenne-e „becsületel” spectrumosnak, aki mondjuk önrészről csak a Bizományból vagy a Novotrade Rt.-től venné a programjait! És azt hiszem, nem nehéz elhinni, hogy erre alaposan rámenne a heti közspézn. Egyszerűsen nem tudom megérteni, miért ellenzik a becsületelcsere? Tudom, meg tudom érteni, hogy a másolat nem teljesen jogszerű. De kérem azokat, akik szidják a programcsért, értsék meg, hogy egyszerűen nincs más mód a programok terjesztésére!! Aki — el kell ismerni: nem kis pénzért — megvesz egy számítógépet, érthető, hogy használni szeretné.

Térjünk át egy kicsit más témára. Nagyfokú aljasságnak és visszaélésként tartom azt a cselekedetet, amit én valódi lopásnak tekintek. Ezen értem a pénzzel történő kereskedést. Példa erre: valaki lemásol egy programot, majd hirdeti egy újságban, hogy eladó. Azt hiszem, mindenki érzi a különbséget e két tevékenység között. Sajnos, sokszor a Mikro M.-ban is jelennek meg ilyen hirdetések!!! Nem értem, hogy engedheti meg a lap magának, hogy pártoljon egy szoftvertolvajt!!! Itt van például egy hirdetés az egyik Mikróból: „C64 programkaszetták eladó. Kazettánként 50-60 program. Egy kazetta 500 Ft”. Gondolom, erre a hirdetésről senki nem gondolja, hogy az illető üzletben vásárolt vagy saját gyártmányú terméket kínál eladásra. Nem értem, nem tudom felfogni, hogy lehet ez egyáltalán eltűnni, hogy lehet valakinek képe ahhoz, hogy ilyet csináljon!?!? Hogy mer valaki más termékével pénzért üzletelni!?!?

A nyitllopás másik, enyhébb formájának érzem a másolóklubbokat. Természetesen itt sem arra gondolok, ha mondjuk 20 fő esténként összejön másolni. Azt értem ezen, ha valaki vásárolt tíz számítógépet, bérel egy üzlethelyiséget, és az arajró mondjuk óránként szét forintért másolhat programot. A tulajnak aztán vannak anyagbeszerzők, akik mindig hozzák a friss programokat, különben hanyatlak az üzlet. Nem tudom, hogy lehet ilyen tevékenységre iparendélyt kiadni!

Végül felteszem a kérdést: Az említett három közül melyik a lopás? Ki szerez jogtalan hasznot? Ki követ el törvénytést? Én úgy érzem, az utóbbi kettő mindenképpen. Lehet, hogy másnak más a véleménye, nekem ez. Elnézést az esetleges felindult hangvételért!

Hogy a végén kezdem, a hangvétel vérmérséklet kérdése, amíg a nyomdafestéket bírja, addig közöljük. Amit ír, azzal egyetérték, azzal nem, hogy a µM gazdasági rendszere egy szerzői jogvédő hivatali járással. A szerkesztőség felvett álláspontra — mi jöhözünk meggyünk. Felvételük, hogy hirdetőink saját tulajdonú (nem bizonyos, hogy saját fejlesztésű) termékeket hirdetnek. Legjobb tudomásom szerint, akinek az érdekét az üzlet sérti, annak kell a megfelelő jogi fórumokhoz fordulni szerzői vagy tulajdonosi jogainak védelme érdekében.

Utolsó: A nyáron összegyűjtött leveleket csak lassan tudom feldolgozni, ezért olvasónk szives elnézést kérem.

Egy — remélem nem bántó — megjegyzés levlélíróinkhoz: ügyeljenek a helyesírásra! Szép számmal kapunk okos leveleket, jó írásokat csapnivaló, de azt is mondhatnám, hogy botrányosan rossz helyesírással. Ha lehet, elküldés előtt olvassák el írásait, és csak akkor adják postára, ha az nemcsak tartalmilag, de nyelvtaniilag is helyes és szabatos.

Ehhez kívánja a legjobbakat:

KOVÁCS GYÖZŐ

Csépai János
— Quittner Pál:
Bit-les számítástechnikai
lemezlovas
(Budapest, 1987.
SZÁMALK,
223 oldal. Ára: 125,— Ft.)

A lemezlovas két zeneszám között érdekes történetekkel szórakoztatja közönségét; hasznosan tölti ki a lemezcserek alatti szüneteket.

A szerzők, vállalva a „hajlékonylemez-lovas” szerepét, szeretnék a személyi számítógépek használóinak széles tömegét megnyerni, hogy üres perceikben, amikor úgyis csak a gép dolgozik, hasznos ismeretek szerzése céljából lapozzák fel a könyvet egy-egy „chip-etnyi” számítástechnika kedvéért.

Mondanivalójukat 69 fejezetben gyűjtötték össze. Az olvasó a számítástechnikával kapcsolatban szinte minden érdekes és érdelemes dologra rátalál. Tájékozódásában a jól szerkesztett tárgymutató is segíti.

A kötet forgatását a szellemes, számítógéppel készített illusztrációk még kellemesebbé teszik. Készítjük, Halász Géza nagyban hozzájárult munkájával a könyv sikeréhez.

Szűcs Ervin:
A számítógép
tegnaptól holnapig
(Budapest, 1987.
Műszaki Könyvkiadó,
136 oldal. Ára: 48,— Ft.)

Mire képes a számítógép és mire nem? Segíti vagy gátolja az emberi élet teljességét? Mindenkinek beszélnie kell-e „komputer”-ről? Színes képekkel, humoros ábrákkal és a hozzájuk tartozó rövid szövegekkel ezekre a kérdésekre ad választ a könyv. Közben meggyőzi az olvasót arról, hogy a számítógép nem „számító”-gép, nem gondolkodó, hanem gondolkodtató gép, amely sohasem teszi feleslegessé az embert.

A könyv nem számítástechnikai szakembereknek íródott. Elsődleges célja, hogy felkelte a fiatalok érdeklődését a mikroelektronika, a számítógép-tudomány, a számítástechnika iránt.

Bodor Tibor:
A Commodore64
programozásának
gyakorlata 2. kötet.
Soros lemezállományok

(Budapest, 1986. (1987.)
SZÁMALK,
188 oldal. Ára: 133,— Ft.)

A kötet a lemezkezelési alapismereteken belül összefoglalja a lemezkezelési szabályokat, ismerteti a lemez fizikai tulajdonságait, több lemezegység kezelését. Attekintést ad a soros állományokkal végezhető alapműveletekről, a rendezett állományokról. Az adatfeldolgozó rendszereken belül a keretrendszert, a rendezetlen karbantartást, a feldolgozási meneteket és a gyorsított karbantartást külön tárgyalja.

dBASE III Plusz szoftver
alkalmazói segédlet
PC adatfeldolgozó
rendszer
(Budapest, 1987.
IPiK,
58 oldal. Ára: 150,— Ft.)

Az Ipari Informatikai Központ által közreadott szoftver alkalmazói segédletek sorozata újabb kötettel gyarapodott. A könyv a dBASE III Plusz három olyan tulajdonságát emeli ki, amely a legelterjedtebb adatfeldolgozó alapnyelvvel tette: könnyen elsajátítható, a feladatokhoz hajlékonyan alkalmazható, s emellett más programnyelvekkel együtt széles körű lehetőséget biztosít nemcsak az adatkezelésre, hanem bonyolultabb feldolgozásokra is.

A kézikönyvben megtalálható a dBASE III Plusz teljes szintaxisa. A dBASE III Plusz hálózatban is üzemeltethető, a hálózat kialakítását azonban a könyv már adottnak tételezi fel.

Interaktív üzemmódban a dBASE III Pluszsal a munka élvezetesebbé válik, mivel a feldolgozás a legfrissebb információkkal történhet, és a személyi számítógépek jellemzőinél fogva azonnali döntéseket tesznek lehetővé.

Bognár Júlia:
dBASE III Plusz
(Budapest, 1987.
SZÁMALK,
255 oldal. Ára: 156,— Ft.)

A könyv az ASHTON-TATE szoftverház dBASE III Plusz adatbázis-kezelő programját ismerteti. Használatát nem igényel mélyebb számítástechnikai vagy adatbázis-kezelői előismereteket. A példákban szereplő parancsok eredményét a szerző a dBASE III Plusz segítségével nyomtatta ki, pontosan olyan formában, ahogy az a képernyőn megjelenik. A könyvben szereplő állományok felépítése és tartalma függelékben megtalálható, és ugyanitt van egy referencia jellegű rész is, az összes parancs pontos szintaktikai leírásával. Külön táblázat foglalja össze a különleges hálózati parancsokat és függvényeket.



COMPAIR 88

1988. október 17–21. NEMZETKÖZI SZÁMÍTÁSTECHNIKAI SZAKKIÁLLÍTÁS

A magyar és nemzetközi számítástechnikai kultúra eredményeinek bemutatója. Fórum a kooperációs, vegyes vállalati együttműködésre. Kelet–nyugati számítástechnikai találkozók.

Tematika:

- számítástechnikai rendszerek, hálózatok
 - hatékonyságnövelő egyedi berendezések és perifériák
 - személyi számítógépek (hobby, irodai, professzionális)
 - OEM-perifériák és mikroperifériák
 - rendszer- és alkalmazói programok
 - hardver-szoftver eszközök
 - elektronikai háttérpár
 - távtanulási rendszerek és eszközök
 - lakossági hálózatok, például videotext
- 5000 négyzetméter a Budapesti Kongresszusi Központban

Felvilágosítás:
 COMPAIR IRODA
 1022 Budapest,
 Bég u. 3–5.
 Telefon: 150-856
 Telex: 22-6708

Várjuk jelentkezésüket!

Bolgár AT

ESZ 1838 néven az IBM PC AT-val kompatibilis mikroszámítógépet mutattak be Bulgáriában. Mint ismeretes, ilyen teljesítményű mikroszámítógépek a szocialista országok közül hazánkban kívül még csak Lengyelországban és Bulgáriában jelent meg, s az XT kategória bemutatása is csak az idén várható Romániában és a Szovjetunióban.

gok közül hazánkban kívül még csak Lengyelországban és Bulgáriában jelent meg, s az XT kategória bemutatása is csak az idén várható Romániában és a Szovjetunióban.



A bolgár ESZ 1838

Tárgynereménysorsolás

A Sportfogatási és Lottóigazgatóságon számítógéppel sorsolják a tárgynereményeket. A feldolgozás előkészítéseként a sorsoláson részt vevő játékhéten forgalomban levő szelvényekből a számítógép kezelési jegyzéket készít. Ennek alapján a szelvényfeldolgozó helyeken megállapítják, hogy a kibocsátott szelvények közül melyek vettek részt ténylegesen a játékban. Ezután már az „elő” szelvé-

gép kezelési jegyzéket készít. Ennek alapján a szelvényfeldolgozó helyeken megállapítják, hogy a kibocsátott szelvények közül melyek vettek részt ténylegesen a játékban. Ezután már az „elő” szelvé-

nyekre sorsolja a Siemens 7740 típusú számítógép az előre megállapított darabszámú és értékű tárgynereményeket: mintegy 7 millió szelvényre kb. 1300 darab nyereményt. A sorsolás közjegyző és pénzügyminisztériumi ellenőr jelenlétében zajlik, de nyilvános, úgyhogy külön kéreésre bárki megtekintheti.

Telefonkönyv

Két éve döntött úgy a Magyar Posta, hogy a korábbi, postaigazgatóságok szerinti bontás helyett megyénkénti telefonkönyveket ad ki. A döntést elsősorban az indokolta, hogy az utóbbi időben megsaporodott a telefonállomások száma, és minden megyében összejött egy kötetre való előfizetői névsor.

Az első hat megyei telefonkönyv elkészülte után a posta megállapodást kötött az Idegenforgalmi Propaganda és Kiadó Vállalattal. Ennek értelmében a vállalat számítógéppel szedik és tördelik a telefonkönyvet, majd lézernyomtatón kinyomtatják, lefotózzák, és a filmről sokszorosítják az oldalakat.

A számítógépes feldolgozás előnye, hogy bizonyos ellenőrzési lehetőségeket nyújt, és jóval gyorsabb a hagyományos eljárásnál: feleannyi időre van szükség egy-egy telefonkönyv előállításához, mint korábban. A gyors átfutási időnek köszönhetően az új kötetekben a legfrissebb adatváltozások is szerepelnek. A múlt év végén megjelent budapesti egyéni előfizetői névjegyzék is bizonyítja, hogy az új technikával készült telefonkönyv áttekinthetőbb és jobb a réginél.

Adóhivatal

1987. július 1-jén alakult meg az Adó- és Pénzügyi Ellenőrzési Hivatal. A mintegy 3000 fős induló gárdájának létszáma 1988 első felében a növekvő feladatokhoz igazodóan 6000-re emelkedik. Hamarosan nagy teljesítményű számítógépekre is szükség lesz; az egyszeri létesítési költség kb. 2,6 milliárd forint. Ebből többek között két darab Siemens 7560/E típusú nagyszámítógépet szereznek be.

Az operatív nyilvántartási feladatok megoldására — például a mezőgazdasági kistermelők vagy a magánlakásépítők forgalmiadó-visszatérítésének lebonyolítására — országszerte kétszáz VT 110 és VT 160-as gépet alkalmaznak.

●●● Pontvadászat ●●●

Új rejtvényt indítottunk útjára lapunkban, remélve, hogy elnyeri olvasóink tetszését. Minden számunkban két feladatot közlünk: az első logikai, matematikai tudást, a második számítástechnikai alapismereteket is igényel.

A feladatok után értékeljük a rész megoldásokat és közöljük az elérhető maximális pontszámot.

A pontgyűjtést, vagyis a pontvadászatot az év végén zárjuk. A legjobb tíz versenyző nevét közzé is tesszük; ők lesznek azok, akik könyvtalványt kapnak.

A helyes megoldások a feladatok közreadása után, két lapszámmal később jelennek meg, így a pontvadászoknak jut idejük a gondolkodásra.

Beküldési határidő: 1988. április 14.

Címünk:

Mikroszámítógép Magazin Szerkesztősége,

1371 Budapest, Pf. 433

Jó vadászatot kíván a feladatok összeállítója:

DR. HOFFMANN TIBOR

1. feladat

Bizonyítsuk be, hogy az első n egész szám köbének összege mindig egy négyzetszám! (5 pont)

2. feladat

Az előző számban közölt 2. feladathoz csatlakozva, az ott kapott mátrixmértellel a nyomtatott angol ABC mely betűit lehet egyszerűen és a számjegyeiktől megkülönböztethetően ábrázolni? (4 pont)

Ha az angol nyomtatott ABC összes betűjét le akarjuk írni a mátrixnyomatóval, mekkora n szélességet és m magasságot kell előírunk? (2 pont)

„Az eddigi legnagyobb COMDEX”

hangoztatta Peter B. Young a COMDEX és még sok más kiállítást szervező cég — THE INTERFACE GROUP — reklámigazgatója. És valóban. 1987 tavaszán az atlantai bemutatóra csaknem hatvanezren, az őszire több mint százezren (!) érkeztek. Ezeröttszáz kiállító jelentkezett termékeivel.

De lássuk, mi is a COMDEX? Ez a legnagyobb amerikai bemutató vásár, amelyen a viszonteladóknak — a kis- és nagykereskedőknek, továbbfeldolgozóknak — évente kétszer bemutatják azt, amit a közeljövőben eladni kívánnak. A termékismertető eredetileg csak belföldi cégek termékeinek bemutatására szerveződött. Azonban mára már húsz ország is képviselt magát a nagy „attrakción”. Mindenki aki számít, ott van. Így jelennek meg az IBM-ék, a belföldi nagyok, a japánok, a tajvaniak éppúgy, mint a számítástechnikában legnagyobb nyugat-európai cégek. Csak a szocialista tábor termékeiből nincs óriási kínálat. Egyedül a kínaiak állították ki, méghozzá IBM kompatibilis gépeket.

Nemcsak hogy magyar vállalat nem képviselte hazánkat, de Young úr közlése szerint a hazai szaksajtó területéről a Mikroszámítógép Magazin — személyemben — volt az egyetlen. Ez pedig már baj, hiszen hogyan lehet úgy tájékozódni a szakközönéséget, ha magának az újságíróknak sincs alkalma személyesen megismerkedni a világ kínálatával... Ugyanis, mire a nyugati szaksajtóból át vesszük a híreket, addigra azok közül nem egy már aktualitását veszítette, illetve bizonyos okok miatt csak részgazságokat tartalmaz. A fentiek miatt is növekszik hátrányunk. Mert például a tajvaniak többek között annak köszönhetik sikereiket, hogy tömegestől küldik szakember-újságíróikat a legkülönbözőbb szakmai kiállításokra, ahol is azok megfelelő — és nem ritkán kellő — információt gyűjtenek össze, s azokat alkalmasint hazájukban hasznosítják is. Talán e módszerrel sikerült elérniük azt, hogy a tőlünk két évtizede még jócskán elmaradt ország mára nemhogy felzárkózott volna, de rettenetesen elhagyott bennünket. Ma mindenholva elektronikus berendezéseket, részegységeket exportálnak. (Azonban ez a cikk nem a tajvaniak sikeréről kíván szólni, hanem egy rövidebb-hosszabb sorozat első ré-

sze akar lenni. Ugyanis a legfontosabb fejlődési tendenciákról szeretnénk betekintést adni. Elsőként a múlt évi hatalmas COMDEX bemutatóról kínáljuk az alábbiakat.)

Las Vegas 7 legnagyobb, legrangosabb szállodája adott helyet a kiállításnak. (Igaz, nem tagadták meg eredeti profiljukat sem, így például 4 perces játékok, s ital állt a vendégek szolgálatára, noha ezúttal ingyen szolgáltatták.) A képen látható Caesar's Palace is kártya- és rulettjátékokra csábított, stílszerűen számítógépes és hagyományos formában. Saját televízió-műsort sugároztak a kiállítás teljes ideje alatt, sőt egy 192 oldalas színes napilapot is megjelentettek, kellő hiranyaggal. Mindeközben természetesen a helyi televízió — de az országos is — rendszeresen foglalkozott a bemutatóval. A sajtót egyszerűen követhetetlen mennyiségű anyaggal igyekeztek ellátni. (Az újságírók munkáját könnyítendő a Novell Inc. 16 terminált bocsátott rendelkezésre.)

Mit láthatott a kíváncsi érdeklődő? A felsorolás véget nem érő lenne. A bemutatóról így csupán izelítőt adhatunk.

Az „amit lát az kapja” (WYSIWYG) rendszerek a képernyőn láthatókat eredeti alakban és színben viszik a nyomtatókra. Például a hátsó borítékon bemutatott menüt a KPG Incorporated Cosmopolitan Centre Panda nyomtatóhajtója segítségével, s valamennyi fontosabb nyomtatóra használható.

Vagy itt vannak az új IBM PS/2 rendszerek adta lehetőségek és az elterjedt professzionális grafikus szoftverek — például AutoCAD, P-CAD, VersaCAD — összekapcsolása. Ezek a szoftverek az új rendszerben meglévő lehetőségek kihasz-

nálására szolgálnak. Továbbá arra, hogy a mérégdírga speciális monitorokat kiküszöböljék. Ilyenek az Electrograph Sales Inc. által forgalmazott Galaxy Mercur 2 kártyák és szoftverek. Ezekkel az említett tervezői szoftverrendszerek kiegészíthetők egy beépített kép — és képrészlet — kinagyítási lehetőségével, 1024 x 768 és 800 x 600 pontos felbontással, a lehetséges 256 000-ból kiválasztott 256 színnel, megtartva az AT-kompatibilitást. (A borítékon lévő kártyákon látható nagy, négyzet alakú alkatrészek is e fejlődés egyik fő irányát képviselik. Nevezetesen, a korábban egy kártyarészleten, teljes kártyán megvalósított áramkörök helyett, ma már egyetlen, felhasználóorientált alkatrészt készítenek. Mégpedig kellő darabszámmal megrendelésnél a tervezéssel együtt 6 hét alatt!)

A nemprofesszionális felhasználóknak is sokat nyújt a grafika fejlődése. Valamennyi géphez készülnek nagy felbontású, sokszínű, sokmögös játékok. Így például az USA-ban igen elterjedt sport a baseball. Ennek magas szintű szimulációjára az NBC — az egyik legnépszerűbb tv-társaság — neves sportriporterének részvételével készült IBM PC/XT/AT-re íródott játék. E programot a Főliga Játékosainak Szövetsége dolgoztatta ki és ismerte el, mint olyat, amely realis játékot tesz lehetővé.

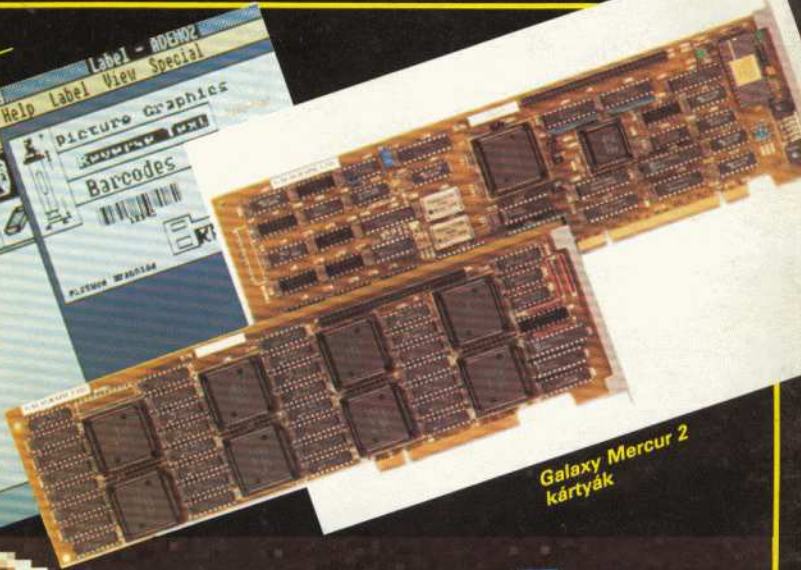
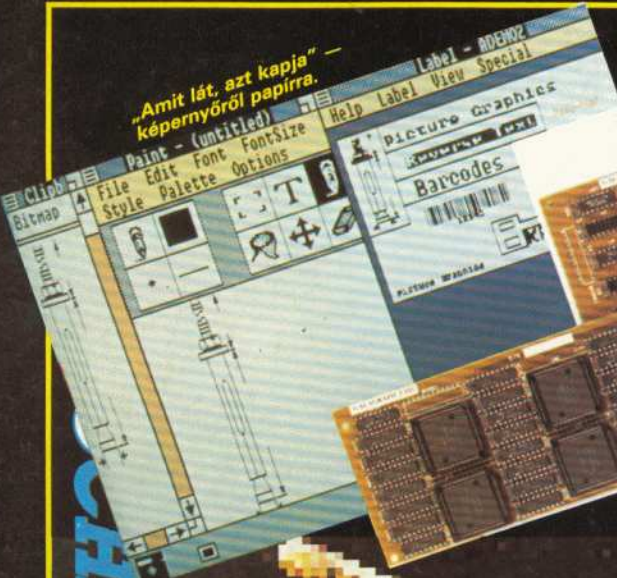
A fentiek természetesen csak izelítőt, pillanatfelvételt kívántak nyújtani a kiállítás újdonságaiból. A többitől majd később...

Végezetül, ha valakit érdekelnek a részletek, szívesen válaszolunk a kérdésekre lapunk hétfő délután 5–7-ig tartó „foga-dóráin”.

SIMONYI ENDRÉ



"Amit lát, azt kapja" —
képernyőről papírra.



Galaxy Mercur 2
kártyák

CHALLENGE BASEBALL



Officially Licensed by
Major League Baseball
Players Association.

Szimulációs
baseball játék