

mikro számítógép magazin





Egyedi és általános szoftverek a vállalati ügyvitel területén. Termelésirányítási, raktárgazdálkodási programcsomagok egyedi és hálózatos gépeken.

A szoftvereket kívánság szerint, megrendelőink igényeinek és a helyi viszonyoknak megfelelően átalakítjuk, adaptáljuk. Helyzetfeltárással, rendszertervezéssel, be tanítással, a megfelelő számítógépes konfiguráció kiválasztásával és beszerzésével segítünk vállalati problémáink megoldásában.

A garanciális idő letelte után átalánydíjas szervizszolgáltatást vállalunk számítógépeikre.

Egyedi fejlesztések, tervezések és témaorientált feladatok megoldásával is forduljanak hozzánk.

MS MIKROSZERVIZ Számítástechnikai
Műszaki Fejlesztő
1141 Budapest, Kőszeg u. 4. Kiszövetkezet

Telefon: 831-805

Bankszámla: MHB 207-57 821



A NEUMANN JÁNOS SZÁMÍTÓGÉPTUDOMÁNYI TÁRSASÁG ÉS A KISZ KÖZPONTI BIZOTTSÁG LAPJA

**A kiadvány
a Tudományos-
és Informatikai
Intézet
együttműködve készül**

**A szerkesztőbizottság
vezetője:
Kovács Győző**

**E számunkat
szerkesztették:**

**Bakos Tamás
(programozástechnika)**

**Broczkó Péter
(hírek)**

**Kovács Győző
(levelezés)**

**Lindner László
(sakkprogramozás)**

**Petróczy Judit
(könyvek)**

**Simonyi Endre
(klub)**

**Varga András
(iskola-számítógép)**

**Címkézpünk:
Ramocsal Imri munkája**

**Felelős szerkesztő:
Könyves Tóth Pál**

**Szerkesztőség:
1027 Budapest, Fő u. 68.
Telefon: 154-250**

**Levél cím:
1371 Budapest
Pf. 433.**

**Kiadja az Ifjúsági Lap-
és Könyvkiadó Vállalat**

**Felelős kiadó:
dr. Petrus György
igazgató**

**Kiadóhivatal:
1065 Budapest, Révay u. 16.
Telefon: 116-660**

**Terjeszti a Magyar Posta
Előfizethető a hírlapkezelés
hivataloknál
és a Posta Hírlapelőfizetési
és Lapellátási Irodáján
(1900 Budapest V.,
József nádor tér 1.)
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.
86-0253**

Tartalom

Műszaki emlékeink	2
OKTA-TOTÓ	11
Miért nem stupid a MUPID?	24
Nyomatok holnapja	26
Rendszerfejlesztési eszközök. Szoftver	28
Ígéretes kezdet	29
Még egyszer az Enterprise-ről	30
μINFORM	37
Az Olivetti nem hibázik	39
Adok — veszek — cserélek	39
Olvastunk . . .	40
Petike kártyázik	43
A természet vigyáz ránk	44

ISKOLA-SZÁMÍTÓGÉP

MINI PLUS-Comp BASIC fordító	3
A ROM-ban tárolt aritmetikai rutinok használata	3
TechnoMIR	5

DIÁKROVAT

Betöltő, darazsak, energiaelosztás	6
------------------------------------	---

PROGRAMOZÁSTECHNIKA

BASIC és gépi kód	12
Z80 programok haladóknak Spectrumra és Primóra	13
Verem a Spectrum BASIC-hez	14
A Spectrum másolói, másolók spektruma	15
Számítógépes grafika Pascalban. Egyenes	17
Hangos billentyűzet	20
Lebegőpontos tömbök átadása. BASIC-Pascal	21

μPROGRAMOK

Memóriaolvasó program	31
Felhasználói karakterek	31

μKLUB

2—4 kb-ítos EPROM-égető a Spectrumhoz	32
Integrált szoftver	36

SAKKPROGRAMOZÁS

A játéka és kiértékelése	42
--------------------------	----

AZ OLVASÓ ÍRJA

	45
--	----

KÖNYVEK

	47
--	----

HÍREK, ÉRDEKESSEGEK

	48
--	----

**μ mikro számítógép
magazin**



**Szikra Lapnyomda
Budapest (87-1520)
Felelős vezető:
Csöndes Zoltán vezérigazgató**

**INDEX: 25 629
ISSN 0236-6088**

Műszaki emlékeink

„Világos, hogy az olyan problémák, amelyeknek tipikus példája a numerikus időjárás-előrejelzés, teljességgel megoldhatatlanok volnának elektronikus számítógépek nélkül. Eppen ez a modern gépek lényege. Nemcsak egyszerűen meggyorsítják az ember vagy az elektromechanikus gép által végzett unalmas, fáradszó és hosszadalmas számításokat. Olyasmint tesznek lehetővé, amit korábban egyáltalán nem lehetett volna megcsinálni. Az elektronikus elv sokkal többet tett, mint csak megszabadította az embert attól, hogy rabszolga módjára órákat vesztessen el a számítások végzésére; általa olyasmis is lehetővé és végrehajthatóvá vált, amit az ember egyedül sohasem lett volna képes véghezvinni. Ez tette lehetővé, hogy emberek lépjenek a Holdra — és ennek segítségével sikerült másokat épségben visszahozni a Holdra irányuló sikertelen útról.”

(H. H. Goldstine: A számítógép Pascaltól Neumannig)

Úgy emlékszem, hogy egyik írásom élen egyszer már idéztem H. H. Goldstine gondolatait. Most azért teszem újra, mert — ha erre egyáltalán szükség van — bizonyítani szeretném, hogy a számítógép nemcsak egy a világ számos műszaki alkotása közül, hanem éppen az egyik legfontosabb: a számítógép révén indult el a XX. század második felének az a szédületes műszaki, intellektuális, gazdasági, sőt társadalmi fejlődése, amelyet méltán neveznek a mai kor ipari forradalmának.

Akik ismernek, tudják, hogy minden iránt, ami számítógép, számítástechnika, informatika, mérhetetlenül elfoglalt vagyok. Tehát ebben a rövid írásban se várjanak tőlem mást, mint szubjektivitást.

Számítástechnikai műszaki emlékeinket vagy 15 éve próbáljuk megemlíteni a következő generáció számára. Akkor határoztuk el ugyanis: nem hagyjuk, hogy a kiöregedett számítógépek a MÉH-telepre jussanak, és így örökre eltűnjenek a semmiben, hanem elraktározzuk valahol, és ha lesz egyszer egy olyan országos műszaki múzeum, amely a műszaki emlékeket ki is tudja állítani, akkor ezeket a gépeket bemutatjuk a nagyközönségnek.

A gyűjtőmunka megkezdődött. Erről dr. Muszka Dániel részletesen beszámolt a *µM* 1986. márciusi számában. Ő vállalta a Nagyalföldi Köolaj- és Földgáztermelő Vállalat algóji telephelyén raktárában tárolt számítógépek szakszerű gondozását és nyilvántartását is. Cikkéből kiderül, hogy egyre nehezebb a gyűjtők munkája, hiszen a korlátozott területű raktárban a hatvanas évek ma már kiöregedettnek számító ICT, Elliott, Siemens, Minszk, Razdan és egyéb gépei mellett egyre kevesebb a hely a folyamatosan selejtezett újabb berendezések számára.

Azt szerettük volna, hogy a szakma kívül-alába tartozó vállalatok, kutatóintézetek, egyetemek, főiskolák vállalják el egy-egy öreg számítógép egyes darabjainak megőrzését: készíttessenek egy-két üveges szekrényt, és azokban állítsák ki a gépeket márvánnyal burkolt, elegáns belépőikben, előcsarnokaikban. Ezáltal nemcsak a tárolóhelyet tudnánk növelni, hanem sok ember számára lehetővé tennék, hogy megismerje a közelmúlt egykor nagy teljesítményűnek tartott számítógépeit is. Erre a kérésre azonban akkor egyetlen hazai intézmény sem reagált.

Most, ahogy mondani szokták, van egy jó és egy rossz hírem.

A jó hír, hogy néhány lelkes szakember támogatásával talán még az idén létrejön két állandó számítástechnika-történeti kiállítás: az egyik a budapesti Landler Jenő Gimnáziumban, a másik a hőmezővásárhelyi Kossuth Szakközépiskolában. Ezekben a „számítógépes középkadereket” képző intézményekben felismerték annak fontosságát, hogy az odajáró tizenéves gyerekek ne csak BASIC programot tudjanak írni személyi számítógépre, hanem ott az iskolában tanulmányozhassák a számítógépek és számolóeszközök fejlődését is.

Az iskolák vezetőinek egyetértésével úgy tervezzük, hogy a kiállítások nyilvánosak lesznek, tehát amikor nincs tanítás, például a hétvégeken, bárki látogathassa azokat. A diákok vállalják a gépek felügyeletét, és szakszerű magyarázattal szolgálnak az érdeklődőknek. Ha nem tévedek, ez a két iskola elsőként vállalkozik ilyen fontos kulturális szolgálatra, egy állandó technikatörténeti bemutató üzemeltetésére.

A rossz hírem az, hogy még ezzel sem enyhülnek lényegesen a raktározás gondjai. Az Országos Műszaki Múzeum vezetésével újabb számítógépek múzeumi átvételéről folytatott tárgyalásunk sajnos eredménytelen volt. A szűkös elhelyezési lehetőségekre hivatkozva az álláspontunk, hogy nagyon megvágatják, milyen darabokat őrizzenek meg a kiöregedett számítógépekből; arról szó sem lehet, hogy teljes konfigurációk raktározását megoldják.

Ezzel a véleményem azonban nem lehet egyetérteni. Már csak azért sem, mert nézetem szerint föltétlenül elhibázott szelekciós szempont a helyhiány, hiszen egy régi darab technikatörténeti értékét nem az általa a raktárban elfoglalt terület mérete határozza meg, hanem valami egészen más. Ha arra gondolok, hogy nincs a gyűjtemény-

ben EMG 803-as, hogy a KFKI gépeiből csak egyetlen TPA—8-ast, a Videontól csak az R10-est őrzik, és nincs a régi lyuk-kártyás rendszereinkből sem... Pedig el kellene raktárnak a SZTAKI, az SZKI, a SZÁMALK, a KFKI esetleg be sem fejezt kísérleti eszközeit is. Szóval kellene a hely és kellene a gépek is.

Így megint a Neumann Társaságra maradt az újabb gépek gyűjtése és tárolása, de ezt a feladatot csak a műszaki értékekért felelősséget érző és vállaló intézményekkel együtt tudja megoldani.

Ismételt felhívásunkat kultúrházaknak, művelődési központoknak, oktatási intézményeknek, főleg egyetemeknek és főiskoláknak, nem kevésbé a hazai számítástechnikai gyártás és kutatás felelőseinek, kutató- és fejlesztőintézeteknek címezzük. Azt kérjük, vállalják el a gyűjtemény egyes darabjainak kiállítását és megőrzését.

Kérjük továbbá mindazokat, akiknek régi gép van a birtokukban, és azt ki akarják selejtezni, értesítsék a NJSZT titkárságát, hogy ezeknek a becses műszaki emlékeknek a megőrzéséről együttesen tudjunk gondoskodni. És mivel a gépek szállítása is problémát jelent, azoknak az intézményeknek a jelentkezését is várjuk, amelyek ezt vállalnák.

De nemcsak a gépeket szeretnénk megőrizni, hanem egyéb informatikai emlékeket is: programleírásokat, érdekesebb programfuttatások eredményeit, grafikákat, fényképeket, adathordozókat is. Mindezekből igen kevés maradt meg, de talán valaki megtartott néhányat emléklül. Kérjük ezeket is, annak reményében, hogy egyszer lesz majd egy informatikai múzeumunk is, ha nem is olyan szép és nagy, mint a londoni, a müncheni, a bostoni vagy éppen a bécsi, de mindenképpen a magyar műszaki múlt, a hazai informatikortörténet egy darabjának kiállítóhelye.

Be kell vallanom, egy kicsit irigykedem, hogy ezek a gyönyörű műszaki gyűjtemények éppen az iparilag legfejlettebb országokban vannak. Valaki azt mondta: azért, mert ezeknek az országoknak sok pénzüket van, így arra is jut, hogy egy halom „öcskavasnak” külön múzeumot építsenek. Bennem viszont az a kérdés ötlött fel: vajon nem azért olyan gazdagok és fejlettek-e ezek az országok, mert meg arra is képesek voltak, hogy műszaki alkotásaikat — azt a bizonyos halomnyi „öcskavasat” — a jövő generációjának, tanulásul, megőrizzék.

KOVÁCS GYÖZÖ

MINI PLUS-Comp BASIC fordító

Talán mire megjelennek ezek a sorok, már nem is számít újdonságnak a hír: elkészült a PLUS-Comp, a Plus/4-es BASIC fordítóprogram kazettás verziója. (A lemezes változatot lapunk ez évi 4. számában ismertettük.)

Tudomásom szerint ez az első olyan Commodore BASIC fordítóprogram, amely nem mágneslemezzel, hanem kazettával működik. Bizom benne, hogy nem az utolsó, de erről majd később...

Miben különbözik a MINI a PLUS-Comp-tól? Elsősorban a lefordítható BASIC program nagyságában:

	Lemezes	Kazettás
A BASIC sorok max. száma	1600	310
A vezérlés-átadó utasítások száma	2000	400
A program-méret	kb. 25 kbájt kb. 12 kbájt	15 kbájt (nem graf.) 7 kbájt (grafika)

A másik különbség a munkaterület helye. A PLUS-Comp ugyanis mágneslemezen tárolja a munkafájlokat, a MINI esetében pedig a memóriában készülő a futtatható P-kódú program. Kazettára csak a hibafájli íródik, ha ezt a fordítás előtt kérte.

A fordítóprogram, hasonlóan a „nagyobb testvérhez”, elsősorban a CBM 2.0-ás BASIC verziót támogatja, de direktívák segítségével a 3.5-ös verzió is beépíthető a P-kódú programba.

Mivel a fordító felülről lefelé (top-down), balról jobbra (left-right), visszalépés nélküli algoritmussal dolgozik, ha az elemzett BASIC kifejezés legelső tagja nem BASIC 3.5-ös utasítás, direktívával meg kell jelölni, hogy a fordító helyesen építse be a P-kódú programba. Az így megjelölt sor nem tartalmazhat vezérlésátadást és FOR...NEXT ciklust (FOR, NEXT, STEP).

A direktívában álló kifejezés vagy sor szintaktikus elemzése csak futás közben folyik. Így fordításkor ezekben a részekben még maradhat hiba. Általában igaz, hogy csak alaposan tesztelt programokat érdemes lefordítani.

A lefordítandó BASIC programban a fordító nem használhatja az alábbi utasításokat:

CONT, LIST – a P-kódból származó korlátozás

ELSE

DO UNTIL/WHILE EXIT LOOP

RESTORE n

TRAP/TRON/TROFF/RESUME

A fejlett programozási technikáról való lemondás az ára a mintegy 4–12-szeres sebességnövekedésnek. A legnagyobb sebességet egész típusú FOR ciklusba ágyazott GOSUB-oknál mértem: ez kb. 25-szörös volt a BASIC-hez képest.

A fordító ezek alapján Commodore 64-es programok Plus/4-re átirít változatainak fordítására kiválóan alkalmas. A direktívakezelés elsajátítása kezdő programozóknak több időbe telhet, mint azoknak, akik már dolgoztak fordítóprogramokkal. Ezt a kellemetlenséget egy készülő előfordító fogja megszüntetni, amely a lefordítandó BASIC programot automatikusan fordítható állapotba alakítja.

Tervezés alatt áll egy 15 kbájtnál nagyobb programok fordítására alkalmas Plus/4-es fordító, amely a munkafájlokat kazettára írva hozná létre a P-kódú programot. Természetesen ez több időbe telik, mint a memóriában való fordítás. Ha igény lesz rá, TURBO MPC néven kerül majd forgalomba.

Egy bonyolult, sok feltételt tartalmazó program futási ideje BASIC-ben 5 óra volt, ami a lefordított program esetében majd 30 percre csökkent. Ekkora különbséget már mindenképpen meg lehet érezni!

Mivel a MINI fordító készítésénél sok olyan problémát meg kellett oldani, amely fellépne Commodore 64 és C-1530 esetében is, ha megfelelő igény lesz rá, a C64-es verzió elkészítése is szóba jöhet. A leendő fordító érdekessége, hogy a lefordított program a BASIC munkaterületből kevesebb helyet foglal, mint BASIC-ben, függetlenül a BASIC program nagyságától – ellentétben a közismert C64-es BASIC fordítókkal, ahol a szükséges RUN-TIME modul egysoros program esetén is a program elé generálódik, minimálisan 17 blokknyi helyet foglalva a BASIC munkaterületből.

BÁRÓ CSABA

A ROM-ban tárolt aritmetikai rutinok használata

TVC

A TV-Computeren az assemblér hiánya miatt csak rövid gépi kódú programok írására vállalkozhatunk. A sokat használt ciklusmagok gépi kódban való megírásával azonban csökkenthető a futási idő, és olyan programok is futtathatók lesznek, melyek tisztán BASIC-ben megírva igen lassúak. A ROM adatmozgató és aritmetikai rutinjaival gépi kódú programunkat szubrutinhívások sorozatává rövidíthetjük. Ezen a módon tesztölegetesen bővíthető a beépített függvénykészlet is.

Amint a gép leírásából kiderül, a TV-Computer az aritmetikai műveleteket a BASIC veremben végzi, és ez a helye a beépít-

tett függvények közötti paraméterátadásnak is. A verem utolsó elemére az IY regiszter mutat. A programírás során – különösen ciklusoknál – mindig figyelni kell ennek helyes állítására.

A lebegőpontos számokat a gép háromféle számábrázolási formátum szerint kezeli, amit az ábrákon szemléltetünk. Az ábrázolás bájtönként történik. Egy mantisszabájt két BCD számjegyet tartalmaz.

Az 1. ábrán a verem látható. A számok magukban foglalják a túlszordulásbájtot, amellyel együtt a verembe elhelyezett számok 14 decimális számjegy pontosságúak (egy bájt = két decimális számjegy).

Az ún. regiszter formátumú számoknál nincs túlszordulásbájt (2. ábra). A gép két beépített aritmetikai regiszterét (X és Y) a beépített függvények használják; a processzor HL és DE regiszterpárjainak megfelelő állításával több ilyen regiszter kialakítható a gép memóriájában.

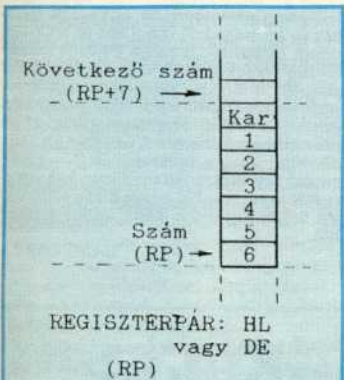
A 3. ábra az X és Y belső aritmetikai regiszterek felépítését, a 4. ábra a szimbólum-táblázatban (változótáblázatban) levő számok formátumát mutatja be. Ez utóbbiak már csak 10 decimális számjegy pontosságúak, ami egyben a megjelenítés pontossága is. A különböző formátumú számok közötti konverziót a ROM adatmozgató rutinjai végzik, a szubrutinhívások során erre nem kell ügyelni.

A szubrutinok vagy a szokásos módon, CALL utasítással, vagy táblázatból való kijelöléssel hívhatók. Ilyenkor a szubrutin táblázatbeli sorszámát kell megadni az alábbiak szerint:

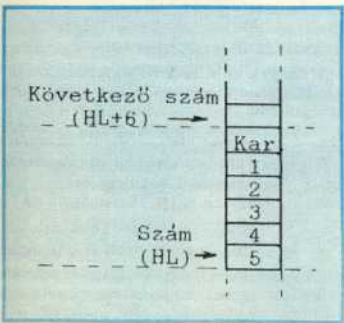
- RST 18H A táblázat aktivizálása
sorszám 1 Az 1-es sorszámú szubrutin végrehajtása
sorszám 2 A 2-es sorszámú szubrutin végrehajtása



1. ábra. Az aritmetikai verem



2. ábra. Egy tetszőlegesen kialakítható aritmetikai regiszter



3. ábra. Szimbólumtáblázat-elem

rutinlista végét az jelzi, hogy a sorszám legnagyobb helyértékű (128-as súlyú) bite 1-et tartalmaz. A processzor az ezután következő kódot már utasításként értelmezi, és végrehajtja.

A hívható adatmozgató szubrutinokat a táblázat tartalmazza.

A szubrutin hívása	Hová	Honnan	Megjegyzés
CALL EAA2H	Verem (IY)	Regiszter (HL)	IY=IY-9
CALL EAC6G	Regiszter (DE)	Verem (IY)	IY=IY+9
CALL EAD5H	Regiszter (DE)	Verem (IY)	IY marad
CALL FFA63H	Verem (IY)	Szimbólumtábla (HL)	IY=IY-9
CALL FB28H	Szimbólumtábla (HL)	Verem (IY)	IY=IY+9
RST 18H 06H	Verem (IY)	X regiszter	IY=IY-9
RST 18H 07H	Verem (IY)	Y regiszter	IY=IY-9
RST 18H 08H	X regiszter (IY)	Verem (IY)	IY marad
RST 18H 09H	Y regiszter (IY)	Verem (IY)	IY marad
RST 18H 0AH	X regiszter (IY)	Verem (IY)	IY=IY+9
RST 18H 0BH	Y regiszter (IY)	Verem (IY)	IY=IY+9

Példaként az alábbi programrészlet megcseréli az X és Y aritmetikai regiszterek tartalmát:

```

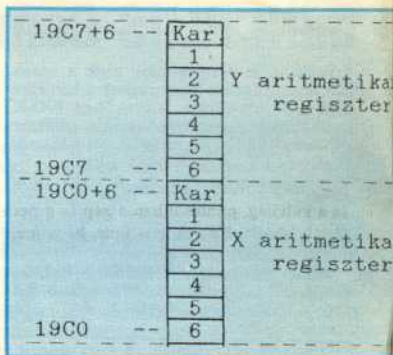
RST 18H
06H X regiszter tartalma a verembe (IY=IY-9)
07H Y regiszter tartalma a verembe (IY=IY-18)
0AH Veremből az X regiszterbe (IY=IY-9)
8BH Veremből az Y regiszterbe (IY=eredeti érték)
    
```

A 8BH hatásában megegyezik 0BH-val, de a szubrutinhívások végét jelzi.

1. lista

```

10 LOMEM 8000 : C=0 : D=0
20 FOR A=0 TO 7
30 READ B:POKE 7000+A,B:NEXT
40 C=VARPTR(B)
50 PRINT "Sorszám, Konstans"
60 FOR A=0 TO 39
70 POKE 7003,A : D=USR(7000,C)
80 PRINT A,B
90 IF INKEY$="" THEN 90
100 NEXT
110 DATA 229,223,133,39,225,195,40,251
    
```



4. ábra. A beépített X és Y aritmetikai regiszterek

A táblázatban nem szereplő további adatmozgató rutin, az RST 18H 0CH

hatására a veremben levő szám még egyszer rákerül a veremre, önmaga fölé, az IY pedig a másolt számról mutat.

Végül az utolsó adatmozgató rutin a ROM-ban tárolt, megadott sorszámú konstans a verembe teszi. A sorszámok 0-39-ig terjedhetnek; a hozzájuk tartozó konstansok értéke az 1. listán levő programmal határozható meg.

A következő programrészlet a veremben egymás fölé helyezi a 4-es, 23-as és a 16-os decimális sorszámú konstansokat:

```

RST 18H
05H A szubrutin sorszáma
04H A 4-es sorszámú konstans a verembe kerül (IY=IY-9)
05H A szubrutin sorszáma
17H A 23-as sorszámú konstans a verembe kerül (IY=IY-18)
85H A szubrutin sorszáma befejezőkor
10H A 16-os sorszámú konstans a verembe kerül (IY=IY-27)
... A processzor ezt a kódot már utasításként kezeli
    
```

A 2. lista az USR függvény felépítését mutatja.

BOTTYÁN ZOLTÁN

2. lista

```

1B58H PUSH HL
RST 18H
85H
POKE 7003,A → ...
POP HL
JP FB28H
    
```

sorszám n Az n-edik sorszámú szubrutin végrehajtása után a szub-

TechnoMIR

Egy interfészrendszernek az a feladata, hogy kapcsolatot teremtsen a számítógép és a külvilág, pontosabban a gép és a perifériák között. Ez akkor is igaz, ha a megszokott perifériákra (képernyő, nyomtató stb.) gondolunk. Így aztán, ha van egy számítógépünk, biztosan kapunk hozzá legalább egy interfészt: azt, amelyik a gépet és a billentyűzetet kapcsolja össze.

Miért mondtuk tehát az előző részben azt, hogy a felhasználás minősége nem a számítógép típusának függvénye? Azért, mert így van! Ezek az interfészek több értelemben is specifikáltak: számítógép-specifikusak (erre még visszatérünk) és periféria-specifikusak.

Ez utóbbi érthető. Minél „emberszabásúbb” egy periféria, fizikailag annál összetettebb a feladata. Például a billentyűzet nem egy, hanem kb. 60 kapcsoló; működését bonyolult elektronika, az erre szolgáló interfész vezérli. Ez nyilvánvalóan nem lehet azonos a tévékészülék vagy monitort működtető áramkörrel.

Ennek alapján kimondhatjuk: mivel egy interfész egy és csak egy összetett perifériát tud kezelni, minél egyszerűbb a periféria, annál többféle interfész képes azt kiszolgálni, és megfordítva: egy interfész annál többféle perifériát kezelhet, minél egyszerűbbek azok. (Ez ellen hatnak a szabványosítások, de nagy számuk miatt az állítás — sajnos — jó megközelítése a valóságnak.)

Így aztán, ha számítógépünkkel közvetlenül a külvilág információit akarjuk feldolgoztatni, szükségünk van egy vagy több érzékelőre és az ezekhez alkalmazható interfészre. Ez elég ijesztően hangzik, de minél egyszerűbb az érzékelőnk, annál egyszerűbb a rendszer. Az egyszerű érzékelő jeleit viszonylag egyszerű felépítésű, és az előbb kimondott állítás szerint — ez itt a lényeg! — több érzékelőhöz használható interfész képes feldolgozni. Előny az egyszerűséghez általában tartozó olcsóság, hátrány a pontatlanság. Ez az a kategória, amire jogosan mondják: iskolai demonstrációs célokra kitűnő.

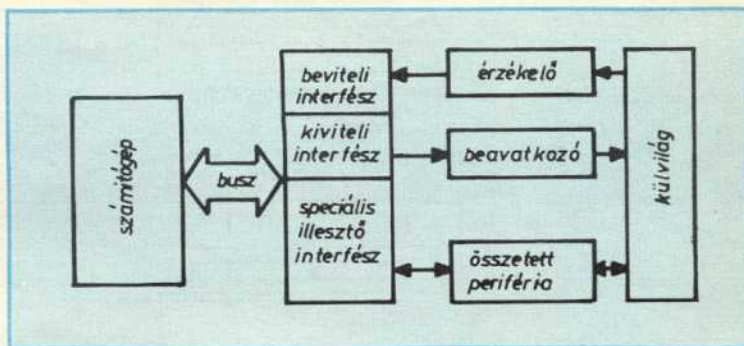
Ide tartozik a TechnoMIR rendszer is, különösen ha a másik előnyt is szem előtt tartjuk: az iskolába telepített géptípusok közül a legtöbbször csatlakoztatható. A rendszer elvi és fizikai felépítését az 1. és a 2. ábra szemlélteti.

Jól látható, hogy a TechnoMIR rendszer a számítógépeket két csoportba sorolja: az egyikhez szükség van egy speciális modulra is. A hozzáértőbbek látják, hogy a processzor fajtája határozza meg a csatlakozás módját. A rendszer Z80 alapú gépekhez, konkrétan a HT-gépekhez készült. Az, hogy a másik elterjedt processzorcsaládhoz is illeszthető, szintén az egyszerűség következménye: főleg az igen egyszerű és szellemes címzési módé.

ellátására alkalmas összeállítást nem bírnia el, külső táplálást biztosít.

- Digitális bemenet, DINP (digital input). 8 darab TTL-szintű bináris jel fogadására alkalmas, aktív bemenetekkel rendelkező egység.

- Digitális kimenet, DOUT (digital output). 8 darab TTL-szintű bináris jelet szolgáltató, több változatban kapható, aktív és passzív kimeneteket tartalmazó egy-



1. ábra

Mivel a különböző számítógépek különböző memóriacímeken engedélyeznek felhasználói bővítéseket, az egységek címzeit a HT-gépeken használt címekkel adjuk meg. A többi számítógépen használható címzés:

Primo: HT + 192
 ZX-Spectrum: 32 + HT + 31
 Commodore 16: HT + 65024
 Commodore Plus/4: HT + 65024
 Commodore 64: HT + 56832

Például a következő részben részletesen tárgyalt DINP egység HT-címe 24. Primón ez a cím 216, mert $24 + 192 = 216$.

Jelenleg a készletben a következő egységek kaphatók:

- Külső óra, HCLK (hardware clock). Csak Z80 alapú gépekhez illeszkedő, programozható számláló. Külső óraként is használható.

- Tápegység, POW (power). Ha a számítógép belső tápegysége a kívánt feladat

ség. Változatai az aktív kimenetek számában különböznek.

- Megszakítást kezdeményező bemenet, ITIN (interrupt input). Ugyanolyan, mint a DINP, de a beérkező jel megszakítást kezdeményez. Gépi szintű programozási ismereteket igénylő egység.

- Digitális—analog átalakító, D/A (digital/analog converter). 3 darab, egyenként 10 volt feszültségtartományban programozható kimenetet tartalmazó egység.

- Analóg—digitális átalakító, A/D (analog/digital converter). Több változatban kapható, meghatározott feszültségtartományon belül digitális feszültségmérésre alkalmas egység. Érdekessége, hogy a mérési tartomány a program segítségével elcsúsztható.

- Analóg sokszorozó, AMUX (analog multiplexer). Hat csatornán analog jeleket képes fogni, és programból kiválasztható, hogy melyik jelet továbbítsa az A/D egységnek.



Betöltő

Akik szeretnek Simon's BASIC-ben programozni, tudják, hogy elég bonyolult ezzel a nyelvvel dolgozni: minden alkalommal be kell tölteni a Simon's BASIC-et, elindítani, majd betölteni a programot és ezt is végzi:

- 1-7 A menü képernyőre írása. Szövege természetesen más is lehet
- 8 A billentyűzetről csak az 1-est vagy a 2-est fogadja el
- 10 A puffer maximálisra állítása
- 20 Adatok betöltése a billentyűzetpufferba
- 30 A képernyőre kikerülő betűk ASCII kódja
- 40 A Simon's betöltése

A programokat 0-tól 9-ig sorszámozva lehet kimenteni lemezre. Például: SA-VE"6",8. Természetesen a menüben ehhez a számhoz is kell egy programnevet rendelni. Úgy is átalakíthatjuk, hogy nem számot, hanem karakter nevű programot töltsön be:

8 IF A4="6" THEN A4="D"
A lényeg az, hogy a program neve csak egy karakter lehet.

Darazsak

A gimnáziumok I. osztályos fizikájához kapcsolódó program a részecskék eloszlását szimulálja 8 darazszal. Két szobában 4-4 darázs van. A két szoba között kinyitjuk az ajtót, és a darazsak elkezdenek röpködni egyik szobából a másikba. A program megvizsgálja és kijelzi az eloszlások számát, erről grafikont készít, amit ki is lehet nyomtatni. A program működése:

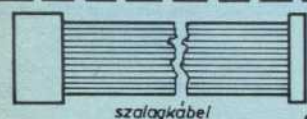
- 10-140 Sprite-ok kikapcsolása, kiírások, adatok bekérése
- 150 GOSUB 580
- 580-990 A sprite-ok definiálása, a szobák kirajzolása, a darazsak elhelyezése
- 1000 RETURN
- 160-240 A röptetendő darázs kiválasztása
- 200 GOSUB 1010, zümmögés
- 210 GOSUB 470, röptetés: a bal oldali szoba egyik darazsát a jobb oldalba
- 220 GOSUB 520, a bal szobából röptetés a jobb szobába
- 250-290 Az egyes állapotok kiírása
- 300-400 Grafikonrajzolás
- 410-440 GOSUB 1100, nyomtatás



C16
C64

szalagkábel

illesztőmodul



HT
PRIMO
ZX

szalagkábel

interfészmodulok

mérőszinór



tápegység

2. ábra

- Próbapanel, PB (proto board). 3 x 8 bit programozható ki- vagy bemenettel és tápfeszültségpontokkal ellátott, speciálisan kialakított felületű egység. Egyszerűbb digitális kapcsolások építhetők vele, amelyek a számítógéppel vizsgálhatók is.

- Botkormányillesztő, JOY (joystick). Olyan géphez való, amelynek nincs ilyen csatlakozója.

Tudomásunk szerint több érdekes modul áll fejlesztés vagy gyártás-előkészítés alatt; például az a léptetőmotor-kezelő modul, amely a készülő iskolarobotot fogja az iskolai számítógépek segítségével vezérelni.

Az általános áttekintés után a következő részben elkezdjük a modulok konkrét ismertetését.

ALBU LÁSZLÓ-KIRÁLY LÁSZLÓ

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.

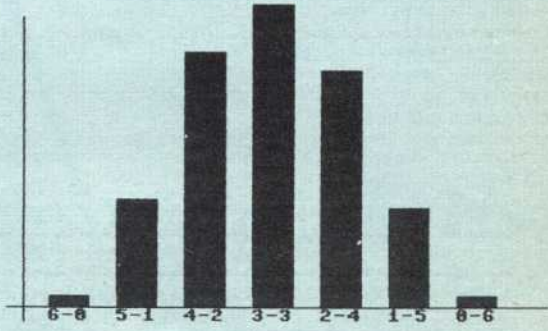
**Minden kedden
17-től 20 óráig
ENTERPRISE-klub
a VSZM
Közösségi Házban**

(Bp. XI., Fehérvári út 120.)
Klubvezető: Romvári Gábor


```

440 IF A$((">"N" THEN420
450 RUN
460 STOP
470 RLOCMB G,136,106,0,S
480 RLOCMB G,152+INT(RND(0)*80),74+INT(RND(0)*80),0,S
490 A=A-1:B=B+1:H(A)=H(A)+1:X(G)=1
500 PRINTAT(7,19)A:PRINTAT(21,19)B
510 RETURN
520 RLOCMB G,136,106,0,S
530 RLOCMB G,40+INT(RND(0)*80),74+INT(RND(0)*80),0,S
540 A=A+1:B=B-1:H(A)=H(A)+1:X(G)=0
550 PRINTAT(7,19)A:PRINTAT(21,19)B
560 RETURN
570 STOP
580 POKE53281,5:PRINT"▣"
590 DESIGN 1,64*128
600 @.....B.....
610 @.....B.....
620 @....CBC.....
630 @....CBC.....
640 @....BBB.....
650 @.....B.....
660 @...DCCCD...
670 @..DDBBBD...
680 @..DDCCDD...
690 @.DDDBBDD...
700 @.DDBCCDD...
710 @.DDBBBDD...
720 @DDDDCCDDDD
730 @DDDBBDDDD
740 @DDDDCCDDDD
750 @DDDBBDDDD
760 @DDDDCCDDDD
770 @DDDBBDDDD
780 @DDD.CCC.DDD
790 @.D...B...D..
800 @.....
810 MOB SET 1,128,7,1,1
820 MOB SET 2,128,7,1,1
830 MOB SET 3,128,7,1,1
840 MOB SET 4,128,7,1,1
850 MOB SET 5,128,7,1,1
860 MOB SET 6,128,7,1,1
870 CMOB 0,3
880 PRINTAT(1,1)"▣"
890 PRINTAT(1,16)"▣"
900 FORI=2T015:PRINTAT(1,I)"▣" ▣ ▣
910 FORI=1T03
920 MMOB 1,56,74+J,56,74+J,0,0
930 MMOB I+3,96,74+J,96,74+J,0,0
940 J=J+32:NEXTI
950 D$(1)=" [ ] "
960 D$(2)=" | 0 | "
970 D$(3)=" [ ] "
980 FORI=1T03
990 PRINTAT(6,17+I)D$(I):PRINTAT(20,17+I)D$(I):NEXTI
1000 RETURN
1010 REM ZENE-----
1020 SI=54272:FL=SI:FH=SI+1:W=SI+4:AA=SI+5:HH=SI+6:LL=SI+24
1030 POKEW,0:POKEAA,0:POKEHH,0
1040 POKELL,15
1050 POKEAA,16*5+16
1060 POKEHH,0

```



```

1070 POKE FH,F1:POKEFL,F2
1080 POKEW,33
1090 RETURN
1100 OPEN 4,4
1110 PRINT#4," *****"
1120 PRINT#4," *** DARAZSAK ***"
1130 PRINT#4," *****"
1140 PRINT#4:PRINT#4
1150 PRINT#4," ROFTTESEK SZAMA"V
1160 FOR I=0TO6:PRINT#4," "G-I"- "I": "H(I):NEXT I
1170 PRINT#4:CLOSE 4
1180 COPY
1190 RUN

```

Energiaeloszlás

1070,1080,1090,1100

```

0 PRINT " " : POKE 53281, 11 : POKE 53280, 11
1 PRINT " " : CENTRE " *** J.A.G. *** " : PRINT " " : CENTRE " ENERGIA ELOSZLAS "
2 PRINT " " : CENTRE " *** TOROK GYORGY *** "
3 PRINT " " : CENTRE " (C) 1986 "
5 GETA$: IFA$ = " THEN 5
7 DIM B(500,7) : R$ = CHR$(8) : T$ = CHR$(15)
8 GOSUB 600 : PRINT " "
10 A$(1) = " "
20 A$(2) = " | "
30 A$(3) = " "
40 FOR I=0TO2
50 PRINT AT(I*13+1,0)A$(1)
60 PRINT AT(I*13+1,12)A$(1)
70 FOR J=0TO8
80 PRINT AT(I*13+1,J+1)A$(2)
90 PRINT AT(I*13+1,J+13)A$(2)
100 NEXT J
110 PRINT AT(I*13+1,10)A$(3)
120 PRINT AT(I*13+1,22)A$(3)
130 NEXT I
150 DESIGN 0,13*64
160 @.....BBBB.....
162 @.....BBBBBBBB.....
164 @.....BBBBBBBBBB.....
166 @.....BBBBBBBBBBBB.....
168 @.....BBBBBBBBBBBBBB.....
170 @.....BBBBBBBBBBBBBBB.....
172 @.....BBBBBBBBBBBBBB..B.....
174 @.....BBBBBBBBBBBB..BB.....
176 @.....BBBBBBBBBBB.BBB.....
178 @.....BBBBBBBBBBB.BB.....
180 @.....BBBBBBBBBB.BBB.....
182 @.....BBBBBBBBBB.BBB.....
184 @.....BBBBBBBBBBB.BB.....
186 @.....BBBBBBBBBBB.BBB.....
188 @.....BBBBBBBBBBB.BB.....
190 @.....BBBBBBBBBBBB..B.....
192 @.....BBBBBBBBBBBBBBB.....
194 @.....BBBBBBBBBBBBBBBB.....
196 @.....BBBBBBBIBBBBBBB.....
198 @.....BBBBBBBBBBBB.....
200 @.....BBBBBB.....
210 MOB SET0,13,7,0,0
220 MOB SET1,13,7,0,0
225 MOB SET2,13,7,0,0
230 MOB SET3,13,7,0,0
235 MOB SET4,13,7,0,0
240 MOB SET5,13,7,0,0
245 MOB SET6,13,7,0,0
250 FOR I=1TO6:F(I,I)=1:READX(I),Y(I):Z(I)=1
252 MMOBI,X(I),Y(I),X(I),Y(I),0,0
253 NEXT
255 REM-----
256 REM-----
300 FORU=1TO01
302 FORU=1TO02
305 R1=INT(RND(0)*6)+1:R2=INT(RND(0)*6)+1
310 IF Z(R1)=0ORR1=R2THEN305
320 FOR I=1TO6
330 IF F(R1,I)=1 THENH=I:GOTO350
340 NEXTI:GOTO300
350 F(R1,H)=0:Z(R1)=Z(R1)-1
360 Z(R2)=Z(R2)+1:F(R2,H)=1
365 YY=INT(RND(0)*40):XX=INT(RND(0)*40)
370 RLOC MOB H,X(R2)+XX,Y(R2)+YY,0,0
380 NEXTU:PRINT AT(18,24)U;
390 FORI=1TO6
400 E(I)=E(I)+Z(I)
405 B(U,I)=Z(I):NEXTI

```

```

410 NEXTU:FOR I=1T06:MOB OFF I:NEXTI:PRINT"U":GOTO487
425 PRINT"U";
430 CENTRE"ELOSZLAS":PRINT
435 PRINT"-----"
440 PRINT"| DOBAS | 1. | 2. | 3. | 4. | 5. | 6. |"
445 PRINT"-----"
450 FOR I=1T016
460 PRINT" | | | | | | |"
470 NEXT
475 PRINT"-----"
477 PRINT"-----";
481 PRINTAT(0,18)" | | | | | | |"
482 PRINTAT(1,19)"ATLAG"
485 PRINT"-----:RETURN
487 GOSUB425
490 FOR II=1T001
500 PRINTAT(1,6+6)II*02
510 PRINT"-----"B(II,1)"-----"B(II,2)"-----"B(II,3)"-----"B(II,4)"-----";
520 PRINTB(II,5)"-----"B(II,6)
530 G=G+1:IFG=11THENG=0:GOSUB1100
540 NEXTII
545 FOR I=1T06:E(I)=INT((E(I)/01)*100)/100
546 PRINTAT(I*5+2,19)RIGHT$( " "+STR$(E(I)),4):NEXTI
547 W8=0:FOR I=1T06:W8=W8+E(I):NEXTI:YU=W8/6
548 PRINTAT(6,23)"AZ ATLAGOK ATLAGA:";INT(YU*1000)/1000
550 GET A$:IFA$(("<del>-----</del>")) THEN550
560 PRINT"-----INYOMTASSAM AZ ADATOKAT (1/N)?"
570 GETA$:IF A$="I"THEN2000
580 IFA$(("<del>-----</del>")) THEN570
590 RUN
600 REM*****
610 PRINT"U"
620 INPUT"-----HANYSZOR DOBJAK";O1$:O1=VAL(O1$):IF O1=0 THEN620
630 PRINT"-----HANYSZORAVALL JELEZZEM KI":INPUT"-----AZ EREDMENYT";O2$
635 O2=VAL(O2$):IF O2=0 THEN630
640 RETURN
997 REM-----
998 REM-----
999 REM-----
1000 DATA56,58,152,58,260,58,56,154,152,154,260,154
1100 PRINTAT(16,22)"SPACE=>TOVABB"
1110 GETA$:IFA$(("<del>-----</del>")) THEN1110
1120 GOSUB425:RETURN
2000 PRINT"U";
2005 OPEN 4,4
2010 PRINT#4," ELOSZLAS"
2020 PRINT#4,"-----";R$
2030 PRINT#4,T$:"| DOBAS | 1. | 2. | 3. | 4. | 5. | 6. |";R$
2040 PRINT#4,T$:"|-----|";R$
3030 FOR II=1T001
3040 PRINT#4,T$:"| II*02;
3050 PRINT#4,CHR$(16);"06";" | "B(II,1)"| "B(II,2)"| "B(II,3)"| "B(II,4)"| ";
3060 PRINT#4,B(II,5)"| "B(II,6)"| ";R$
3080 NEXTII
3082 PRINT#4,T$:"|-----|";R$
3084 PRINT#4,T$;CHR$(16);"06";
3090 FOR I=1T06:E(I)=INT((E(I)/01)*100)/100
4000 PRINT#4,RIGHT$( " "+STR$(E(I)),5):NEXTI
4002 PRINT#4,CHR$(16);"00";" | | | | | | |";R$
4010 PRINT#4,T$:"|-----|";R$
4012 PRINT#4:PRINT#4,"AZ ATLAGOK ATLAGA:";INT(YU*1000)/1000:PRINT#4:PRINT#4
4015 CLOSE4
4020 RUN

```

Nézzük az előző számunkban megjelent és egyben utolsó forduló megfejtését. A kérdéseket általános témákból merítettük: operációs rendszer, programozás, adatbázis-kezelés stb. Aki az első kérdésre I-est tippelt, majdnem eltalálta a helyes választ, mert a számítógép operációs rendszer nélkül az esetek túlnyomó többségében nem üzemeltethető. Azonban létezik olyan üzemmód és készítenek olyan programokat, amikor a gépet alap operációs rendszer nélkül is lehet használni (X).

A számítógép bekapcsolását követően az operációs rendszert tartalmazó szoftvert a hajlékonylemez-meghajtón keresi, és ha ott nem találja, akkor automatikusan a merevlemezről olvassa. Mivel a hajlékonylemezek elég sérülékenyek, ezért általában, hogy a merevlemezre megrendelkező gépeknek az operációs rendszert azon helyezik el, illetve rámásolják. Ez azonban nem zárja ki, hogy hajlékonylemezről is betölthető (2).

A jelenleg gyártott digitális számítógépek működési elve a Boole-algebrán alapul. Nem szándékozunk jóslatokba bocsátkozni, de valószínűleg ez még sokáig így lesz (X).

A hexadecimális, azaz tizenhatos számrendszer tizenhat jelből áll. A számítástechnikában használt általános konvenció szerint az arab számjeleket az angol ábécé betűivel egészíti ki. Így a „G” nem fordulhat elő (X).

Az adatbázis szempontjából a mező a rekordnak az a része, amely az adatfeldolgozás szempontjából egységnek kezelhető. A rekord egymással kapcsolatban álló adatok (mezők) egységként kezelhetők. A fájl egymással kapcsolatban álló adatok szervezett halmaza (2).

Az ellenőrző szám a hibásan bevitt, begépetelt adatok kiszűrésére szolgál. Képzése az egyszerű összeadástól a komplikált logikai elven felépített rendszerrig terjedhet. Szűkebb értelemben a paritás-ellenőrzés is ebbe a témakörbe tartozik (1).

A Chapin kártya a folyamatábrához hasonló programozási segédeszköz (2).

A félvezetők és integrált áramköri megvalósításai a modern számítógépek építőelemei. A logikai áramkörök felépítésénél a félvezetőknek azon tulajdonságait használják, hogy két állapotuk van. Vagy van jel („igen”), vagy nincs jel („nem”) (2).

Szekvenciális adatállományt általában mágnesszalagon hoznak létre. Indexelt szekvenciális adatállomány azonban csak mágnesszalagon valószínűleg meg (1).

A disassembler gépi kódból assembler nyelvű programot állít elő (X).

Az algoritmus olyan szabályokból áll, amelyek egy feladat végrehajtásának egymást követő lépéseit határozzák meg. Algoritmizálható feladatokra minden esetben program is készíthető. Bebizonyították, hogy léteznek olyan matematikai, kibernetikai problémák, amelyek nem algoritmizálhatók (1).

A PS/2 (Personal System) az új IBM gépcsalád elnevezése (2).

A parancsot bevitelének pillanatában végrehajtja a számítógép (például LIST), az utasítást pedig akkor, amikor a vezérlés rákerül, a program futása közben (1).

Az MS-DOS operációs rendszernél a DOS parancs segítségével vissza lehet térni az alap operációs rendszerbe a futó programból való kilépés nélkül. A programba az EXIT parancsral térhetünk vissza (1).

E helyesen kitöltött szelvény:

X 2 X X 2 1 2 2 1 X 1 2 1 1

Az ember elgépiesedik —

Az ember, az emberi munka elgépiesedésének témaköre a humor egyik kifoghatatlan ötlettára. Sokan hitték, hiszik ma is, hogy az emberiség — az emberség — egyik nagy ellensége a számítógép. Miközben erről értekeznek, a számítástechnikusok jóvoltából egyre intelligensebb, egyre inkább emberszabású számítógépek, okos programok készülnek. Komoly eredményeket értek el a mesterséges intelligencia kutatásában, s a számítógépek megértik az emberi szót, válaszolnak, kézírást olvasnak, kottából muzsikálnak; partnereink lettek nemcsak a munkában, de a szórakozásban is.

Az ember elgépiesedik — a gép „elemberiesedik”. Összecsap a mesterséges intelligencia a természetes értelemmel. Mi lesz ebből a párviadalból? Erre várunk választ a világ karikaturistáitól és rajzos kedvű számítástechnikusaitól.

A pályázatot a Neumann János Számítógéptudományi Társaság hirdeti meg. A kiállítás és díjkiosztás a Tavasz Fesztivál '88 rendezvényei között szerepel, 1988 márciusában.

Két kategóriát hirdetünk meg. Az elsőbe a hagyományos eszközökkel (ceruza, toll, ecset stb.) dolgozó karikaturisták műveit várjuk. A második kategóriába azok a karikaturisták és számítógépes szakemberek,



Karikatúra-pályázat

amatőrök jelentkezhetnek, akik műveiket számítógép segítségével készítették.

A pályázók mindkét kategóriában 3-3 alkotásukkal vehetnek részt. A második kategóriában kérjük a szerzőket, hogy műveiket vagy nyomtatva, vagy fényképen (tehát közvetlenül kiállítható formában) küldjék be.

A pályázat nyílt: bárki indulhat, aki művét a határidőig (1987. dec. 31.) elküldi.

a gép elemberiesedik

A képek méretét nem kötvük meg, hiszen annak sokszor a technika szab határokat.

A pályázók nevük, lakcímük mellett írják meg korukat és foglalkozásukat is. A második kategória szerzőitől kérjük a géptípus, illetve a felhasznált program(ok) megnevezését, saját program esetén ennek feltüntetését.

A műveket nemzetközi zsűri értékeli. Egy nagydíjat adunk ki, valamint kategóriánként egy-egy első, második és harmadik díjat. Ezek mellett néhány különdíjat is kiosztunk.

A zsűri által a kiállításra kiválasztott karikatúrákból reprezentatív albumot jelentünk meg, melyet a benne szereplő pályázók ajándékba megkapnak, mások számára pedig a kiállításon kezdjük meg az árusítást. Az album postán is megrendelhető (SZÁMALK Kiadó, Budapest, Bartók Béla út 104. 1113).

A beküldött karikatúrákat nem küldjük vissza. A kiállítás utolsó napján a műveket árverésre bocsátjuk, s a bevételt testi fogyatékosok számítástechnikai munkalehetőségei megteremtésére fordítjuk.

A pályaműveket az alábbi címre kérjük: Neumann János Számítógéptudományi Társaság, Budapest, Báthory u. 16. 1054.

A kiállítás szervezője: dr. Halász Géza, SZÁMALK. Tel.: 851-011.

BASIC és gépi kód

Legutóbb a BIT és a NOP utasításról volt szó. Most egy újabb utasításcsoporttal ismerkedhetünk meg, a veremkezelő utasításokkal. Ez a csoportosítás egy kissé önkényes, mert általában csak a PHA, PHP, PLA és PLP utasításokat szokták ide számítani, tágabb értelemben viszont a JSR és RTS utasításokat is ide soroljuk.

A veremkezelő utasításokat nagyon körültekintően kell használni, mert a legkisebb gondatlanság is kellemetlen következményekkel járhat, amin csak a gép kikapcsolásával, vagy szerencsés esetben a RESET gomb használatával tudunk segíteni.

A PHA

Neve az angol „Push Accumulator on stack” kifejezés betűiből származik. Ennek megfelelően az A regiszter (másik néven akkumulátor) tartalmát a processzorverembe másolja, a veremmutató által meghatározott címre, majd a veremmutató tartalmát eggyel csökkenti. Az állapotbitek értékét nem módosítja.

A PLA

Neve az angol „Pull Accumulator from stack” kifejezés betűiből keletkezett. A veremmutató tartalmát eggyel megnöveli, majd a veremmutató által meghatározott bajt tartalmát az A regiszterbe tölti. A Z és N állapotbitek az A regiszternek az utasítás végrehajtása utáni tartalmát függően állítodnak be az ismert módon.

A PHP

Neve az angol „Push Processor status on stack” kifejezés betűiből származik. Az állapotregiszter tartalmát a verembe másolja, a veremmutató által meghatározott címre, majd a veremmutató tartalmát eggyel csökkenti. Az állapotbitek értéke nem változik meg.

A PLP

Neve az angol „Pull Processor status from stack” kifejezés betűiből keletkezett. A veremmutató tartalmát eggyel megnöveli, majd a veremmutató által meghatározott bajt tartalmát az állapotregiszterbe tölti.

A TXS

Az X regiszter tartalmát a veremmutató regiszterbe másolja. Az állapotbitek értéke nem változik. Ezzel az utasítással tudjuk a veremmutató értékét módosítani. Alkalmazására az 1. listán láthatunk példát.

A TSX

A veremmutató tartalmát az X regiszterbe másolja. A Z és N állapotbitek az X regiszter tartalmának megfelelően állítodnak be. Ennek az utasításnak a segítségével tudunk programból a veremmutató tartalmához hozzáférni. Alkalmazására a 2. listán láthatunk egy példát.

A BRK

Neve az angol „BReak” (törés, megszakítás) szó betűiből származik. A B és I állapotbiteket l-re állítja, azután az utasításszámláló és az állapotregiszter tartalmát a verembe teszi, majd az IRQ vektor tartalmát az utasításszámláló regiszterbe tölti. Az IRQ vektor a SFFFE...SFFFE címeken található, a megszakításkezelő rutin címét tartalmazza. Erről az I bitet állító utasítások ismertetések (Mikroszámítógép Magazin 1987/5. BASIC és gépi kód) már volt szó.

Az RTI

Neve az angol „ReTurn from Interrupt” (visszatérés megszakításból) betűiből származik. A feltételbitek állításától eltekintve ugyanazt csinálja, mint a BRK, de fordított sorrendben és fordított irányban. Az RTS utasítástól csak annyiban különbözik, hogy a veremből először az állapotregiszter korábbi tartalmát tölti vissza, és csak azután tölti az utasításszámlálóba — az RTS-hez hasonlóan — a visszatérési címet.

A programlistákról

A mostani programlistákon a C64 KERNAL ROM-jának egy-egy környezetéből kiemelt részlete látható. Ezek a másik két géptípuson is megtalálhatók, az ittenitől eltérő címeken, a VC20-n pontosan megegyező formában, a C16-on apró — a lényegét nem érintő — eltérésekkel.

fce2	a2ff	ldx	#\$ff
fce4	78	sei	
fce5	9a	txs	
fce6	d8	cld	

1. lista

ff48	48	pha	
ff49	8a	txa	
ff4a	48	pha	
ff4b	98	tya	
ff4c	48	pha	
ff4d	ba	tex	
ff4e	bd0401	lda	\$0104, x
ff51	2910	and	#\$10
ff53	f003	beq	#\$f58
ff55	6c1603	jmp	(\$0316)
ff58	6c1403	jmp	(\$0314)

2. lista

3. lista

ea81	68	pla	
ea82	a8	tay	
ea83	68	pla	
ea84	aa	tax	
ea85	68	pla	
ea86	40	rti	

Az 1. listán a gép bekapcsolása után végrehajtott első négy utasítást látjuk. Ezekre korábban már többször hivatkoztam. Itt található a veremmutató kezdőértékének beállítása: a \$FCE5 című utasítás végrehajtása után a mutató a processzorverem utolsó bajtjának címére, \$01FF-re mutat, ami azt jelenti, hogy a verem üres. Remélem, senki nem feledkezett meg arról, hogy a veremutató regiszternek csak az alsó 8 bitjéhez tudunk hozzáférni, a 9. biten mindig 1 van, ezért mindig a \$0100...\$01FF (256...511) tartomány címezhető vele.

A 2. listán a C64 IRQ rutinjának az a része látható, amely az A, X és Y regiszter tartalmának a mentése után aszerint határozza meg a megszakítási rutin futásának folytatását, hogy a megszakítást hardveresemény vagy BRK utasítás idézte elő.

A rutinba való belépéskor az utasításszámláló és az állapotregiszter korábbi tartalma már e mellett van. A többi regiszter tartalmát is el kell menteni, mert a megszakított program futásának folytatásához az eredeti regisztertartalmakat helyre kell állítani. Láthatjuk, hogy csak az A regiszter tartalmát tudjuk közvetlenül a verembe tenni, az indexregisztereket csak az A regiszteren keresztül.

A rutin a regiszterek mentése után az állapotregiszter korábbi állapotát vizsgálja meg. Ehhez a veremmutató tartalmát az X regiszterbe tölti. Ezután arról a címről tölt egy bajtot az A regiszterbe, amely \$0104-nek mint báziscímnek és az X indexregiszter tartalmának az összege.

Vizsgáljuk meg, hogy miért éppen \$0104 a báziscím. Tudjuk, hogy X-ben a veremmutató aktuális értékének alsó bajtja van, ami a tényleges veremcíműl \$0100-zal tér el. Ezt az értéket báziscímnek tekintve, megkaphatnánk a verem következő szabad helyének jelenlegi tartalmát, itt nyilvánvalóan valamilyen — általában a korábbi veremhasználatból származó — „hulladékot” találunk. A báziscímek eggyel-eggyel növelve, egymás után férhetünk hozzá a verembe korábban letett értékekkel, jelen esetben sorrendben az Y, az X, az A, majd az állapotregiszter mentés előtti tartalmával. Az első hármat a listában látható rutin, az utolsó a BRK utasítás tette a verembe. Az ujjaink is kiszámolhatjuk, hogy ez utóbbihoz a \$0104-es báziscímnek keresztül férhetünk hozzá.

A \$FF51 című utasítás végrehajtása előtt az A regiszterben az állapotregiszternek a megszakítás előtti tartalmának megfelelő érték van. A BRK által előidézett megszakítás abban különbözik a hardver IRQ által okozottól, hogy az előbbi az állapotbajt B bitjét l-re állítja, míg az utóbbi nem. A segédletben (Mikroszámítógép Magazin, 1986/3. A 65XX mikroprocesszor gépi és assembler szintű utasításkészlete) láthatjuk, hogy a B az állapotregiszter 4. bitje, ennek az állapotot kell megvizsgálni. Az AND #\$10 utasítás a B eredeti értékével ellentétesen állítja be a Z bit mostani értékét, és ettől függően történik vagy nem történik ugrás \$FF58-ra.

A \$0314...\$0315 című bajtkon a hardver által előidézett megszakítást kezelő rutin címe van elhelyezve, a \$0316...\$0317 címeken pedig az úgynevezett BRK rutiné. Ezeket a 3-as memórialap elején lévő címeket vektoroknak nevezik, ismertetésükre hamarosan sor kerül.

A 3. listán a hardver megszakítási rutin befejező utasításait látjuk. Az előbbieket ismeretében szükségletlen a magyarázat. A regisztertartalmak visszatöltésének sorrendje fordítottja a mentés sorrendjének.

BARNA LÁSZLÓ

Z80 programok haladóknak Spectrumra és Primóra

Az előző részben láttuk, hogyan lehet zárt alakzatot teljesen befesteni. Tapasztalatainkat és a befestőprogramot felhasználva, már egyszerűen megoldhatjuk a mintával való festés problémáját.

Az 1. listán a mintás festést megvalósító FILL nevű program látható. Sebessége a PAINT-tal gyakorlatilag megegyező. Inverz zárt alakzat mintás festését is hibátlanul elvégzi.

Az 1. lista mellé be kell még tölteni az előző részben szereplő PAINT rutint, valamint az 1. részben szereplő gyors LDIR-t (LDIR0). A PAINT rutinban néhány változtatást kell elvégezni, ezek a 2. listán láthatók. Primo-tulajdonosok a képernyőcím megadására (SCREEN EQU) ügyeljenek.

A program használata

Az indulási pont koordinátáit, akárcsak a PAINT rutinnál, a BC regiszterpárban kell megadni. A minta egy 8 x 8-as bitminta valahol a memóriában. Címét a DE regiszterpárba kell tölteni. Spectrumon a PROBA rutin itt is a BASIC-ből való hívást támogatja. Ekkor a minta a grafikus "U" karakter képe. (A memória utolsó 8 bájtja.) Primósok írhatnak maguknak hasonló rutint, amely a BASIC CALL utasítás paramétereit átveszi.

A program működése

A program képpuffert használ. Mérete a képernyővel egyezik meg. (Nem pazarlás ez?) A mintás feltöltés menete az ábrán követhető nyomon. A bal oldali oszlopban a művelet neve látható, amely egyben a művelet megvalósító rutinnak a címkéje is. Mellette van a művelet végrehajtása utáni kép-, illetve puffertartalom vázlata.

A STOB rutin bemásolja a képtartalmat a pufferbe. A PAINTB rutin a pufferen festi be az alakzatot. A BXORS rutin bitenkénti kizáró vagy műveletet hajt végre a kép és a puffer között, az eredménybiteket pufferbe írja vissza. Így a pufferen pontosan a feltöltendő alakzat belseje marad. A BANDM rutin a mintát sorban ráilleszti az egész pufferre, és bitenkénti és műveletet hajt végre. Az eredménybitek szintén a pufferbe kerülnek. A pufferen az alakzat belseje látható, mintásan. Az ábrán (hibásan) az alakzat kerete is szerepel, másképp nem sikerült szépen megrajzolni. Újra a BXORS rutin fut, a pufferen ekkor megjelenik a kész, feltöltött alakzat. Az EXBS rutin gondoskodik arról, hogy a képernyőre kerüljön a kész ábra.

Mint látható, a műveletek mindvégig a pufferen folynak, a kész kép csak a legvégén kerül a felhasználó csodálkozó szemé elé.

5. Festés mintával

1. lista

```

*L-
*C-
*D+
;*****
;#          #
;# FILL-3   #
;# (c)1986.11.29 UHI #
;#          #
;*****

                ORG  $F800
SCREEN EQU    $4000
BUFFER EQU   $E000
PATTER DEFS  2  !INTACIM
BUFEND EQU   BUFFER+$17FF
SCRLEN EQU   $1800
;*****
FILL  PUSH BC
      PUSH DE
      PUSH HL
      PUSH IX
      LD  (PATTER),DE
      CALL STOB
      CALL PAINTB
      CALL BXORS
      CALL BANDM
      CALL BXORS
      CALL BEXS
      POP IX
      POP HL
      POP DE
      POP BC
      RET
;*****
STOB  PUSH BC
      CALL SBREGS
      CALL LDIR0
      POP BC
      RET
;*****
BXORS CALL SBREGS
BXS1  LD  A,(DE)
      XOR (HL)
      LD  (DE),A
      INC DE
      INC HL
      DEC BC
      LD  A,B
      OR  C

```

```

JP  NZ,BXS1
RET
;*****
BANDM LD  HL,BUFFER
      LD  IX,(PATTER)
      LD  B,3
BAM1  LD  C,0
BAM2  LD  D,0
BAM3  LD  A,D
      LD  (BAM4+2),A
      LD  A,(HL)
BAM4  AND (IX+0)
      LD  (HL),A
      INC D
      BIT 3,D
      JP  Z,BAM3
      LD  A,H
      SUB B
      LD  H,A
      INC HL
      DEC C
      JP  NZ,BAM2
      LD  A,H
      ADD A,7
      LD  H,A
      DJNZ BAM1
      RET
;*****
BEXS  CALL SBREGS
BES1  LD  A,(DE)
      EX  AF,AF'
      LD  A,(HL)
      LD  (DE),A
      EX  AF,AF'
      LD  (HL),A
      INC HL
      INC DE
      DEC BC
      LD  A,B
      OR  C
      JP  NZ,BES1
      RET
;*****
SBREGS LD  HL,SCREEN
      LD  DE,BUFFER
      LD  BC,SCRLEN
      RET
*L-
*C-
*D+

```

```

;=====
PAINTB CALL TUBINI
CALL PIXADB

CALL MASKG
CALL MODE
CALL PUTTUB
PAINT1 CALL PLINE
JR NZ,PAINT1
RET
;=====
PIXADB CALL PIXADD
EX AF,AF'
XOR A
LD BC,SCREEN
SBC HL,BC
LD BC,BUFFER
ADD HL,BC
EX AF,AF'
RET
;=====
PROBA LD BC,(5C7D)
LD DE,FFFF
JP FILL
    
```

Érdekes megfigyelni, hogy a program lefutása után a pufferben éppen az eredeti, feltöltetlen ábra van. Ez igen előnyös lehet. Gondoljunk csak arra, hogy a felhasználó valamilyen grafikus szerkesztőprogrammal dolgozik, és éppen be akar tölteni egy általa rajzolt kis alakzatot. Nem veszi észre, hogy egy picurka lyukat hagyott rajta, és mit sem sejtve megnyomja a feltöltést kérő billentyűt. Ekkor a PAINT rutin, természetesen megfelelően kibújik a lyukon, és szorgalmasan feltölti az alakzat környezetét is, esetleg az egész képernyőt. Az eredmény: sokkal nagyobb terület lesz mintás, mint ahogyan azt a mesebeli felhasználónak elképzelte. Mit tesz ekkor a szerkesztlen? Nem veszett kárba a többórás rajzolgatás: megnyom egy olyan gombot, amire visszajön a pufferből az eredeti kép, és vidáman kijavíthatja hibáját. Hát ezért nem pazarlás a pufferkép használata. Egyébként egy grafikai szerkesztőprogram más rutinjai is használhatnak pufferképet, ez megegyezhet azzal.

Ezt a FILL rutint a manőszerkesztő programban fel fogjuk használni. A következő részben a manőszerkesztő további rutinjait mutatom be.

2. lista

UHERKOVICH PÉTER

művelet	képernyő	pufferkép
		? tetszőleges
STOB		
PAINTB		
EXORS		
BANDM		
EXORS		
EXBS		

Verem a Spectrum BASIC-hez

BASIC-ben programozva gyakran kerülünk olyan helyzetbe, hogy feltétlenül verem (stack) használatára volna szükségünk (például rekurzív algoritmusok kódolásakor), de van, amikor a verem használata egyszerűen csak kényelmesebbé teszi a program megírását. A verem fogalmát jól ismerik azok, akik gépi kódban is programoznak. A verem az adattárolásnak egy szervezési formája, amelyet úgy használunk, hogy adatokat írhatunk a verembe, és olvashatunk is belőle. Adatolvasáskor mindig a felső, vagyis az utoljára beírt adatot kapjuk, s az ekkor törlődik a veremből.

A listán látható program egy veremtartartat valósít meg, a Spectrum BASIC-jéhez illesztve. Adatot írni a PRINT #3 utasítással, olvasni pedig az INPUT #3 utasítással lehet. (Használhatjuk az INKEYS-t is, de nem célszerű!) A program egy CR-rel lezárt szöveget tekint egy adatnak, tehát a verembe írt szöveget nem „megfordítva” kapjuk vissza. A verembe író PRINT #3 utasítás végén ne legyen pontosvessző, több adatot pedig aposztróffal (') elválasztva írhatunk egy PRINT #3 utasítással.

A vermet az első használat előtt inicializálni kell, hiszen a felsorolt utasítások a nyomtatót működtetik. A verem inicializálása a RANDOMIZEUSR 23296 utasítással történjen. Ezután már nem használható a nyomtató, így a program is célszerűen a nyomtatópuffer-területen helyezkedik el. Az inicializálási utasítást egy BASIC programban többször is használhatjuk. Hatása: kiüríti a vermet.

Ha üres veremből próbálunk olvasni, akkor csak CR (enter) karaktert kapunk, hibajelzés nincs. A forrásszöveg 26-os sorától hibajelzést adó utasításokat is be lehet írni: RST 8, utána DEFBI hibakód—1. Ekkor RET nem kell.

A program működésében csak egyetlen érdekesebb dolog van: annak a megvalósítása, hogy olvassáskor a szöveg ne forduljon meg. Erre itt most nem térek ki. A forrásszöveg GENS3 assembleren készült.

U. P.

```

1 ;===== 47 LD (BP)+HL, IJZ, botpoint
2 18 stack Spectrum betitchez 48 DEC HL, IJZ, csokly
3 ;===== 49 LD (BL)+HL, IJZ, hossz
4 ORG 23296 ;enter buffer 50 SCF
5 DEFIH LD HL, W11E ;Ira rutin 51 RET
6 LD (2374F)+HL, Ichrout 52 WK11E LD HL, (BP)
7 LD HL, REAH ;Iolvaso rut., 53 LD DE, (LP)
8 LD (23751)+HL, Ichrin 54 OR A, Icf, torlas
9 LD HL, (RAM)DP, Iififo c 55 SBC HL, DL, IJZF, LI"
10 IMC HL, I... T, loze 56 JR Z, W11E ;Izen
11 LD (LP)+HL, I"iffo, ores" 57 W3 LD (DE)+I, Ichar, hex
12 LD (BP)+HL, I... 58 INC DL, IJZ, Ienter, nov
13 LD HL, I... 59 LD (LP)+DL, IJZ, Ienter
14 LD (LL)+HL, I... 60 LD HL, (LL), I"iffo, ossz...
15 LD (BL)+HL, I... 61 IMC HL, I...
16 RET 62 LD (LL)+HL
17 REAH RES 3, (IY+2) 63 RET
18 LD DE, (BL), Ibuff, hossz 64 W1 LD HL, (BF)
19 LD A, B, Ivezess, 0-e? 65 LD (CH)+A, Ichar, mentes
20 OR A, I... 66 LD BC, (BL)
21 JR NZ, A1 ;nem 67 LD DL, (LDG) ;Izen, cik
22 LD BC, (LL), Iiffo, hossz 68 ADU HL, B
23 LD A, B, I0-e? 69 PUSH HL, Ibuffer, vese
24 OR C, I... 70 SBC HL, DE, Ienter?
25 JR NZ, A2 ;nem 71 POP DE, IJZF, BL
26 LD A, I3 ;Izen, cik 72 JP Z, IIF15 ;I0N, error
27 SCF ;Ikapott! 73 LD A, B
28 RET 74 LD HL, E
29 R2 LD HL, (LP), I"iffo, ossz, cik 75 LD HL, B
30 R1 DEC HL, ICR-1, keres 76 OR C, I...
31 DEC BC 77 JR Z, W4
32 INC DL 78 DEC HL
33 LD A, B 79 LDUR ;Ihelucs, Ilozas
34 OR C, I... 80 W4 LD HL, (LP)
35 JR Z, R3 ;Ikerult! 81 LD HL, (BP)
36 DEC HL 82 IMC HL
37 LD A, (HL) 83 LD (BF)+HL
38 IMC HL 84 LD A, (CH)
39 CP 13 ;IK? 85 LD JL, W3
40 JR NZ, R21 ;Ives, nem 86 LF DEFS 2 ;Iiffo+pointer
41 R3 LD (LP)+HL, IJZ, pointer 87 LL DEFS 2 ;Iiffo+lengeth
42 LD (LL)+BC, IJZ, hossz 88 BP DEFS 2 ;Ibuffer+pointer
43 LD (BP)+HL, Ibuffer, cik 89 BL DEFS 2 ;Ibuffer+tengeth
44 R1 LD HL, (BP), Ibuffer, olv. 90 CH DEFS 1 ;Iwork
45 LD A, (HL), I"iffo, olv. char 91 UDB EQU 23670 ;Irelo, hater
46 IMC HL, I"iffo, olv. lev 92 RMAPT EDU 23730 ;Iiffo, kezdet
    
```


A Spectrum másoló, másolók spektruma

Miközben folyik a vita a szoftverlopásokról, a hazai Spectrum-tulajdonosok programjaik többségét ha akarnák sem tudnák másképpen megszerezni, csak cserébe, azaz másolás útján. A vásárolható szoftver nemcsak drága, de választéka is elképesztően szegényes. Míg a Spectrumra készült programok száma a nagyvilágban több ezerre rúg, a hazai forgalmazók (a BAV, a Novotrade stb.) alig tucatnyit tudnak ebből kínálni. Ilyen körülmények között a másolóprogramok alkalmazása szükségessé válik, nem pedig erkölcsi kérdés.

A hazai Spectrum-amatőrök körében többtucatnyi másolóprogram van forgalomban. Érdemes is belőlük minél többre szert tenni, mert alapszabály, hogy amit az egyikkel nem tudunk lemásolni, azt lemásolhatjuk egy másikkal; azaz nincs másolhatatlan program, csak esetleg még nem szereztük meg a megfelelő másolóprogramot.

Ismeretes, hogy a Spectrum egyik nagy előnye az állományok (ismertebb nevükön: fájlok) betöltésének és kimentésének rendkívüli egyszerűsége. A megfelelően „ragozott” LOAD, SAVE parancsokkal minden szabványos állomány (BASIC, gépi kód, képernyő, adattomb) könnyedén kezelhető. Ilyen állományokat kis ügyességgel minden nehézség nélkül közvetlenül másolhatunk, még másolóprogram sem kell hozzá. A rafináltan elkészített, különböző trükkökkel titkosított állományokat azonban már csak speciális másolóprogramokkal vehetjük át saját programkönyvtárunkba.

A másolóprogramok megítéséhez tudni kell, hogy egy szabványos Spectrum-állomány két részből: a fejből és a törzsből áll. Tulajdonképpen mind a kettő egy-egy fejnélküli állomány, és a ROM-program egy speciális feljőbajt — az OMNICOPIY terminológiájával markbajt — alapján különbözteti meg őket: a fejnél ez 0, a törzsnél 256. (Jó lenne, ha valaki közölné, hogy hol található ez a bajt!)

A fej 17 bajton tartalmazza a törzs adatait; leírásuk több magyar nyelvű publikációban megtalálható. Ezeket az adatokat minden valamirevaló másolóprogram kiírja.

Melyiket és miért?

A másolóprogramokat két osztályba sorolhatjuk: a *szimpla másolók* egyszerre csak egy fájl másolását teszik lehetővé, a *multi-másolók* pedig egyszerre több állományt is másolhatók. Mivel egy nagyobb program általában több állomány füzere, a szimpla másolókkal elég kényelmetlen dolgozni.

A továbbiakban a multimásolókról lesz szó.

Egy multimásoló megítésénél a következő szempontokat mérlegeljük:

- mekkora a szabad tárterülete,
- hány darab állomány tölthető be egyszerre,
- hogyan tölthetők a feleslegesen felvett állományok,
- milyen különleges (fejnélküli, hamisfejes, turbósított stb.) állományok másolhatók vele,
- milyen fejadatokat tud kiírni.

Egy jól megírt multimásoló szabad tárterülete kb. 40–42 k. Az ennél nagyobb állományok, a megafájlok speciális másolással másolhatók.

A legtöbb másolóprogram a képernyő felső vagy alsó harmadában helyezkedik el, és vagy látható, vagy elfedik az attribútumok. Ebbe a harmadba természetesen semmit sem lehet írni, vagyis a képernyőn egyszerre legfeljebb 10–15 állomány adatai jeleníthetők meg. A legtöbb másolónál ez behatárolja a betölthető állományok számát. Vannak azonban kivételek is: például az OMNICOPIY mozgó állománylistát készít és jeleníti meg, ezért a betölthető állományok számának csak a szabad tárkapacitás szab határt.

Egyes másolónál — például az FM3-nál — egyszerre tölthető valamennyi addig betöltött állomány. Ez néha nagyon kényelmetlen. A jobb másolók munkaállomány-jelzői (a fájlkurzorok) állíthatók — az FM3-nál csak SAVE módban —, és a törlés csak a munkaállománytól kezdődik. Ilyenek például a Zotycopy+ és a COPY—COPY. A legtokételesebb, ha csak a munkaállomány törlődik, mint például az OMNICOPIY-nál.

Egy másoló annál jobb, minél többféle különleges állományt tud kezelni. Nem ismerek olyan multimásolót, amely nem tud fejnélküli állományt másolni; az ilyen nem is érne sokat. Viszont például a turbósított állományok kezelése már extra szolgáltatásnak számít. A legtöbb fejtorést a hamisfejes állományok okozzák, melyek az állomány típusát, hosszát és betöltési címét illetően valótlán adatok tartalmazzák. A beolvasásra képes másolók sajnos csak a legdurvább szalaghíbat jelzik hibáüzenettel, és így ismeretlen állománynál sokszor nem tudhatjuk, hogy a fej hamis-e, vagy a beolvasás volt-e hibás. Általában eredményt hoz a megismételt betöltés: ha

ugyanis a valódi hossz másodszor nem ugyanaz, mint először, akkor biztos, hogy rossz volt a beolvasás. Az ellenkező eset sajnos semmit nem bizonyít, mert az azonos hossz egy makacs szalaghiba is eredményezhette.

Itt említem meg, hogy minden tisztesseges másoló kiírja a fejadatokat és a tényleges betöltési hosszt. Egyes másolók tizes számrendszert használnak, mások tizenhatost. Általában a tizes alak kényelmesebb, de éppen a gyanús állományok tesztelésénél hasznos lehet a tizenhatos alak is. Az elegánsan megírt gépi kódú állományok hossza ugyanis gyakran kerek szám, de a kerekesség érthető hexadecimálisan is. Például a képernyőállomány hossza 6912 bajt, ami hexadecimálisan 1B00. Ha tehát egy fejnélküli vagy hamisfejes kétes állomány hossza kerek szám (tízben vagy tizenhatosban), akkor már nem is olyan gyanús. De minden gyanú felett csak a hibátlanul futó programok állnak.

A jó másolóprogramtól extra szolgáltatásokat is várhatunk, mint például a szabad tárterület nagyságának kijelzése, a fej és a törzs külön állománykénti kezelése (ez akkor hasznos, ha az állományok egyenként tölthetők), a fejnélküli állomány megfejtése, az autoRUN megszüntetése, a másolat ellenőrzése (VERIFY), a fej módosítása, a fej titkosítása, megafájlmásolás, katalógus készítése, monitor üzemmód stb.

A másolóprogramokkal általában nem nehéz boldogulni, csak némi leleményességre és nagy türelemre van szükség. Legtöbbjük „önmagyarozó”, sőt sokat meg is „magyarítottak”. Mégis úgy gondolom, nem hiábavaló, ha megosztom tapasztalataimat az olvasókkal az általam előszeretettel használt négy multimásoló sajátosságairól. Három közülük — a Zotycopy, a COPIER FM3 és az OMNICOPIY—2 — Spectrum-körökben valószínűleg elterjedt, de a negyedik, a COPY—COPY talán kevésbé ismert.

A Zotycopy

Hibátlanul felvett, hibátlanul működő, nem turbósított és nem meghamisított fejnélküli állományok másolására a Zotycopy újabb változatait, a Zotycopy+—t és a Zotycopy#—t ajánlom. A kettő között a megafájlmásolásban különbség. Az előbbivel tetszőleges számú megafájlról lehet egy-

egy másolatot készíteni, az utóbbival egyetlen állomány másolható le több példányban. Egyébként a két változat azonos. A Zotyó legnagyobb előnye, hogy tiszta, szép másolatokat készít. A kék-piros csíkokat okozó betöltő részek hossza majdnem akkora, mint a közvetlenül SAVE-vel kimentett állományoknál. A fej és a törzs között érzékelhető szünetet tart, az egyes állományok között pedig ennél valamivel hosszabbat, ami lehetővé teszi össze nem tartozó állományok egy töltéssel való másolását. A SAVE (S), VERIFY (V) és FORGET (F — törlés) üzemmódokban a fájlkurzor a D (Down — le), illetve az U (Up — fel) gombokkal mozgatható. A SAVE és a VERIFY kapcsolható kézzel (Hand), amikor csak a kijelölt munkaállományokkal dolgozunk, vagy automatikusan (Auto), amikor a munkaállománytól az állománysor végéig történik a feldolgozás. A törlés a munkaállományra és az alatta felsorakozó állományokra vonatkozik.

AutoSAVE esetén a másolás az Enter lenyomása után néhány másodperccel kezdődik, HandSAVE esetén viszont azonnal.

A megafájlmásoló a Capsszift + M gombok megnyomásával indítható. Természetesen az alapüzemmódba való visszatérés nem lehetséges, ami egyébként minden megafájlmásolóra igaz. A Zotyó, ellentétben például a TICY (BTICY) szimpla megafájlmásolóval, csak fejnélküli megafájlmásolásra alkalmas. A fej elválasztását például az OMNICOPIY-val oldhatjuk meg, amely ugyan maga is képes megafájlmásolásra, de túlságosan rövid betöltő résszel.

Tapasztalatom szerint a bizonytalan felvételeket az OMNICOPIY általában jobban „megeszi”, mint a Zotyó, bár előfordult már az ellenkezője is. Olyan programot, ahol a fejben közlött állományhossz nem egyezik a ténylegessel, a Zotyó általában nem fogad el. Ezeknél az FM3 vagy az OMNICOPIY segíthet.

Ha a Zotyó betöltési hibát vagy memória-túlszordulást jelezve leáll, az L (LOAD) megnyomása újraindítja, miközben a hibás munkaállomány törlődik. Break vagy más üzemmód nem törli le a hibás állományt, de nem is kezeli.

A COPIER FM3

A COPIER FM3 nem ellenőrzi, a fejadatokat hexadecimálisan írja ki (léteznek decimális variánsok is), nem jelzi ki a szabad tárkapacitást, a betöltött állományokat csak egyszerre tudja kitorolni. Ezek a hátrányai. Előnye, hogy hibás és hamisfejes állományokat is elfogad; csillaggal jelöli őket a képernyő utolsó oszlopában. A hibás állományok nem mindig hibásan másolódnak ki!

Megfelelő türelemmel az FM3-at programmentés eszközeként is használhatjuk. A másolás indításakor hangjelzést ad, ami előnyös

lehet a visszakereséskor. Az AutoSAVE módban a fájlkurzortól kezd a kimentést. Az állományok betöltő részei valamivel rövidebbek, mint a Zotyónál, de azért elég hosszúak. A fej és a törzs közötti, illetve az automatikusan felvett állományok közötti szünetek hossza kielégítő. A fájlkurzort az S billentyű mozgatja.

Az OMNICOPIY—2

Az egyik legkiválóbb multimásoló az OMNICOPIY—2. Mindössze két hátránya van a Zotyóval és az FM3-mal szemben: egyrészt nincs lehetőség egyenkénti másolásra, másrészt túlságosan takarékosan bánt az idővel és a mágnesszalaggal: igen rövid a betöltő részek és fej a törzssel, állomány az állománnyal teljesen összefolyik. Annál több azonban az előnye! Külön állományként kezeli a fejet és a törzset, a munkaállomány egyenként az állománysoron belül is törölhető (Q), a fejnélküli állományok megfejelhetők (H), áttárolható dupla sebességgel turbó üzembe (D), megszüntethető az autoRUN (R), megafájlmásolókat is másol (M és N), titkosítja a fejet (Header Pitch on — P), megszmólja az állomány bájtaikat (C), a fejadatok között közli a markbájt tartalmát és a BASIC program változómezejének hosszát.

Ha a Break megnyomásával leállítjuk az OMNICOPIY-t és az L gombot nyomjuk meg, a LOAD üzemmód a tár teljes kitisztításával kezdődik. Ha folytatni akarjuk az állománysort, akkor a B gombot kell megnyomni, majd az Entert. Az OMNICOPIY így is LOAD módba kerül, de nem törli a munkatárát.

A COPY—COPY

A végére hagytam az Eniac Software által forgalmazott COPY—COPY multimásolót. A feliratok alapján valószínűleg lengyel termék. Rendkívül sokoldalú másoló, az OMNICOPIY-val együtt használva a kódtrés és a programtitkosítás hatásos eszköze. Különböző funkciói egy-egy opcióbillentyű, majd az Enter lenyomásával hozhatók működésbe.

(J) LOAD <n> Állomány beolvasása az n-edik helyre, n legfeljebb az utolsó sorszámánál egyvel nagyobb és 15-nél kisebb lehet. Ha n nem szerepel, akkor a betöltés a sorban következő helyre történik. Az n-edik helytől kezdve a már betöltött állományok törölődnek.

(S) SAVE <n> Állománykimentés az n-edik helytől végig. Egyenkénti (Hand) kimentésre nincs mód, n hiánya esetén

a kimentést az első állománytól kezd.

(V) VERIFY <n> Ellenőrzés. Az n szerepe ugyanaz, mint az S-nél.

(Z) COPY Megafájlmásoló

(L) LET n=név Az n-edik állomány átnevezése névre. A névsztriget nem szabad idézőjelbe tenni, az is része lesz az új névnek. Csak az első tíz karaktert használja fel. Ha nem adunk meg nevet, összezavarodhat a rendszer!

(C) CAT Szalagkatalógus

Monitorparancsok

(K) LIST <a> ($0 \leq a \leq 65535$). Négyoszlopos monitorlista az a címtől kezdve. A négy oszlop a következő: 1. cím, 2. a címen kezdődő kétbájtos szó, 3. a címzett bájt tartalma (dd), 4. CHR\$ dd. A nem opcionált billentyűk lapoznak, az opcionáltak kiadják saját parancsukat.

(O) POKE <a,dd> Az a címen elhelyezi a dd bájtot. Egylapnyi monitorlistát ad akkor is, ha dd-t nem adunk meg.

(U) USR <a> Az a címtől elindít egy gépi kódú programot. Nagyon óvatosan kell vele bánni, mert igen könnyen elszállhat a rendszerünk.

(Y) RETURN Visszatérés a BASIC-hez, a másoló törlése a tárból.

A rosszul beütött, nem kívánt opciók a DELETE segítségével törölhetők. A folyamatban levő funkciókat a Breakel lehet leállítani. Enter lenyomása esetén némely funkció (VERIFY, SAVE stb.) ismét jelentkezik.

A maximálisan 15 db betölthető program feladatai a következő szavak alatt jelennek meg:

nazve — sorszám (n) és név
 typ — típus (P, B stb.)
 dlug — a program hossza. Ezt az adatot a tényleges hossz felülírja
 start — autoRUN sorszám, illetve betöltési cím
 prog — ennek az információnak az értelmét nem sikerült megfejtenem.
 Az utolsóknak betöltött program neve alatt WOLNE inverz felirat látható, mellette a „dlug” oszlopban a még szabad tárkapacitás olvasható.

A betöltő részek elég hosszúak, de az egyszerre felvett állományok között érzékelhető szünet nincs, ezért csak összetartozó állományokat célszerű egy menetben másolni. A másolás befejezett bűgö hangjelzi, ami jól elválasztja egymástól a felvételeket.

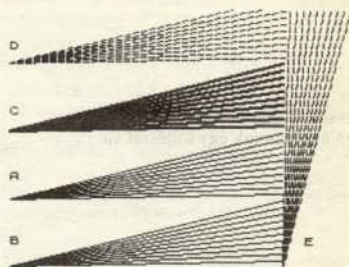
DR. NAGY ANDRÁS

Számítógépes grafika Pascalban

EGYENES

ZX-Spectrum

A ZX-Spectrumra készült Hisoft Pascal kitűnő eszköz olyan programok írására, amelyeknél a gyorsaság különösen fontos szempont. Ilyen területe a számítástechnikának a grafika is. Úgy hisszük, hogy az alábbi sorokat — és majd a következő számainkban a folytatást jelentő cikkeket is — tehát azok az olvasók is hasznosíthatják, akik valamilyen oknál fogva nem akarnak Pascalban programozni, viszont szeretnének megismerkedni a számítógépes grafikában használatos módszerekkel.



Néhány olyan számítógépes grafikai alapfeladat Pascalban irt megoldására mutatunk példát, amelyek megvalósítására a BASIC interpreterek lassúságuk miatt gyakorlatilag alkalmatlanok. A sebesség mellett a Pascal programok előnye, hogy a forrásnyelvű szöveg strukturaltsága következtében jól olvashatók és igen alkalmasak az algoritmusok tanulmányozására.

Sajnos a Hisoft Pascal nem ismer grafikai utasításokat, eljárásokat vagy függvényeket, szerencsére azonban a BASIC-hez hasonlóan lehetőséget ad gépi kódú eljárások hívására. Így a hiányzó és alapvető grafikai eszközöket magunk is megírhatjuk. Nos, rajzolás elvben is csak akkor lehetséges a képernyőre, ha legalább egy pont megjelenítésére szolgáló eljárásunk van.

Az 1. lista néhány grafikai eljárást tartalmaz. Ezek közül az első, a PLOT azonosítóju egy X, Y koordinátájú pont felrajzolására szolgál. A koordináta-rendszer kezdőpontját a BASIC PLOT utasításáéhoz hasonlóan a képernyő bal alsó sarkába helyeztük. A többlételet az adja, hogy a teljes képernyőre rajzolhatunk (az alsó két karakter sorba is), tehát a képernyő mérete 256×192 pont. Még egy előnye van az itt közölt PLOT eljárásnak a BASIC PLOT utasításával szemben: ha X vagy Y értéke nagyobb, mint a fenti határ, vagy negatív, ha tehát képernyőn kívülre esik, nem áll le a program, hanem egyszerűen nem hajtja végre az utasítást.

A gépi kódú rutin a COORDS rendszer-változóba (23672 = 507DH) tölti az X és Y koordinátát, majd az A regiszterbe 191-et — az Y irányban lehetséges pontok legnagyobb számát — tölti, és meghívja a képleme címe (PIXEL-ADDRESS) ROM rutint. Ezt követően hívja a PLOT szubrutint a 22ACH és a 22ECH címen. (Így végül is a ROM rajzoló eljárást használjuk a Pascalban is.) Mivel az egész kis gépi kódú programrész relocálható, egy INLINE-ba ágyazhattuk.

A LINE eljárás a ROM egyenesgenerátorát használja a BASIC koordináta-rendszer szerint. Csak azért mutatjuk be, hogy össze lehessen hasonlítani a Pascal egyenesgenerátorával. Akit az algoritmus működése érdekel, megtalálhatja az ismertetését a Hisoft Pascal leírásában.

A PixelLine azonosítóju eljárás az előbbieken bemutatott PLOT segítségével egyenes szakaszt rajzol. Az egyenes rajzolásánál az a probléma, hogy egyetlennek a képernyő raszterbeosztásának megfelelő pontjaiból olyan alakzatot hozunk létre, amelyet az emberi szem egyenessé észlel ki. Könnyen belátható, hogy raszteres grafikánál pontos egyenes csak kivételes esetekben rajzolható (például ha a párhuzamos valamely koordinátangellyel vagy a szögfelezőjükkel párhuzamos). Egyéb esetekben csak az biztosítható, hogy a felrajzolt pontsor az elméletileg pontos egyenes közvetlen környezetében helyezkedjen el.

A programunkban azt az eljárást valósítottuk meg, amelyet Bresenham fejlesztett ki. Az eljárás az X_1 , Y_1 és X_2 , Y_2 koordinátájú pontok közötti egyenes szakaszt jeleníti meg. A végrehajtandó algoritmus 45 fok hajlásszög alatti és feletti egyeneseknél eltérő. Ha az egyenes meredekségének abszolút értéke kicsi, további két eset lehetséges: az egyenes jobbra vagy balra mutat. Mindkét esetben az egyenest úgy rajzoljuk fel, hogy X irányban az egyenes pontjai a lehető legsűrűbben helyezkedjenek el, tehát raszterosztásonként léptetjük X értékét. Az X növekmény, melyet a program A-val jelöl, 1 vagy -1 attól függően, hogy a szakasz az első ponttól jobbra vagy bal irányba fut.

Az Y irányú növekményt a program annak megfelelően állítja be, hogy az egyenes az első ponttól fel vagy le mutat. A B növekmény ennek megfelelően 1 vagy -1. Ennyi előkészület után már hozzáférhetnénk az egyenes rajzolásához, csak még azt kellene eldönteni, hogy az egyenes X irá-

nyú lépdelés során mikor kell az Y irányú lépést végrehajtani. Ehhez a D változóban tárolt ún. döntési függvényt használjuk, amely azt tartja számon, hogy a pontos egyenes az éppen felrajzolandó pont alatt vagy felett haladna-e el.

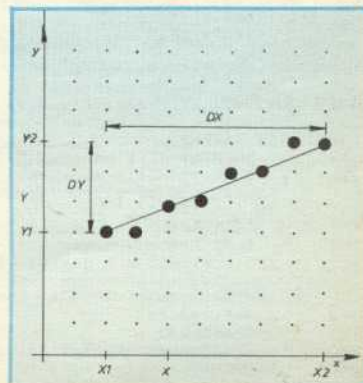
Az egyenes rajzolása mindig valóságos képpontból indul. Ez azért van így, mert X_1 és Y_1 integer paraméterek. Az egyenes elméleti egyenlete a vázlat szerint:

$$Y_e = \frac{DY}{DX} * (X - X_1) + Y_1$$

Ye általában nem egész szám, ezért el kell döntenünk, hogy az előző pont integer Y értékét az új pont esetén növeljük-e vagy sem. Úgy célszerű eljárunk, hogy ha az elméleti egyenes az eredeti Y értékhez közelebb esik, Y-t változatlanul hagyjuk, ha messzebb, Y-t eggyel növeljük. Ennek a kritériumnak az alábbi vizsgálat felel meg: Ha $Y_e - Y > 0,5$, akkor Y legyen $Y + 1$ vagy ha

$Y_e - Y > 0,5 > 0$, akkor Y legyen $Y + 1$.

A fenti vizsgálat könnyen elvégezhető az



ún. hibafüggvény (E) vizsgálatával, amely az előbbiektől szerint

$$E = \frac{DY}{DX} * (X - X1) + Y1 - Y - 0,5$$

összefüggésből számítható. A képlet hasz-

nálata azonban kényelmetlen, mivel lebegőpontos (valós) számokkal dolgozik, és az ezekkel végzett műveletek időigényesek. Számunkra a gyorsaság igen fontos, hiszen ezt a számítást minden pontra el kell végez-

ni. Egy kis ügyességgel elkerülhetjük a lebegőpontos műveleteket. Ha mindkét oldalt beszorzuk $2 * DX$ -szel, az így kapott $D = 2 * DY * (X - X1) - 2 * DX * (Y - Y1) - DX$

1. lista

```

10 <S:=?
20 <I:LINE F:lenevvel kimentett include rutinok>
30
40 PROCEDURE PLOT(X,Y:INTEGER);
50 <Pont rajzoló eljárás. A ROM rutinját
51 használja. Csak akkor rajzol, ha a pont
60 a 0<(X<=255 és 0<(Y<=191 meretu
61 képernyőre esik.)
70 BEGIN
80 IF (X)=0 AND (X<=255) AND (Y)=0 AND (Y<=191) THEN
90 INLINE(*FD,*21,*3A,*5C. CLD IY,*5C3A)
100 *DD,*46,*2. CLD B.(IX+2)
110 *DD,*4E,*4. CLD C.(IX+4)
120 *ED,*43,*7D,*5C. CLD (*SC7D).BC)
130 *3E,*BF. CLD A.191)
140 *CD,*AC,*22. <CALL *22AC)
150 *CD,*EC,*22 <CALL *22EC)
160 END: <PLOT)
170
180 PROCEDURE LINE(DX,DY:INTEGER);
190 <Az alábbi két eljárás a Spectrum ROM
191 egyenes rajzoló rutinját használja
200 HISDFT alapján>
210 VAR SGNX,SGNY:INTEGER;
220
230 PROCEDURE LINE1(X,Y, SX,SY:INTEGER);
240 BEGIN
250 INLINE(*FD,*21,*3A,*5C.
260 *DD,*56,*2,*DD,*5E,*4.
270 *DD,*46,*6,*DD,*4E,*B.
280 *CD,*BA,*24)
290 END: <LINE1)
300
310 BEGIN
320 WRITE(CHR(21),CHR(0));
330 IF DX<0 THEN SGNX:=-1 ELSE SGNX:=1;
340 IF DY<0 THEN SGNY:=-1 ELSE SGNY:=1;
350 LINE1(ABS(DX),ABS(DY),SGNX,SGNY)
360 END: <LINE)
370
380 PROCEDURE PixelLine(X1,Y1,X2,Y2:INTEGER);
390 <Ez az eljárás vízszont csak a pont
391 rajzoló algoritmust használja a ROM
400 programjából, az egyenes pontjainak
401 koordinátáit a Bresenham módszerrel
410 Pascalban számítja ki>
420 VAR X,Y,Z,A,B,DX,DY,D,DELTAQ,DELTAQ:INTEGER;
430 BEGIN
440 DX:=ABS(X2-X1);
450 DY:=ABS(Y2-Y1);
460 IF DY<=DX THEN <Az iránytangens <=1)
470 BEGIN
480 X:=X1: <X kezdőerteke)
490 Y:=Y1: <Y kezdőerteke)
500 Z:=X2: <X irányt figyel)
510 <Beállítja az X irányu növekményt)
520 IF X1<X2
530 THEN A:=1: <X növekszik)
540 ELSE A:=-1: <X csökken)
550 <Beállítja az Y irányu növekményt)
560 IF Y1<Y2
570 THEN B:=1: <Y növekszik)
580 ELSE B:=-1: <Y csökken)
590 <Inicializálja a döntési függvenyt
591 es deltat)
600 DELTAP:=DY+DY;
610 D:=DELTAP-DX;
620 DELTAQ:=D-D;
630 <Meghatározza a pont helyet es irányomatja)
640 PLOT(X,Y): <Első pont)

```

```

650 WHILE X<>Z DO
660 BEGIN
670 X:=X+A;
680 IF D<0
690 THEN D:=D+DELTAQ
700 ELSE
710 BEGIN
720 Y:=Y+B;
730 D:=D+DELTAQ
740 END;
750 PLOT(X,Y);
760 END <WHILE);
770 END <Ha DY=DX)
780 ELSE <Ha DY>DX)
790 BEGIN
800 Y:=Y1: <Y kezdőerteke)
810 X:=X1: <X kezdőerteke)
820 Z:=Y2: <Y irányt figyel)
830 <Y növekmény beállítása)
840 IF Y1<Y2
850 THEN A:=1: <Y növekszik)
860 ELSE A:=-1: <Y csökken)
870 <Beállítja X növekményt)
880 IF X1<X2
890 THEN B:=1: <X növekszik)
900 ELSE B:=-1: <X csökken)
910 <Inicializálja a döntési függvenyt
911 es deltat)
920 DELTAP:=DX+DX;
930 D:=DELTAP-DY;
940 DELTAQ:=D-D;
950 <Meghatározza a pont helyet es
951 irányomatja)
960 PLOT(X,Y): <Első pont)
970 WHILE Y<>Z DO
980 BEGIN
990 Y:=Y+A;
1000 IF D<0
1010 THEN D:=D+DELTAQ
1020 ELSE
1030 BEGIN
1040 X:=X+B;
1050 D:=D+DELTAQ
1060 END <ELSE)
1070 PLOT(X,Y);
1080 END <WHILE);
1090 END <ELSE)
1100 END: <PixelLine)
1110
1120 PROCEDURE FatLine(X0,Y0,X,Y,H:INTEGER);
1130 <Vastag egyenes rajzolása)
1140 VAR I,J:INTEGER;
1150 BEGIN
1160 FOR I:=1 TO H DO
1170 FOR J:=1 TO H DO
1180 PixelLine(X0+I-1,Y0+J-1,X+I-1,Y+J-1)
1190 END: <FatLine)
1200
1210 PROCEDURE DotLine(X0,Y0,X,Y:INTEGER);
1220 <Szaggatott vonal rajzolása)
1230 CONST H:=4:D:=6;
1240 VAR I,N,XX0,XX:INTEGER;
1250 T,YY0,YY:REAL;
1260 BEGIN
1270 IF ABS(X-X0)>0 THEN
1280 T:=(Y-Y0)/(X-X0)
1290 ELSE T:=2000;
1300 IF T<1 THEN
1310 BEGIN
1320 N:=ROUND((X-X0)/D);
1330 XX0:=X0;
1340 XX:=X0+H;
1350 FOR I:=0 TO N DO

```

```

1360 BEGIN
1370 YY0:=T*XX0+Y0;
1380 YY:=T*XX+Y0;
1390 PixelLine(XX0,ROUND(YY0),XX,
1391 ROUND(YY),XX);
1400 XX0:=XX0+D;
1410 XX:=XX+D;
1420 END
1430 END ELSE
1440 BEGIN
1450 T:=1/T;
1460 N:=ROUND((Y-Y0)/D);
1470 XX0:=Y0;
1480 XX:=Y0+H;
1490 FOR I:=0 TO N DO
1500 BEGIN
1510 YY0:=T*XX0+X0;
1520 YY:=T*XX+X0;
1530 PixelLine(ROUND(YY0),XX0,
1531 ROUND(YY),XX);
1540 XX0:=XX0+D;
1550 XX:=XX+D;
1560 END
1570 END
1580 END: (DotLine)
L

```

2. lista

ún. döntési függvény vizsgálatát kell csak végrehajtunk, mely érték egész szám. A PixelLine eljárás 600-760-as soraiban ezt az algoritmust programoztuk azzal a módosítással, hogy a kijelölt szorzásokat is sorozatos összeadással helyettesítettük.

Az eljárás második része a fenti algoritmust ismétli meg arra az esetre, ha az egyenes iránytangense a 45 fokos egyeneséhez képest nagyobb. Ennek működése az előzőek szerint kézenfekvő. Most az Y-t léptetjük egyenletesen, és a döntési függvénnyel azt vizsgáljuk, mikor kell az X-en változtatni.

Az eddigiek alapján egy sor egyenesekkel kapcsolatos feladatot oldhatunk meg. Az 1. lista FatLine eljárásával tetszőleges vastagságú egyeneseket jeleníthetünk meg. Az eljárás paraméterei megegyeznek a PixelLine-ével, de kiegészülnek még egy H integrérték szerint hívott paraméterrel, amely a vonal vastagságát adja meg képpontvonalvastagság egységben.

A DotLine eljárás viszont szaggatott vonalat rajzol. Az eljárásba nem paraméterként, hanem állandóként definiáltuk, hogy az egyenes négy képpontnyi egyenes szaka-

szokból (H) és ezeket elválasztó két képpontnyi szünetekből áll, amely mintázat tehát hat képpontként ismétlődik.

Az ismertetett eljárásokhoz készített demonstrációs program a 2. listán látható. Ezt többféleképpen is kipróbálhatjuk. Célszerű először a Pascal szövegszerkesztőjével az 1. listát elkészíteni, majd elmenteni magnókazettára vagy Microdrive cartridge-ra. (A listák a lemez meghajtós változatot mutatják be.) Itt a fájl P1,9999,1:LINE paranccsal mentjük ki. Ha ezután elkészítjük a 2. lista programját és azt C paranccsal lefordítjuk, a fordítás a 70-es sornál INCLUDE paranccsot talál, amelynek hatására a képernyőre kiírja a fájlnevet, beolvassa a meghajtóból a LINE nevű szövegfájlt és befordítja a főprogramba. Ilyenkor azt tapasztalhatjuk, hogy fordítás közben a meghajtó többször beindul, leáll. Ez nem hiba, mivel mindig csak egy puffernyit olvas be a gép, ezt lefordítja, majd ha kifogy a szövegből, újabb adagot olvas be és így tovább.

A másik lehetőség a következő. Ha a demonstrációs program bebillentyűzése során a 60-as sorig jutunk, G,1:LINE paranccsal hívjuk be a meghajtóról az egyenesrajzoló

eljárásaink forrásszövegét. Ez a művelet a BASIC-kel ellentétben nem írja felül a memóriában levő programot, hanem hozzáfűzi. Egyidejűleg a sorokat egyesével át is számozza, de ez az N paranccsal megváltoztatható. A beolvasás után a főprogram bebillentyűzését a 2. lista 90-es sornál folytatjuk. Természetesen ez a sor az összeszerkesztett listában nem a 90. lesz.

Ennek a második módszernek az az előnye, hogy a fordításhoz a lemez meghajtót nem kell használnunk, ami a munka lényeges gyorsítását jelenti. Hátránya viszont, hogy ezzel a módszerrel csak kisebb programok fordíthatók. Azt tehát, hogy melyik módszer a célszerűbb, a feladat jellege dönti el.

Kazettás magnetofonnal a fenti eljárás kissé módosul. A fájlnev ebben az esetben LINE lesz. Két fájl kell készítenünk az eljárásokról. Az egyiket P1,9999,LINE paranccsal hajtjuk végre. Ezt lehet majd a fent ismertetett második módszer szerint beszerkeszteni a főprogram szövegébe. Az INCLUDE módszerhez a fájl a W1,9999,LINE paranccsal kell a kazettára menteni. Ekkor a Pascal a fájl 128 bajtos rekordok formájában viszi ki szalagra, és közöttük elegendő szüneteket hagy ahhoz, hogy visszaolvasás közben a fordítást a fordítóprogram elvégezhesse.

Az eljárások külön fájlokban való tárolása azért célszerű, mert ezekből egy új program gyorsan összeszerkeszthető. Az ismertetett eljárásokat úgy irtuk meg, hogy globális változókat nem tartalmaznak, így az új programnál csak a paraméterek kitöltésére kell ügyelnünk. Gyakorlott Pascal programozó az idő folyamán egész kis eljárás- és funkciókönyvtárat gyűjthet össze, amelynek segítségével a Pascaltól izlése nek megfelelő, mondhatni testre szabott nyelvet alakíthat ki.

Végül a bemutató programról annyit hogy a grafikai példa A egyenesseregét a LINE eljárással, tehát az eredeti ROM rutinán, a B-vel jelölt egyeneseket pedig a PixelLine eljárás segítségével rajzoljuk fel. Mivel a két egyenessereg egyes elemei egymással párhuzamosak, jól látható, hogy ugyanolyan algoritmusra épül mindkét eljárás.

DR. KABOLDY PÉTER

AZ ATARI-PÁLYÁZAT NYERTESEI

A KISZ KB Középiskolai és Szakmunkástanulói Tanácsa (KSZT) a Skála Coop számítástechnikai leányvállalatával, a Computer-S-sel közösen „ATARI-PÁLYÁZAT”-ot hirdetett (lásd magazinunk 1987/1. számában).

A KSZT a beérkezett pályázatokat katalogizálta és titkosította, majd bírálatra átadta a Skálának.

A nyertesek és díjazói:

1. Reith József (Budapest, Városmajor u. 66.)
ATARI 800 XL számítógép teljes kiépítéssel, egyéves használatra
2. ifj. Kékkői Zoltán (Villány, Tánicsics Mihály u. 13.)
ATARI 800 XL számítógép teljes kiépítéssel, fél éves használatra
3. Kovács Tibor (Szeghalom, Kinizsi u. 37.)
ATARI-program kazettán
4. ifj. Brosig János (Szombathely, Bolyai u. 13/a.)
ATARI-program kazettán

5. Antal József (Sopron)
C64-program kazettán

6. Vidács Jenő (Szarvas, Vajda P. u. 20.)
C64-program kazettán

A nyertesek ezúton is meghívást kapnak a KISZ KB KSZT 1987/88. tanévi szaktáborába.

A Középiskolai és Szakmunkástanulói Tanács Titkársága

Hangos billentyűzet

COMMODORE—64

Közismert, hogy a Commodore 64-en a billentyűk lenyomásakor nem kapunk hangvisszajelzést, és így könnyen elkerülheti figyelmünket, ha a lenyomott billentyűt a számítógép nem érzékeli. Ezen segít az alábbi gépi kódú program: a gong hangjára emlékeztető jelzés csak akkor következik be, ha a számítógép a lenyomott gomb hatására elfogadta az adatot.

Mint tudjuk, a C64 minden hatvanad másodpercben végrehajt egy belső megszakítást, melynek több célja van: ilyenkor kérdezi le a számítógép, hogy le van-e nyomva valamelyik billentyű, a program ellenőrzi a STOP billentyű lenyomását, villogtatja a kurzort, felfrissíti a belső órát stb. Ehhez a belső megszakításhoz csatlakoztathatunk, ha a \$314–315 címen (788–789) található belső megszakítási

vektor címét átírjuk úgy, hogy először a mi programunkra mutasson, majd a program végén visszaugrunk az eredeti belső megszakítási rutinra, amelynek belépési pontja \$EA31.

Az 1. listán a gépi kódú program látható. Maga a program a \$C000 (=49152) fölötti memóriaterületen van, így a BASIC programok működését nem zavarja, és a NEW-utasítás is hatástalan rá. BASIC-ből a SYS(49152)-vel lehet elindítani és mindaddig működik, amíg a gépet ki nem kapcsoljuk, vagy a RUN/STOP és RESTORE billentyűket egyszerre le nem nyomjuk, vagy újra ki nem adunk egy SYS(49152) utasítást.

A listán bekeretezett számok megváltoztatásával egyszerűen az 1. és 3. hanggenerátor frekvenciáját módosíthatjuk, másrészt a hang felfutásának és lecsengésének idejét.

Ha például a középső bekeretezett szám — a \$79 — helyett \$7E-t írunk, akkor a lecsengés ideje 750 ms-ról 15 s-ra változik. POKE 49197, HANG1-gyel az első hanggenerátor frekvenciáját, POKE 49207, HANG3-mal a harmadik hanggenerátor frekvenciáját, POKE 49202, ATDEC utasítással pedig a hang felfutásának és lecsengésének idejét lehet módosítani. Így mindenki beállíthatja a számára legkellemebb hangot.

A program természetesen BASIC-ből is a memóriába vihető (2. lista). A listán bekeretezett számokkal a hanggenerátorok paramétereit változtathatók meg: 28 az első hanggenerátor frekvenciáját, 121 a hang felfutását és lecsengését, 39 a harmadik hanggenerátor frekvenciáját szabja meg.

SZABÓ PÉTER PÁL

```

10PEN4,4
3 SYS(36864)
5 ;
7 ; GONG - HANGOS KLAVIATURA
9 ;
11 .OPT P4,00
13 ;
15 IDD = #05
17 KEY = #C5
19 IRQOLD = #EA31
21 IRQVEK = #314
23 SID = #D400
25 ;
27 * = #C000
29 ;
31 BE SEI
33 LDA IRQVEK
35 EOR #< SCAN ^ IRQOLD
37 STA IRQVEK
39 LDA IRQVEK+1
41 EOR #> SCAN ^ IRQOLD
43 STA IRQVEK+1
45 CLI
47 RTS
49 ;
51 SCAN LDA KEY
53 CMP #*40 ; VAN-E BILLENYU LENYOMVA
55 BEQ NOKEY ; NINCS!
57 JSR SOUND ; HANGKELTES
59 LDA #IDD ; AZ IDOZITES HOSSZA
61 STA CNTR
63 NOKEY DEC CNTR
65 BNE OUT ; VISSZA A REGI IRQ-HOZ.
67 JSR NOSOUND ; HANG KI
69 OUT JMP IRQOLD
71 ;
73 SOUND LDA #*1C ; FREKVENCIA
75 STA SID+1 ; HANG-1

```

```

77 LDA #*79 ; A HANG FELFUTASA 80MS,
79 STA SID+5 ; ES LECSENGSE 750MS.
81 ;
83 LDA #*27 ; FREKVENCIA
85 STA SID+15 ; HANG-3
87 ;
89 LDA #*0F ; A HANGERO 15.
91 STA SID+24
93 ;
95 LDA #*15 ; HAROMSZOG HULLAMFORMA
97 STA SID+4 ; ES KORMODULACIO HANG 1-3
99 RTS ; KOZOTT.
101 ;
103 NOSOUND LDY #*1C
105 LDA #0
107 LP1 STA #D400,Y
109 DEY
111 BPL LP1
113 RTS
115 ;
117 CNTR .BYTE #05
119 ;
121 .END

```

1. lista
2. lista

```

1000 FOR I= 49152 TO 49233
1010 READ X:POKE I,X:S=S+X:NEXT
1020 DATA 120,173, 20, 3, 73, 34,141, 20
1030 DATA 3,173, 21, 3, 73, 42,141, 21
1040 DATA 3, 88, 96,165,197,201, 64,240
1050 DATA 8, 32, 44,192,169, 5,141, 81
1060 DATA 192,206, 81,192,208, 3, 32, 70
1070 DATA 192, 76, 49,234,169, 28,141, 1
1080 DATA 212,169, 121, 141, 5,212,169, 39
1090 DATA 141, 15,212,169, 15,141, 24,212
1100 DATA 169, 21,141, 4,212, 96,160, 28
1110 DATA 169, 0,153, 0,212,136, 16,250
1120 DATA 96, 5
1130 IF S<> 8526 THEN PRINT"ADATHIBA !":END
1140 PRINT"OK !!"

```

Lebegőpontos tömbök átadása BASIC → Pascal

Programjaink fejlődésében gyakran eljön az a pillanat, amikor az addig kényelmes BASIC túl lassú lesz, vagy más fogyatékoság miatt át akarunk térni Pascal nyelvre. Az alábbi program ahhoz nyújt segítséget, hogy ilyenkor legalább a numerikus tömböket ne kelljen újra adatokkal feltölteni, hanem egyszerűen át lehessen vinni az új programba.

Ehhez először is a két nyelv számbázis-átadási módját kell ismernünk.

A BASIC-ben a számokat 5 bájtton tároljuk. Jelöljük ezeket abban a sorrendben, ahogy a memóriában vannak: B1, B2, B3, B4, B5-tel! Ebből a lebegőpontos szám kitevője a B1-en van tárolva eltolt formátumban, azaz 128 hozzá van adva:

$expB = B1 - 128$ (a BASIC kitevő)

A mantissa a B2 ... B5 bájtokon van, ahol a BASIC mantisszára igaz, hogy $1/2 \leq mB < 1$

és a „kettédespont” a B1 7. bit előtt áll, ami mindig "1" követne, de rá van téve az előjel. Ezekkel a tárolt X szám:

$X = előjel * mB * 2^{expB}$

(speciálisan X=0 esetén minden bájt 0.)

A Hisoft Pascalban ezzel szemben, ha a szám a P1, P2, P3, P4 bájtokon van ábrázolva, ahol P2 mutatja az exponenst, de a változatosság kedvéért kettes komplementum formában, azaz,

$expP = P2$, ha $P2 < 128$

$expP = P2 - 256$, ha $P2 > 128$,

a mantissa pedig sorban a P4, P3, P1 bájtokon van, és a Pascal mantissa tartománya:

$1 < mP < 2$.

Igy a P4 6. bitje mindig 1, míg az előjel a BASIC-kei megegyezően a 7. biten áll. Az X szám tehát:

$X = előjel * mP * 2^{expP}$.

Mivel a mantisszáknak itt 1 és 2 között kell lennie, így az első átirási szabály:

$mP = 2 * mB$, és emiatt

$expP = expB - 1$

Ha ez nem lenne, akkor a két exponens bájtton egyszerűen 128 hozzáadása kapcsoltná össze (gondoljuk meg, hogy pozitív exp esetén a BASIC hozzáadott 128-at, ha megint adunk hozzá 128-at, akkor ugyanannyi lesz, mint eredetileg, és épp ez kell a Pascalban. Negatív exp esetén pedig a kétszer hozzáadott 128 éppen a 256-tal növelést adja), viszont mivel a két kitevő nem egyenlő, így a bájtokon végzendő művelet vég-eredményben:

$P2 = B1 + 127$

A mantisszát pedig jobbra kell léptetni 1 bittel, úgy, hogy B2-ben a 7. bit (előjel) maradjon, a 6. bitet pedig 1-be kell állítani. (X=0 esetén pedig itt is minden bájt 0.)

```

70 ;z!!!
80 *L-
90
100 ;lebegőpontos tömbök átírása PASCAL formára. 87.04.16.
110 ;magnó/drájv
120 ;hívási formája:
130 ;...USR ORG:REM változónév
140 ;Először az adott változó címét megkeresi.
150
160 *D+
170 *L+
180          ORG 49801
190 ;hossza 147 byte
200 *L-
210
220          RST 32          ;a CHADD léptetése kétszer:
230          RST 32          ;":REM" átugrásával rááll a változónévre
240          CALL #28B2      ;a tömb címét megkeresi
250          PUSH HL         ;és elteszi.
260          JR NC,OK        ;ha megtalálta, OK,
270          RST 8           ;egyébként Variable not found
280          DEFB 1
290
300 OK       RST 32          ;a CHADD-ot a sor végére tolja
310          CP 13
320          JR NZ,OK
330
340
350
360
370 START   POP HL          ;A tömb címe.
380          INC HL          ;a név
390          INC HL          ;és a teljes hossz átugrása
400          INC HL
410          LD B,(HL)       ;a dimenziók száma
420          INC HL          ;első dimenzióméret címe
430          PUSH HL
440          LD HL,4         ;(4 byte/szám lesz)
450 ;a méreteket és 4-et összeszorzó ciklus
460 IDE     LD (HOSSZ),HL   ;eddiggi részletszorzat
470          POP HL          ;i-edik dimenzió címe
480          LD E,(HL)       ;mérete -> DE
490          INC HL
500          LD D,(HL)
510          INC HL
520          PUSH HL         ;az i+1 -edik dimenzió címe
530          LD HL,(HOSSZ)  ;eddiggi részletszorzat
540          CALL #30A9      ;HL=HL*DE
550          DJNZ IDE        ;az összes dimenzió összeszorzása.
560
570          LD (HOSSZ),HL
580          PUSH HL         ;Pascal tömb hossza
590          LD DE,50000     ;és kezdőcíme
600          ADD HL,DE
610          JR NC,CONV0     ;ha elfér, mehet,
620          RST 8           ;egyébként Out of memory
630          DEFB 3
    
```

```

640
650 ;a tényleges Átalakító részlet
660
670 CONV0 POP BC ;Pascal tömb hossza
680 POP HL ;Basic tömb címe
690 PUSH BC ;visszatéréshez: Pascal hossz
700
710 CONV PUSH BC
720 CALL #3297 ;egész számot valósra alakít
730 POP BC
740
750 B1 LD A,(HL) ;a kitevő
760 CP 0 ;ha a kitevő 0,
770 JR Z,ZERO ;akkor minden byte 0!
780 INC DE ;P2 címe
790 ADD A,127 ;B1 módosítása
800 LD (HL),A
810 LDI ;áttöltés P2-be
820 INC DE
830
840 B2 SRA (HL) ;jobbra lép, de a 7. bit marad (el
;sjel)
850 SET 6,(HL) ;a 6. bit mindig 1
860 LDI ;áttöltés P4-be
870 PUSH DE ;a köv. szám P1 címe lesz.
880 DEC DE
890 DEC DE ;P3 címe
900
910 B3 RR (HL) ;jobbralép (Átvitelt viszi)
920 LDI ;áttöltés P3-ba
930 DEC DE
940 DEC DE
950 DEC DE ;P1 címe
960
970 B4 RR (HL) ;jobbralép
980 LDI ;áttöltés P1-be
990 INC HL ;B5-öt kihagyjuk
1000 POP DE ;a következő számban P1 címe
1010
1020 NEXT JP PE,CONV ;amíg nincs kész (BC>0), tovább!
1030 POP BC ;kimenő adat: Pascal tömb hossza
1040 RET ;=====
1050
1060 ZERO LDI ;4 db 0 áttöltése
1070 LDI
1080 LDI
1090 LDI
1100 INC HL ;B5-öt kihagyjuk
1110 JR NEXT ;vége
1120
1130 HOSSZ DEFS 2 ;PASCAL HOSSZ
1140
1150 ;+++++
1160
1170 ;drávj részére a PASCAL tömböt egy stringtömbbe tesszük
1180 ;hívási formája:
1190 ;...USR ORG1:REM változónév
1200 ;Először az adott változó címét megkeresi.
1210
1220 ORG1 RST 32 ;a CHADD léptetése kétszer:
1230 RST 32 ;":REM" átugrásával rááll a változ
;ónévre
1240 CALL #2BB2 ;a tömb címét megkeresi
1250 PUSH HL ;és elteszi.
1260 JR NC,OK1 ;ha megtalálta, OK,
1270 RST 8 ;egyébként Variable not found
1280 DEFB 1
1290
1300 OK1 RST 32 ;a CHADD-ot a sor végére tojja
1310 CP 13
1320 JR NZ,OK1
1330
1340

```

Ezek alapján egyszerűen megírhatjuk a konvertáló programot. Először a gépi kódú részt nézzük (1. lista).

A gépi kódú programot a 49801 címre tesszük, a keletkező Pascal tömb az 50000-tól fog indulni (kb. 15 k nagyságú lehet maximálisan). Először egy ROM rutinál megkerestjük az adott változó címét, amit a HL-ben fogunk megkapni. Ez a részlet bármilyen változó keresésére más gépi programokba beépíthető.

A START című részben megnézzük, hány dimenziós a tömb, mekkora a kiterjedése az egyes dimenziókban. Így kiszámítjuk a Pascal tömb hosszát, ezt a HOSSZ címen is tároljuk, és a visszatérésnél a BC-be is betesszük, tehát ez lesz USR 49801 értéke.

A 710-es sornál kezdődik a tényleges átszámítás, az előzőleg megadott képlet szerinti. Ezt a programrészt annyiszor hajtja végre, ahány eleme van a tömbnek. A HL regiszterpár mindig a kezelése alatt álló BASIC bájt címe, a DE a kapott Pascal bájt-ra mutat, míg BC visszafelé számolja az átalakított bajtokat; amikor 0 lesz, akkor van vége a munkának.

A 720-as sorban lévő utasítás fontos, ez ugyanis a BASIC tömbben lévő esetleges egész számot lebegőpontosá alakítja.

A B1...B4 címek csak segítségképpen vannak bejelölve: mindig azt mutatják, hogy éppen a BASIC szám melyik bajtját alakítjuk át.

Az X=0 esetet a rutin a B1=0 alapján ismeri fel, ilyenkor a ZERO című részt hajtja végre.

A lista második részében (1220-as sor) a lemez meghajtások kedvéért van egy kis rutinocská, amely a Pascal tömböt egy BASIC karaktertömbbe másolja. A tömb egydimenziós, és azért tömb, nem pedig egyszerű sztring, hogy a BASIC-ben be lehessen előre állítani a pontos hosszát, ami, mint látni fogjuk, fontos.

Miután a sztringtömb címét megkerestük, az 1430-as sorban már a szöveg első karakterére mutatunk. Ide LDIR-rel áttöltjük a Pascal bajtokat (a HOSSZ-ból tudjuk, hogy mennyit).

A BASIC főprogram (2. lista) úgy használja ezeket a rutinokat — amelyek itt lebegő-m2 néven vannak lefordítva —, hogy lemez meghajtóval és magnóval is mehessen, ezért kétféle indításvan (lásd ehhez a 380-tól kezdődő sorokat is).

A BASIC tömböt olvastatjuk be ezután; a szokás szerint a lemez meghajtó számát kérdezzük, és ha ez 0, akkor magnóról olvasunk.

A 160-as sor konvertál, és megtudjuk a Pascal tömb hosszát is. Ezután felvesszük a Pascal tömböt, az igény szerinti nével. A névnek itt 8 karakteresnek kell lennie. Ha hosszabb, akkor visszadobjuk, ha rövidebb, akkor "." jelekkel megtoldjuk. A Pascal szokásának megfelelően, a név 2. karakterében lévő "." jelzi a meghajtót (lásd 220-as sor), és ekkor az első karakter a lemez meghajtó száma. Jöhet tehát a felvétel akár magnóra, akár lemez meghajtóra.

Ehhez a következőt kell tudni. A magnó esetén a Pascal megeszi a BASIC-ben simán Code-dal felvett blokkot, így nincs sok problémánk (330—350-es sor). A le-


```

1350 POP HL ;A tömb címe.
1360 INC HL ;a név,
1370 INC HL ;a teljes hossz,
1380 INC HL
1390 INC HL ;a dimenziók száma és az
1400 INC HL ;első dimenzióméret átugrása után
1410 INC HL
1420 ;itt kezdődik a szöveg
1430 EX DE,HL ;a string lesz a cél
1440 LD HL,50000 ;ez a forrás
1450 LD BC,(HOSSZ) ;Pascal tömb hossza
1460 LDIR ;áttöltés
1470 RET
1480
1490 *L+
1500 END

```

1. lista

2. lista

```

10 REM lebegőpontos tömb konvertálás Basic->Pascal:
20 REM ----- start -----
30 CLEAR 49800
40 LOAD "lebego-m2"CODE
50 GO TO 90
60 CLEAR 49800
70 LOAD "*"m";1;"lebego-m2"CODE :
80 REM ----- Basic DATA -----
90 INPUT "Basic file name " ;b$
100 INPUT "drive num. " ;d: IF d=0 THEN GO TO 140
110 IF b$="" THEN GO TO 90
120 LOAD "*"m";d;b$ DATA x()
130 GO TO 160
140 LOAD b$ DATA x():
150 REM ----- konvert -----
160 LET hossz=USR 49801: REM x():
170 REM ----- SAVE -----
180 INPUT "Pascal file name (d:..=drive) " ;p$
190 IF LEN p$>8 THEN GO TO 180
200 IF LEN p$<8 THEN LET p$=p$+"_": GO TO 200
210 REM drive OR tape?
220 IF p$(2)<>":" THEN GO TO 330
230 LET d=CODE p$-48
240 IF d<1 OR d>8 THEN GO TO 180
250 REM ----- drive -----
260 OPEN #4;"m";d;p$(3 TO )
270 DIM x(1): DIM d$(hossz)
280 RANDOMIZE USR 49915: REM d$
290 PRINT #4;d$:
300 CLOSE #4
310 GO TO 90
320 REM ----- tape -----
330 SAVE p$CODE 50000,hossz
340 BEEP 1,40: PRINT " RETURN TO VERIFY "
350 VERIFY p$CODE 50000,hossz
360 GO TO 90
370
380 REM ----- szaporítás -----
390 REM ----- tape -----
400 SAVE "lebego.bas" LINE 30
410 SAVE "lebego-m2"CODE 49801,147
420 BEEP 1,40: PRINT " RETURN TO VERIFY "
430 VERIFY "lebego.bas"
440 VERIFY "lebego-m2"CODE
445 STOP
450 REM ----- drive -----
460 SAVE "*"m";1;"lebego.bas" LINE 60
470 VERIFY "*"m";1;"lebego.bas"
480 SAVE "*"m";1;"lebego-m2"CODE 49801,147
490 VERIFY "*"m";1;"lebego-m2"CODE

```

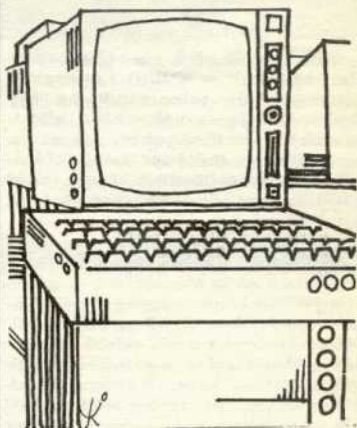
mezemghajtón azonban a Pascal ún. szekvenciális fájlban tárolja az adatokat, tehát nekünk is ilyet kell produkálnunk. Ezért a 260-as sorban először nyitunk egy csatornát a kívánt nevű fájl számára. Az X tömböt lecsökkentjük, hogy legyen helyünk, majd a d sztringet pontosan a megfelelő hosszúra dimenzionáljuk. A 2. gépi kódú rutin ide teszi az ártírt adattömböt, amit az-

tán a 290-es sorban a fájlba teszünk (a "": nem hagyható el!).

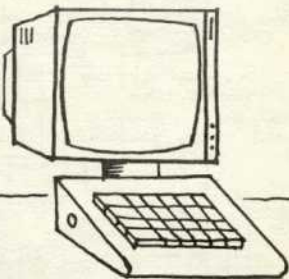
Ne ijedjen meg senki, ha a gépi program működését nem érzékeli! Az lefutott, csak olyan gyorsan, hogy nem lehet észrevenni; máris a Pascal nevet kéri a program.

Mindkét variációban a felvétel után visszaugrunk a program az elejére, újabb tömb beolvasására. SLÍZ MIKLÓS

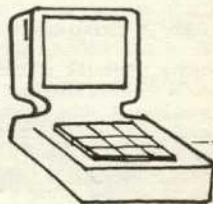
Hierarchia



főgebra



gebra



algebra

Miért nem stupid a MUPID?

avagy:

Milyen az intelligens videotex-dekóder? II.

Sorozatunk első részében röviden vázoltuk a MUPID koncepció lényegét, és felsoroltuk az így nyújtható, igen változatos alkalmazási lehetőségeket. Most a speciális vtx-dekóder belső felépítésével ismerkedünk meg, majd főbb üzemmódjainak ismertetését kezdjük el.

Felépítés és csatlakozó felületek

A MUPID központi egység egy 4 MHz-en ketyegő Z80A. 32 kB-os PROM-mal, 128 kB-os RAM-mal (vtx-oldalak és programok tárolására) és a vezérlőkarakterek számára külön 8 kb-ajtos RAM-mal rendelkezik (1. ábra). Készüléken belüli további 128 kB-os bővítésre is van lehetőség. Egy másik mikroprocesszor figyelni és alakítja a billentyűzetről érkező jeleket soros adatjelékké a dekóder számára, és ugyancsak erre van bízva a billentyűzet kódolása, a pergesmentés és tartós kiváltás, valamint a hangkeltés. A MUPID A1 és A2 jelű ana-

használó számára is lehetővé tették. A MUPID-ot a hozzá csatlakoztatható kiegészítő berendezések teszik még sokoldalúbbá. A tervezők olyan interfészeket alakítottak ki a dekóderen, melyek sokféle külső eszközhöz használatára nyújtanak lehetőséget. Lássuk tehát a csatlakozó felületek (2. ábra) főbb jellemzőit.

A TV/RGB monitor csatlakozás a széles körben terjedő EUROSCART/CENELEC csatlakozó, mely az R, G, B alapszín-összetevőn kívül a kioltójelet, összetett szinkronjelet és a hangjelet szolgáltatja.

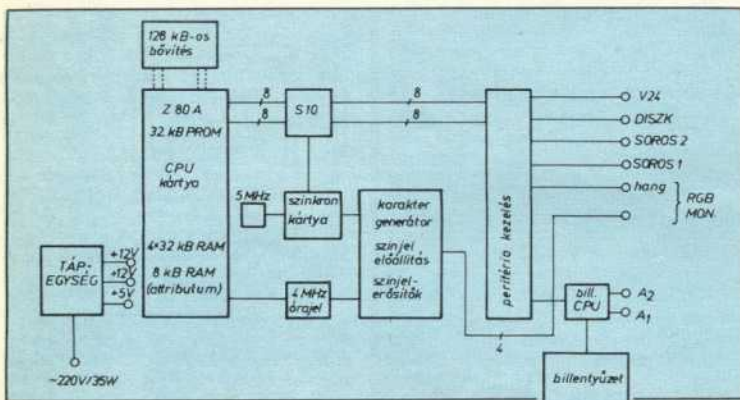
Az A1, A2 analóg bemenetek 6 pólusú DIN aljzatok, elsősorban botkormányok

csatlakoztatására vezették ki őket, de alkalmasak 0- +5 V feszültségtartományba átalakított, tetszőleges analóg mennyiségek fogadására is.

Az 1. soros interfész 9 pólusú szubminiatűr ISO aljzat, ide csatlakozik a speciális vtx-modem. A felületen keresztül soros aszinkron adatátvitel 7 bites (1 start-, 7 karakter-, 1 stopbit) vagy 8 bites (1 start, 8 karakter-, 1 stopbit) adatformátumú jelekkel valósítható meg, alapállapotban 75 bit/s adási és 1200 bit/s vételi sebességekkel. Szimmetrikus 300, 600, 1200, 2400 és 4800 bps adatátviteli sebességek is programozhatók. A csatlakozóra természetesen a modemet vezérlő jeleket is kivetették, az elektromos jelszintek nem szabványosak, a modulációs jellemzők V. 23. ajánlás szerinti.

A 2. soros interfész 8 pólusú DIN aljzat, aszinkron, 1200, 2400 vagy 4800 bps adatátviteli sebességre programozható. A jelszintek V. 28. ajánlás szerinti, az adatformátum 8+2 bit. Közvetlenül ide csatlakoztatható a MUPID grafikus tábla vagy többféle, soros bemenetű nyomtató, vagy a MUPID cég által gyártott külső adapteren keresztül tetszőleges gyártmányú kazettás magnó.

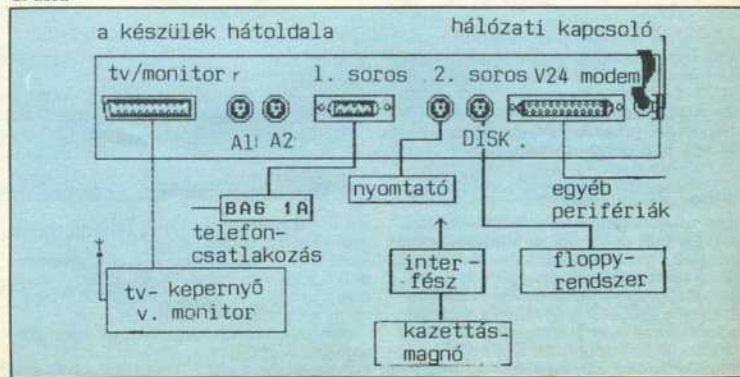
A "DISK" feliratú csatlakozó 8 pólusú DIN aljzat. Kivezetései és azok funkciói hasonlóak a 2. soros interfészhez. Elsősorban a szintén MUPID-gyártmányú M-DISK mágneslemezegység fogadására szolgál.



1. ábra

2. ábra

log bemeneteket (botkormányt is) fogadó A/D átalakító kezelését is ez a mikroprocesszor végzi. A billentyűzet föliaérintós megoldású, az alfanumerikus billentyűktől különválasztva található a számjegy- és hagyományos videotex lekérdező funkciókat (központ hívását, oldal lekérését, oldal ismételt lekérését, felfedését, attributum hatástalanítását stb.) megvalósító billentyűk. A készülékbe tápegységet és hangszórót is beépítettek. Utóbbi elsősorban a billentyűlenyomás akusztikus visszajelzésére szolgál, de zenélő programok is megszólaltathatók rajta — természetesen gyenge minőségben. A központi egység egy SIO áramkörön keresztül tartja a kapcsolatot a külső perifériákkal: a SIO programozását a fel-



tyűtük-e be a gyakraan 8-10 karakter hosszúságú számot, és hogy azt a központ megfelelően vette-e. Törölni a csillaggal ismételt beadásával lehet. Tehát így hívhatjuk le a megjelenítésre szánt információk oldalát, de az információszolgáltatók „válasz”-oldalát vagy más előfűzetéket szánt elektronikus levél (Mail-Box) oldalát is.

Teleprogram lehívásánál a kék K (Kommandó-parancs) billentyűt majd az A betűt kell előzőleg lenyomnunk, mielőtt a teleprogramot tartalmazó oldal első oldalát közvetlenül vagy menüből lehívánk. Így szerez a MUPID tudomást arról, hogy nemcsak a megjelenítendő oldal adatjelét kell fogadni, hanem teleprogramot és azt más memóriaterületre kell betöltenie. (Általában a teleprogram betöltése alatt — melynek hossza jelenleg legfeljebb 24 kB lehet — a felhasználó szíves türelmét kéri és a betöltés várható idejét közlő felirat látható.) Az így betöltött teleprogram vagy automatikusan elindul, vagy a képernyőn látható utasítással — általában egy billentyű lenyomásával — indítható. A teleprogramok helyileg futtathatók, így az összekötöttes a vtx-központtal megszakítható. Vannak olyan programok is, melyek futások közben a vtx-központhoz fordulnak adatkért, így ezeknél a telefonvonalat nem szabad bontani. Az osztrák vtx-központban a MUPID cég több, a MUPID dekóder és a központ együttműködését, valamint a felhasználó komfortosabb kiszolgálását segítő rendszerprogramot helyezett el, melyek egy része a kék K és valamely más billentyű lenyomását követően azonnal, automatikusan betöltődik a dekóderbe. Ilyenek például:

— K I olyan „kulcsszavas” keresőprogramot hív le, amely kikeresi azt az oldalt, ahol egy ABC szerint rendezett listán a keresett téma szerepel és onnan közvetlenül lehívható,

— K M későbbi hívás könnyítése céljából a kért oldalak 0–9-ig megjelölhetők,

— K E szerkesztőprogram lehívása

— K B BASIC program lehívása.

Más rendszerprogramok — nyomtató, másoló, levélszerkesztő stb. — a többi teleprogramhoz hasonló módon tölthetők be.

A központot hívó billentyűt önmagában lenyomva, a MUPID és a központ közötti kapcsolatot bármikor megszakítható. Ha a modem lebonolyítottá az összekötöttes, a sorba kötött telefonkészülék ismét alkalmas a hagyományos telefonbeszélgetésre, a MUPID pedig visszatér alapállapotába. (Folytatjuk)

JURENKA OSZKÁR

ALAPÁLLAPOT

VIDEOTEX
ÜZEMMÓD

HELYI
ÜZEMMÓD

LEMEZEGYSÉG
ÜZEMMÓD

INFORMÁCIÓ
LEKÉRÉS

PROGRAM
FUTTATÁS

INFORMÁCIÓ
LEKÉRÉS

PROGRAM
FUTTATÁS

Komfort HUPIDnál

“RAM-DISK” ÜZEMMÓD

INFORMÁCIÓ
MEGJELENÍTÉS

PROGRAM
FUTTATÁS

3. ábra

A V. 24-es interfész 25 pólusú, szabványos lábkiosztású. A jelszintek a V.28. ajánlásnak megfelelőek. A 9-es lábon a belső +5 V is megjelenik, 100 mA terhelhetőséggel. Az adatformátum itt is 8+2 bites, az átviteli sebesség 300–9600 bit/s közötti szoftverből állítható, az 1200/75, 1200, 2400 és 4800 bit/s sebességek a billentyűzetről menüből is kiválaszthatók.

Az osztrák MUPID háromféle változatban kerül forgalomba:

- Standard-MUPID — a belső felépítés a fent leírtaknak felel meg
- Komfort-MUPID — beépített, akkumulátorral pufferelt 128 kB-os CMOS RAM bővítéssel
- Digi-MUPID — beépített digitalizáló modulal

Ez utóbbira a digitalizáló konfiguráció ismertetésénél még kitérünk. A standard változathoz utólag bármelyik más változat létrehozható, de memóriabővítéssel és digitalizáló modulal együtt nem építhető be. A továbbiak mindhárom változatra érvényesek, a komfortkiegészítést igénylő alkalmazásokat külön említjük meg.

A MUPID üzemmódjai

A MUPID-ot elsősorban a videotex alkalmazások szemszögéből tárgyaljuk, ezen belül is az osztrák vtx-rendszerben való viselkedését. A dekóder lehetséges üzemmódjait mutatja a 3. ábra. Ezeket vagy a felhasználó a billentyűzetről, vagy valamely futó program automatikusan választja ki. A szükséges elemek: MUPID dekóder bármelyik változata, monitor, vtx-modem és M-DISK mágneslemezegység.

Videotex üzemmód

Az 1. soros interfészhez speciális vtx-modemet — azt pedig meglévő telefonkészülékekkel sorosan a távbeszélő-hálóhoz — csatlakoztatva, a SHIFT és a vtx-központot hívó billentyűk együttes lenyomását követően a modem automatikusan felhívja hívószámát vagyis a vtx-központot, amelyet a

PROM-jában tárol. A MUPID alapértelmezésben a vtx-modem automatikus hívásközvetítésére van programozva, s a központtal való kapcsolat felépülte után 75 bit/s sebességgel adja — és 1200 bit/s-mal fogadja — az adatokat. A billentyűzetről megfelelő paranccsal a képernyő 25. sorában megfelelő menü hívható be, amely lehetővé teszi ugyanezen modemmel a manuális hívást, vagy a V. 24-es modem használatával különböző (1200/75, 1200 DX, 2400 DX és 4800 DX) adatátviteli sebességek közötti választást. Ilyenkor a felhasználó dolga, hogy tárcsázzon, és ha a vtx-központtól megjött a megfelelő válaszhang, a billentyűzetről a vonalra kapcsolja a modemet. A sikertelen hívásról — legtöbbször annak okát közlő — üzenet jelenik meg a 25. sorban.

Sikeres hívásra a hívószámhoz tartozó vtx-központ udvarianus bejelentkezik, majd a billentyűzetről bevitt megfelelő karakterekkel azonosíthatjuk magunkat a rendszer számára. Ezután lehetséges a hozzáférés a hön óhajtott információt tartalmazó oldalhoz (menüből való választások útján egyre közelebb juthatunk a célhoz), vagy a keresett oldal számát — ha ezt ismerjük — csillag és kettőskereszt jelek közlé zárva közvetlenül megadhatjuk a központ számára. Utóbbi esetben az oldalszám a 24. sorban megjelenik, de ezt nem a dekóderünk küldi a monitorra, hanem az előbb eljuttatott vtx-központig a 75 bit/s-os csatornán, majd onnan 1200 bit/s sebességgel, mintegy „visszatükrözve” érkezik vissza a dekóderbe, majd onnan a képernyőre. Ily módon ellenőrizhető, hogy helyesen billen-



A szép írásképet mellett a viszonylag kis zaj is jellemzőjük a mátrixprintereknek. Így alkalmasak irodában, gép melletti üzemeltetésre. Manapság szinte csak a szocialista országokban gyártanak fix karakterkészlettel rendelkező berendezéseket. A leggyakoribb típusok „letölthető” karakterkészlettel rendelkeznek. Sőt, a programozási lehetőségeket kihasználó szoftverekkel (ezekből nálunk a LETRIX és a FONTRIX a közismert) lehetőség van változatos betűtípusú, akár magyar ékezetes szövegek előállítására is. A FUJITSU 2400-as típusú nyomtatója a nyugat-európai piacon nagyfokú programozhatóságával tűnik ki. Ezt megkönnyíti a szinte luxusszámba menő 16 karakteres saját LCD displaye.

A nyugat-európai piacon egyre inkább csak tartozékokkal — például lapadagolóval — ellátott nyomtatókat lehet professzionális cétra eladni. Így előállt ugyanaz a helyzet, amely korábban a PC-piacot jellemezte. Az alapgép viszonylag olcsó, de a tartozékok ára összemérhető a berendezésével. Ennek ellenére az ilyen tartozékok alkalmazása a gyakorlatban kifizetődik, mert a hagyományos leporellőknél olcsóbb irodai papírt, illetve hagyományos átirós úralapot lehet használni.

A lapadagolók gyártói szinte külön „kasztot” alkotnak. Igen jó hírnévre tett közzülük szert a RUTISHAUSER, valamint az írógépgyártásáról ismert HERMES-cég.

Az elektronika szinte a végtelenségig továbbfejleszhető. S ez a fejlesztés az egyre terjedő és viszonylag olcsó berendezésorientált mikrochipek segítségével gyorsan és elfogadható költséggel végrehajtható. Ennek szembeszökő példája az OKIDATA 393-as nyomtatója, amely cserélhető emulációs (IBM, EPSON, QUME stb.) modulokkal rendelkezik.

Teljesen új elven alapulnak az új, igen nagy sebességű mátrixnyomtatók. Közülük is kiemelkedik a 800-900 sor/perc (mintegy



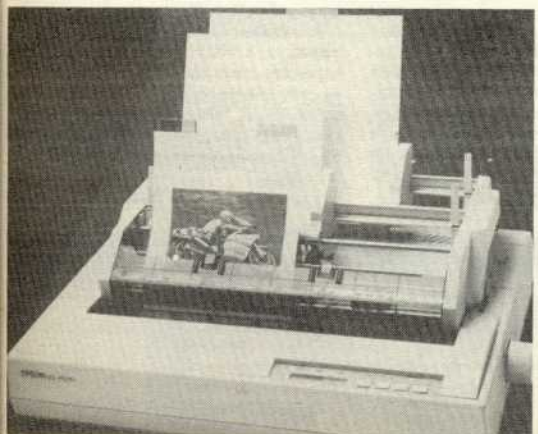
Nyomtatók holnapja..

1970 karakter/másodperc!) sebességű SHUTTLE típusú berendezés. A teljes nyomtatási mezővel szemben egy fűsűfógból álló nyomtatófej helyezkedik el, egy fog/egy karakter sűrűséggel. A fog végén lévő kidomborodás pedig attól függően hagy nyomot a papíron, hogy a hozzá tartozó tekercsben folyik-e áram. Mivel karakterenként egy fűsűfog van, a nyomtatófejet vízszintes irányban csak egykarakternyit kell elmozdítani. Így sikerült egyesíteni a mátrixnyomtatók rugalmas karakterkészletét valamint a nagy sebességű nyomtatást, ami eddig csak a karakterzalagos-kalaposos professzionális nyomtatók sajátja volt, de ott is fix karakterkészletet lehetett csak használni. Ugyanezt az elvet alkalmazza a GENICOM 4410-es és 4440-es típusú vala-

mint a PRINTRONIX P 600 XQ jelű nyomtatója. Ezek a nyomtatók a képet ütéssel alakítják ki, s így 4-5 példány kifogástalan átirására alkalmasak önindígos papíron.

Új elven alapulnak a „resistive ribbon” elvű printerek. Ilyen például a HERMES cég 810-es típusjellet viselő szerkezete. Ennél a típusnál a festékszalag elektromosan vezető, viaszba ágyazott festéket tartalmaz. Az írójegy egy 40 elektródát tartalmazó függőleges túsor, ahol az egyes elektródák (képpontok) átmérője 0,05 mm. Az elektromosan vezető szalagon és az elektródán keresztül áram folyik, amely helyileg felmelegíti a szalagot, és a festék egy 0,1 mm átmérőjű foltban a papírra olvad. Visszamenve

Az EPSON SQ—2500-as típusú tintasugaras nyomtatója e kategóriában megfelelője az LQ—2500-as hagyományos tūmátrixos nyomtatónak. Programozási sajátosságai — így még betölthető karaktergenerátora illetve az LCD kijelzője is — azonosak azzal. Újdonsága, hogy 540 jelet tud kinyomtatni karaktergenerátoros üzemmódban másodpercenként. Ami megkülönbözteti, hogy tintasugaras nyomtatóműve révén, újságminőségben képeket, grafikákat lehet vele előállítani 109 pont/milliméteres felbontással. A nyomdai minőségű betűk generálásánál viszont kissé lassú, csak 180 jelet állít elő másodpercenként.



Az EPSON GQ—3500-as típusú lézernyomtatója, klasszikus konstrukció a maga kategóriájában. A fűvezető henger, az előhívó, valamint a festék egyetlen kazetában egyszerre könnyen cserélhető. A kis teljesítményű (irodai) berendezések közé tartozik, percenként 6 oldal teljesítményével. Az alapfunkciókat az oldallapon lévő kezelőszerveivel lehet beállítani, de természetesen programból is vezérelhető. Egyaránt alkalmas a beépített karaktergenerátorral valamint a RAM-ba betöltött karakterkészlettel való nyomtatásra, illetve grafikus üzemmódban való üzemeltetésre. Színes nyomatot a tónert, előhívót és a hengert tartalmazó kazetta egy kézmozdulattal történő cseréjével állíthatunk elő — természetesen csak egy színben. Jelenleg kék, zöld, piros, barna valamint fekete festékkazettát forgalmaznak. A nyomtató emulációs üzemmódja (HP, IBM, EPSON stb.) az IC kártyára dugaszolható ROM segítségével történik.





Ha körüttekintünk valamelyik nagy számítástechnikai kiállításon, észrevehetjük, hogy a hagyományos 9 vagy 18 tűs nyomtatók placcá telítetté vált. A most forgalomban lévő típusok közös jellemzője a szép íráskép, a megbízható elektronika, a viszonylag egyszerű programozhatóság. Bizonyos szabványok is kialakultak. Így a legtöbb gyártó nyomtatóit IBM vagy Epson kompatibilisre készíti. Ennek megvan az az előnye is, hogy a programcsomagok gyártói ehhez tudnak alkalmazkodni. Ez újabb piaci keresletet jelent.

a kinyomtatott szövegre s alacsonyabb hőmérsékleten futtatva az írófejet, a festék nyomtalanul visszaszorítható a festékszalagra. Nyomtatási sebessége mintegy 40 karakter/s. Csak a fejmozgató motor zaja jelentkezik, ezért igen csendes üzemi. Hátránya, hogy az írás fényes felületű papírról viszonylag könnyen lekopik. Ugyanakkor kifogástalanul fotózható írásképet nyújt.

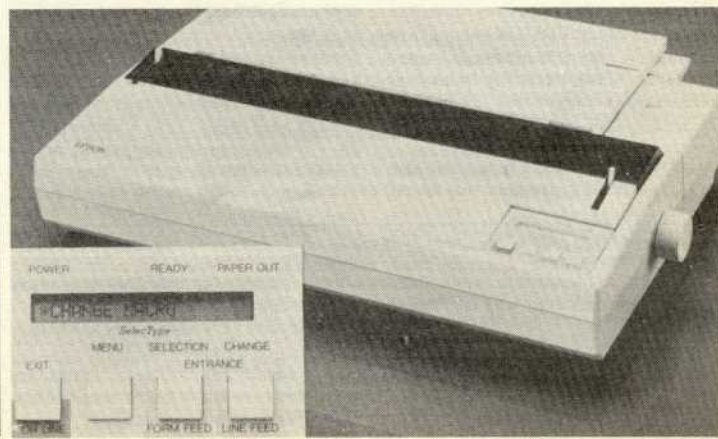
Hasonló átviteli elvet alkalmaznak, csak a festékszalag többszínű, az úgynevezett „thermal transfer” átviteli eljárás alapuló nyomtatók. Mintegy 1 perc alatt kialakul a 3 alapszint és néhány esetben a feketét is tartalmazó viaszos festékszalagok és a hőiró fej segítségével a mintegy 4000 színárnyalatot tartalmazó színes kép. Gyakorlatilag az utolsó előtti lépés a fotóminőségű színes másolatok kialakításához! Jelenleg üzemeltetésük igen drága. RGB interfésszel rendelkeznek, így a színes grafikus kártyák jélevel vezérelhetők. Jelenleg főként a MITSUBISHI G 650 és a MITSUBISHI G 500A nyomtatók uralják a — nem éppen nagy — piacot.

Érdekeség, hogy az ismert „tintaágyús” nyomtatási elvet hasznosító, úgynevezett „Jet ink” nyomtatók is tudtak színes újdonsággal szolgálni. A jelenlegi „sláger” a DATAPRODUCTS SI 480 SOLID INK nyomtató. Benne a megszokottól eltérően a tinta szilárd pasztillában kerül betöltésre, és csak a nyomtatófej hőjének hatására válik folyékonnyá. 32 függőlegesen elhelyezett fűvőkán keresztül csapódik a papírra a tinta. Jelenleg csak a fekete változatát gyártják, a színes most van fejlesztés alatt.

480 pont/inch a felbontása, sebessége 400 karakter másodpercenként.

Egyre többen gyártanak xerografikus elven működő nyomtatót (lézerprintert). Itt a mindennapi gyakorlat az, hogy néhány nagy gyártó nyomtatóművét építik össze saját fejlesztésű elektronikával. A legelter-

Az LQ-2500 24 tűs nyomtató, egyike az EPSON legkorszerűbb berendezéseinek. Normál — karaktergenerátoros — üzemmódban 324 karaktert ír másodpercenként, betölthető karaktergenerátora egyszerre öt betűkészletet képes alkalmazni. A programozást saját LCD display könnyíti meg. Levélminőségű nyomtat ad. Soros (RS 232C) valamint párhuzamos (CENTRONICS) interfésszel rendelkezik.



jelebebeo beépített nyomtatóművet a TOSHIBA készíti (fő felhasználói a Dataproducts és a WEGNER), míg mások általában Canon (Mannesmann Tallya) vagy éppen Xerox nyomtatóművet alkalmaznak. A képet lézer forgótűkkel vagy piezokristályú vezérelt rezgőtűkkel elérített sugár alakítja ki a szerves félévezető vagy szelénhenger felületén. Sebességük általában 8-40 lap/perc teljesítményig terjed.

Újdonságszámba megy a kialakult konstrukciós elvek között a CENTRONICS PAGE PRINTER 8 típusú nyomtatonál, hogy a hagyományos szelénhenger helyett a félévezető réteget egy fóliára vitték fel. Technológiai újdonságot alkalmaz a LIQUID CRYSTAL SHUTTER nyomtató — amelyet a hannoveri CeBIT kiállításon láthatott először a közönség: egy folyadék-kristályos fényrekesz és egy hagyományos megvilágító pontlámpa kombinációjával alakítja ki a képet a fotofélévezető tartalmazó henger felületén. Nem alkalmaz lézert. Felbontása ennek ellenére igen jó, 300 pont/inch.

Érdekeség a lézernyomtatók működési elvét felhasználó, de nem lézeres nyomtatómű, a C.I.TOH CIE 3000-es. Itt a lézer helyét egy ionokat kibocsátó fej veszi át, amely létrehozza a félévezető hengeren a töltésképet. A továbbiakban minden a lézernyomtatóknál megszokott xerox elven történik.

Az abszolút csúcs a nyomtatók piacán jelenleg a japán TOSHIBA cég színes digitális nyomtató-másoló berendezése, amelynek nyomtatóműve valószínűleg sok készülékben fog megjelenni az elkövetkező években.

A karakterszintetizáló nyomtatók — a grafikus lehetőség, a könnyen programozható, cserélhető karakterkészlet miatt — nagy jövő előtt állnak. Ugyanakkor a fix karakterkészlettel rendelkező — karakter-szalagos, gömbfejes, margarétafejes, karakterdobos — nyomtatók, bár belőlük készült jelenleg a legtöbb, rövidesen a számítástechnika ősközületei közé fognak tartozni, s rövidesen eltűnnek a gyártók, kereskedők termékpalettájáról. (VIDEOTON RT. INFO nyomán) KIS JÁNOS

Szoftver

Gikksorozatunkban először a gépi kódú programozást segítő három legfontosabb eszközzel, az EDITOR, az ASSEMBLER és a MONITOR programmal kapcsolatos alapvető ismereteket mutatjuk be. Aki gépi nyelven akar programozni, annak ezekre a programokra feltétlenül szüksége van.

Az EDITOR

A TEXT EDITOR-okról általában

A TEXT EDITOR magyarul szövegszerkesztőt jelent. Segítségével a felhasználó szövegfájlokat hozhat létre, illetve a már meglévőket módosíthatja.

Szövegfájlokban nevezzük azokat az adathalmazokat, amelyek karaktereket tartalmaznak, és megjeleníthetők a számítógép képernyőjén vagy nyomtatón kinyomtathatók. Szövegfájl például egy Z80 assembly nyelven megírt ún. forrásnyelvi program, egy programhoz tartozó, az adathordozón (szalagon vagy lemezen) lévő leírás vagy kezelési útmutató.

Az EDITOR a számítástechnikában az ember-gép kapcsolat megvalósításának legfontosabb eszköze. Az iratok, levelek, dokumentációk számítógépes kezelését biztosító irodai szövegszerkesztő rendszereknek is alapvető, legfontosabb programja. Mivel használata programozási ismereteket nem igényel, kitűnő lehetőséget teremt a számítástechnikával való megismerkedésre.

Hogyan működik az EDITOR program?

Kezdetben az ún. parancs módban arra vár, hogy a felhasználó megadjon egy parancsot. Miután ez megtörtént, átvált a végrehajtási módba, végrehajtja a parancsot, majd ismét parancs módba vált, és ez a folyamat ismétlődik.

A felhasználó az EDITOR-nak szóló parancsokat és a bevinni kívánt szöveget a számítógép billentyűzetén írja be. A begépett karakterek megjelennek a képernyőn. Ezt a visszairási módot echozásnak (echo = visszhang) nevezik.

Az EDITOR a szöveges információ alábbi egységeivel tud dolgozni:

- betű (karakter),
- szó (elején és végén betűkózzal elválasztott karaktersorozat),
- mondat (a mondatáról jeltől (. ! ?) a következő ilyen jelig tartó karaktersorozat),
- bekezdés (a RET karaktertől a következő RET karakterig tartó karaktersorozat),
- blokk (egy kijelölt szövegrész).

A megadható parancsokat hat csoportra oszthatjuk:

- munkaterület-kezelő (buffer management)
- mutatómozgató (pointer move)
- szövegbeillesztő (insert)
- szövegkereső (search)
- szövegtörölő (delete)
- egyéb speciális parancsok

Munkaterület-kezelő parancsok

Az EDITOR az ún. munkaterületen végzi a szövegszerkesztést. A munkaterület az írható-olvasható memória egy darabja, ide kell elhelyezni a módosítandó vagy beírandó szövegfájlt. A munkaterület mérete véges, így gyakran megesik, hogy a teljes szövegfájl nem fér el benne. Ezért a nagyméretű fájlok szerkesztéséhez az ún. lapozási technikát alkalmazzák.

Ennek az a lényege, hogy a munkaterületre betöltik a szövegfájl akkora darabját, amekkora éppen elfér ott, majd az ezen a részen végzendő javítások végrehajtása után a módosított részt elmentik, és a munkaterületre a fájl következő részét töltik be. Egy lapnak nevezik a szövegfájl munkaterületre betöltött részét. Ezzel a megoldással elméletileg végtelen méretű fájlok szerkesztése is megoldható.

A szövegfájl soron következő lapját a forrásbemenetről tölthetjük be a munkaterületre. A forrásbemenet lehet kazettás magnó vagy hajlékonylemez. A megszerkesztett vagy megírt lapot a munkaterületről a forráskimenetre menthetjük, ami az előbbi perifériák valamelyike lehet.

A munkaterületet kezelő típusú parancsok a következők:

- a munkaterület feltöltése a forrásbemenetről,
- a munkaterület egészének vagy részének kimentése a forráskimenetre,
- lapcsere (a munkaterület kimentése, majd a soron következő lap betöltése),
- a munkaterület listázása (ennek során a munkaterület tartalma megjelenik a képernyőn vagy a nyomtatón).

Mutatómozgató parancsok

Mivel a munkaterületen a fájl egy tekintélyes darabját lehet egy időben tárolni, az EDITOR számára itt meg kell jelölni azt a pontot, amelytől kezdve végrehajtsa a megadott parancsot. Az erre szolgáló eszközt mutatómozgató (pointer) nevezzük. Ez tulajdonképpen egy, a tárolóban őrzött számérték, amelynek értékkelésletét a munkaterületen tárolt karakterek lehetséges címei adják.

Ha az EDITOR mutatóját betűnként (karakterenként) lehet mozgatni, akkor karakterorientált, ha csak soronként lehet mozgatni, akkor sororientált EDITOR-ról beszélünk. A típusú mutatómozgató parancsok a következők:

- a mutató megadott számú karakterrel/sorral előre mozgatója,
- a mutató megadott számú karakterrel/sorral vissza mozgatója,
- a mutató beállítás a megadott című karakterre/sorra,
- a mutató beállítás a munkaterület elejére (a legelső karakter/sor elé),
- a mutató beállítás a munkaterület végére (a legutolsó karakter/sor mögé).

A munkaterület végén általában nem a puffér fizikai végét, hanem a karakterekkel még fel nem töltött munkaterületrész elejét értjük.

Szövegbeillesztő parancsok

Szövegbeillesztésnek (insert) nevezzük azt a műveletet, amelynek során beillesztjük a munkaterületre a begépett karaktereket. Az új karakterek az aktuális karakter (sor) elé kerülnek, amely az utána következővel együtt a munkaterület vége felé elmozdul.

Szövegtörölő parancsok

A szövegtörölés (delete) során a felhasználó által megadott számú karaktert (sor) kitöröljük a munkaterületről. A kitörölt részt követő karakterek a munkaterület eleje felé elmozdulnak.

Ígéretes kezdet



Szövegkereső parancsok

Szövegfájlok javításánál gyakran nem tudjuk pontosan, hogy a javítandó sor vagy karakter a fájl elejétől számított hányadik sorban helyezkedik el. A javítás helyének környékén azonban többnyire található valamilyen jellegzetes szövegrészlet. Ilyen esetekben jól használhatjuk a szövegkereső parancsok valamelyikét. Miután megadtuk a keresett szöveget — általában maximum 16–32 karakter lehet —, az EDITOR a munkaterületen megkeresi azt. Ha megtalálja, rendszertől függően vagy a keresett szöveget tartalmazó sor elejére (sororientált) vagy a keresett szöveg mögé (karakterorientált) állítja a mutatót. Ha a keresés sikertelen — az adott szöveg nem fordul elő a munkaterületen —, a rendszer jelzi ezt.

A szövegkereső parancsok csoportjába sorolhatók a szövegcsere (text-change) parancsok is. Hatásukra az EDITOR megkeresi a megadott régi szöveget, törli a munkaterületről, és helyére illeszti az ugyancsak begépelte új szövegrészt.

Speciális EDITOR parancsok

A különféle EDITOR-ok az általános parancsokon túlmenően több-kevesebb speciális parancs végrehajtására is képesek. Ezek célját és felhasználási módját a felhasználási leírásokból ismerhetjük meg.

Megjegyzés

A fentiekben általános képet mutattunk az EDITOR-okról. A jelenleg használt korszerű szövegszerkesztők már kizárólag karakterorientáltak, egy kijelzett képernyőnyi szöveg tetszőleges része szerkeszthető, a kurzorral a képernyő teljes felületén mozoghatunk. Ennek az az előnye, hogy a szöveg azonnal látható a módosítás. Ezek a szövegszerkesztők speciális alkalmazásokra „kihelyezhetők”, azaz hozzáigazíthatók egy-egy programnyelv szövegének beviteléhez, ami azt jelenti, hogy egy-egy speciális billentyű megnyomásakor egy egész — például nyelvi — alapszó jelenik meg a szövegben. Az egyre nagyobb méretű belső memóriák nagy munkaterületek használatát teszik lehetővé.

TISZAI TAMÁS—
DR. KÓNYA LÁSZLÓ

Volt egyszer egy kisfiú, egy nyár és egy számítógép. Az első és az utolsó a középsőben talált egymásra. Hogyan találtak az ismerkedés? Tartózkodással indult, aztán a kíváncsiság, a rajongás következett. Minden idejét, figyelmét lekötötte a számítógép nyugtató élmény.

A profi számítógépesnek mindez nem sokat mond, hiszen számára természetes és nem csoda a programkészítés, a programnyelv használata. De kérdem én, egykoron nem mindenki hasonlóan kezdte? Senkinek sem hullott ölébe a mostani tudás. Játék, igenis csoda, csodajáték lehetett a számítógép, az, ami Gönczi Dávidnak, a nyolcesztendő kisfiúnak ma.

— Anyu kérdezte a nyáron, akarok-e számítógépes táborba menni. Olyanba, ahol sokat játszhatok, meg még úszhatok is. Osztálytársaimtól hallottam, mennyire klassz dolog a számítógépezés, hogy fél napokat elszórakoznak a különböző játéckomplexumokkal. Egyszer egy barátom hívott, hogy náluk játszunk, neki van ugyanis Commodore 64-e. Aztán elmaradt az egész, mert beteg lettem. De azért nagyon kíváncsian vártam a találkozás pótlását.

Az Almássy Téri Szabadidőközpontban a Compania társaság szervezte nekünk, gyerekeknek ezt a vakációs programot. Egy időben híreszlál is többen gyűltünk össze a számítógépes teremben. Az alsókkal és a felsősökkel külön foglalkoztak, úgy, hogy más és más feladatokat adtak nekünk. Egyetemista bácsik vigyáztak ránk, és szintén ők magyarázták el a gépek használatát.

Választék bőven volt, kinek melyik tetszett, ahhoz ült. Én mindjárt a C Plus/4-et választottam, bár akkor még nem is tudtam, hogy az milyen. Egy ismerősről a HT-t nézte ki, egy másik a Videoton TV-Computer. De ezeken kívül volt ott még C64, C16 és Primo is. Nem tudom miért a Plus/4 mellett döntöttem, hirtelen született az elhatározás. Egy-egy géphez két gyereket osztottak, s ez azért is jó volt, mert miután elmagyarázták a gép tulajdonságait, a billentyű jelentéseit és kezelését, akkor kezdődhetett a játék. A párommal csudára drukkoltunk egymásnak. Borzasztóan izgultunk, hogy mi nyerünk, méghozzá nagyon sokat, s ne a számítógép fogjon ki rajtunk.

Azután amikor már jól kiszórakoztuk magunkat, akkor az egyetemista bácsi megtanított bennünket a programozás alapjaira. Naponta másfél óránk telt el így, vagyis ez az idő rendkívül hasznos volt. Sikerült olyan játékosan tanulunk, hogy észre sem vettük azt, hogy ott oktatás folyt. Mindenesetre az egészet nem úgy fogtuk fel, mint a szigorú iskolai tanulást. Talán ezért haladtunk gyorsabban és eredményesebben.

Nem tagadom, a legjobb mégiscsak a játék. Viktor barátommal eleinte a Timeslip játékprogram volt a kedvencünk. Eleinte persze követtünk el hibákat. Amikor például a programmal tápláltuk a gépet, véletlenül másik jélt jött be. Természetesen nem tudtuk mindjárt, hogy az ismeretlen játékot hogyan kell irányítani. A Dorks programban dinamitokat lehetett lerakni, és kikezdeni a látogók szörnyetegekkel. Amikor végiggondoltam a játék tartalmát, már közel sem tetszett annyira. Kicsellezni a másikat — még ha szörnyetegekről van szó, akkor is — úgy, hogy az nem veszi észre, mit csinálunk vele, egészen biztos hogy nem tisztséges.

De folytatom. Szóval, amikor megint betápláltuk a programunkat, ismét elhibáztuk, újból egészen más jött be. Ezt Killerwat-nek nevezik. A barátom kedvenc játéka lett. Bálnákat, struccokat és kísérleteket lehetett kilőni. Az az igazság, hogy nekem ez sem nyerte el a tetszésemet. Ami viszont igen, az az óras program volt. Összesen 40 órát állhattam el igen rövid idő alatt. Ez is volt a játék célja. Meg az, hogy a refleximeit próbára tegye a feladat, ugyanis 1 órára 10 másodpercet kaptam. Ügyeskedni, figyelni kellett a játékban, s ha az ember elkelekötyáskodta a dolgokat, csúnyán veszíthetett. A végén már megszállottként próbáltam minél rövidebb idő alatt, a lehető legnagyobb pontszámot összegyűjteni. Kicsit be is csavarodtam. Egy ideig még éjszaka sem almodtam másról, mint a különböző programokról.

Miután szüniözögtem számítógépezétk, úgy szervezték, hogy délutánonként úszunk. A testmozgás jó és hasznos, de sajnos elvon a monitor mellől. Így tehát, amikor sikerült, akkor kicselleztem az úszmestert és visszaosontam a számítógépterembe. Egy nagyobb sráccal haverok lettünk — délután ugyanis a felsősök használták a gépeket —, és megengedte, hogy tovább játsszam a gépen.

Mit mondjak még? Vidáman, klasszul telt a vakációmnak ez a része. Mindenképpen jobban, hasznosabban, mint azoknak a gyerekeknek, akik kulccsal a nyakukban ezalatt az utcán csavarogtak. Azóta meg hétvégeken folytatom a játékprogramok futtatását a szabadidőközpontban. Hétközben egy nagyfiú barátom programírásra tanít, továbbá tudja mindazt, amit az iskolában oktatnak neki.

Hát így indult Dávidnak az ismerkedés...

Krasznai Éva

Memóriaolvasó program

A program a gép memóriacímeinek értékét írja ki hexadecimális formában, valamint a cím tartalmának megfelelő karaktert.

RUN parancs után a program bekéri azt a két címet, ahonnan az olvasás kezdődjön, illetve ahol befejeződjön. Ezek után megindul az olvasás, melynek sebessége 6 cím/másodperc.

A kinyomtatott sor formája:

A sor első értékének címe (4 karakter + 1 szóköz),

8 db kétjegyű hexadecimális szám, a címek értékei a sor elején kiírt memóriacímtől kezdődően,

8 db negatív karakter, a címek értékeinek megfelelő karakterek.

Ha a memória értékéhez nem tartozik megjeleníthető karakter (például 14 = kisbetűváltó), akkor a program negatív pontot ír. A kiírás a CTRL gomb lenyomásával lassítható, az F7 gomb lenyomásával leállítható, illetve indítható. A program C64, C16 és Plus/4 gépen változtatás nélkül használható. VC20 esetén a következő változtatásokat kell beírunk:

```
30 FOR I=E TO V STEP 4
60 FOR J=I TO I+3
```

A program a tisztán gépi kódban írt assembler fordítók futását nem zavarja.

STUMPHAUSER TAMÁS

```
10 INPUT E,V
20 IF E<0 OR E>65535 OR V<0 OR V>65535 THEN 10
30 FOR I=E TO V STEP 8
40 GOSUB 150
50 KK$=A$+" " : A$=""
60 FOR J=I TO I+7
70 A=PEEK(J) : GOSUB 280
80 KK$=KK$+A$+" " : A$=""
90 NEXT
100 KK$=KK$+"/2/" + K$ : K$=""
110 PRINT KK$
120 GET B$ : IF B$="/F7/" THEN GOSUB 430
130 NEXT
140 RUN
150 A(1)=INT(I/(16*3)) : A=A-(A(1)*(16*3))
160 A(2)=INT(A/(16*2)) : A=A-(A(2)*(16*2))
170 A(3)=INT(A/16) : A=A-(A(3)*16)
180 A(4)=A
190 FOR X=1 TO 4
200 IF A(X)<10 THEN A$(X)=MID$(STR$(A(X)),2,1)
210 IF A(X)>9 THEN A$(X)=CHR$(A(X)+55)
220 IF A$(X)="" THEN A$(X)="0"
230 NEXT
240 A$=""
250 FOR X=1 TO 4
260 A$=A$+A$(X)
270 NEXT : RETURN
280 B=A
290 A(1)=INT(A/16) : A=A-(A(1)*16)
300 A(2)=A
310 FOR X=1 TO 2
320 IF A(X)<10 THEN A$(X)=MID$(STR$(A(X)),2,1)
330 IF A(X)>9 THEN A$(X)=CHR$(A(X)+55)
340 IF A$(X)="" THEN A$(X)="0"
350 NEXT
360 A$=""
370 FOR X=1 TO 2
380 A$=A$+A$(X)
390 NEXT
400 IF B>31 AND B<128 THEN K$=K$+CHR$(B) : RETURN
410 IF B>159 AND B<192 THEN K$=K$+CHR$(B) : RETURN
420 K$=K$+"." : RETURN
430 GET B$ : IF B$="/F7/" THEN RETURN
440 GOTO 430
```

Felhasználói karakterek

A Spectrum SCREEN \$ függvénye nem ismeri fel a felhasználói grafikus karaktereket, de erre egy rövid programmal megtanítható. A 23606 és a 23607 címeken található értékek a karakterkészlet kezdőcímét határozzák meg. Nincs más dolgunk, mint ide az UDG terület címét írni, pontosabban annál 256-tal kevesebbet. Ezután a SCREEN \$ a felhasználói grafikus karakterek között keresi a képernyőn levő jelet. Vég-

rehajtás után ne feledkezzünk meg az eredeti karakterkészlet címének visszaírásáról:

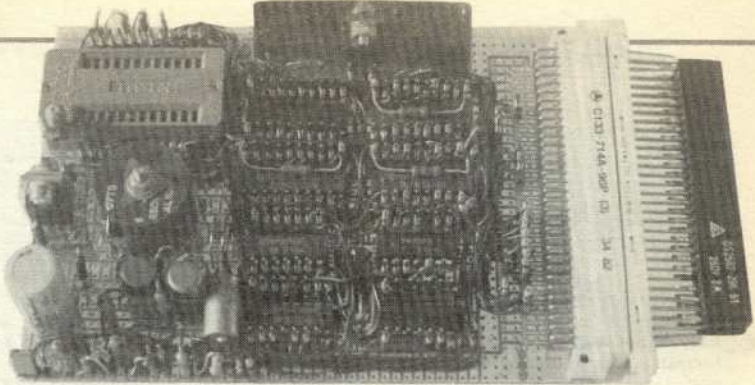
```
POKE 23606,0: POKE 23607,60
Természetesen a két „üzemmód” együtt is használható, például így:
10 IF SCREEN $(x,y)="" THEN GO TO 40
20 LET a$=SCREEN $(x,y)
30 GO TO 80
40 LET s=USR "a"
```

ZX-SPECTRUM

```
50 POKE 23606,s-256*INT (s/256)
60 POKE 23607,INT (s/256)-1
65 LET a$=SCREEN $(x,y)
70 POKE 23606,0: POKE 23607,60
80 PRINT AT 0,0: a$
```

Az x sorszámra, y oszlopszámra írt normál vagy felhasználói grafikus karaktert a program felismeri, az a\$ változóba helyezi, majd a 0,0 pozícióra kiírja.

MOLNÁR LÁSZLÓ



2—4 kbájtos EPROM- a Spectrumhoz

Az amatőr gyakorlatban legtöbbször a 2 és 4 k-s EPROM-ok égetése fordul elő. Az alábbi „toldalék” a 2716, 2732 és 2732A típusú EPROM-ok égetésére szolgál.

A BASIC sorok bebillentyűzése és magnóra mentése után (RUN 9999) a program a későbbi betöltéseknél automatikusan indul. Figyelem! Az EPROM-égetőt a Spectrummal *csak kikapcsolt állapotban* szabad csatlakoztatni, majd bekapcsolás után lehet a BASIC programot betölteni. Minden más esetben előre nem látható károk keletkezhetnek számítógépünkben!

A program leírása

- 170 a RAMTOP beállítása
- 180 a C üzemmód beállítása
- 190—250 a gépi kód betöltése
- 260 2 k üzemmód beállítása
- 270 alaphelyzet-beállítás
- 290—390 a menü kiírása a képernyőre
- 400—500 billentyűfigyelés
- 510—540 magnóra mentés
- 550—570 betöltés magnóról
- 580—640 EPROM-beolvasás
- 650—700 égetés-ellenőrzés
- 710—790 a kiválasztott EPROM égetése
- 800—950 programozás a memóriában
- 960—1040 2 k rendszerváltozók beállítás
- 1050—1130 4 k rendszerváltozók beállítás
- 1140—1330 a gépi kódú rutinok területe
- 9999— magnóra mentés és automatikus indítás

A betöltés után a program automatikusan indul és a képernyőn az ADATHALMAZ TÖLTÉSE A MEMÓRIÁBA szöveg jelenik meg. Ezalatt a Spectrum a DATA sorokban elhelyezett gépi kódokat egy adott memóriaterületre tölti. A töltés végeztével megjelenik a menü:

- MAGNÓRA MENTÉS (S)
- BEOLVASÁS MAGNÓRÓL (L)

- EPROM—BEOLVASÁS (B)
- ÉGETÉS (E)
- ELLENŐRZÉS (V)
- PROGRAMOZÁS (P)
- EPROM TÍPUSA (2—4)

A program működése

Magnóra mentés. Az S gomb hatására a program magnetofonra menti az aktuális memóriaterületet (E000—E7FF vagy E000—EFFF). Ha más elfoglaltság miatt nem tudjuk teletölteni az adott memóriaterületet, akkor az addig megírt részt elmenthetjük magnetofonra, hogy másnap ott folytathassuk, ahol abbahagytuk.

Beolvasás magnóról. Az L gomb választása esetén a program betölti a magnetofonról a már előzőleg kimentett, 2 vagy 4 kbájtos hosszúságú adathalmazt.

EPROM-beolvasás. Ha a B gombot nyomjuk le, megjelenik a következő szöveg: HELYEZD AZ EPROMOT A FOG-

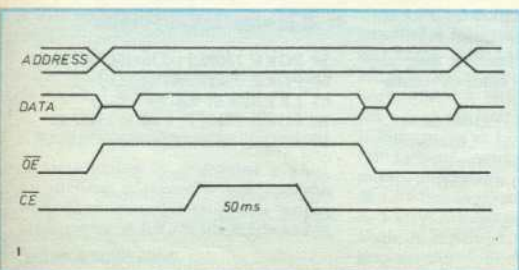
LALATBA ÉS GOMBNYOMÁS. Az EPROM behelyezése és ismételt gombnyomás után az EPROM tartalma beíródik az E000-ás címtől.

Égetés. Az E gomb hatására a program megkérdezi a kezelőt: ÉGETNI AKARSZ(I), védekezve a véletlen gombnyomás ellen. Az I gomb lenyomására a memóriában tárolt információ beíródik az EPROM-ba. Minden más gomb hatására a program visszatér a menühöz. Az égetés végét sípszó jelzi, majd a menü jelentkezik. Az égetés idején a megszakításengedélyezés le van tiltva, hogy az égetés alatt ne következhesen be megszakítás az ULA felől.

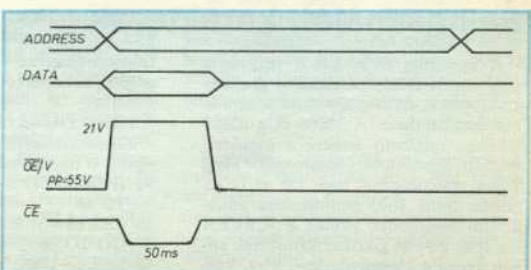
Ellenőrzés. A V gomb lenyomására a program összehasonlítja az EPROM és a memória tartalmát. Hiba esetén kiírja a hibás bájtok számát (a maximális érték 255).

Programozás. A P gombbal a memóriaterületet lehet szerkeszteni, rendezni, javítani. Kilépés: P+ENTER, léptetés előre:

1. ábra. A 2716 égetési diagramja



2. ábra. A 2732 égetési diagramja



ENTER, ugrás előre 15 bájtot: 0+ENTER, ugrás előre 255 bájtot: 1+ENTER. Az E000-ás címtől progamozzhatunk gépi kódban 0—F-ig, más gomb hatására a program hibaüzenettel leáll. Indítása: GO-TO 280.

EPROM típusa. Ha a 2—4 gombot választjuk, a képernyő aljára kiírja az aktuális EPROM típusát (2 vagy 4 bájtt), és átírja a használt rendszerváltókat is. A program nem tesz különbséget a 2732 és a 2732A kö-

égető

zött. Erre a programozónak kell figyelnie, hogy a 2732-re a 25 V, a 2732A-ra 21 V égetőfeszültség jusson. A tápegységnél e célra egy kapcsoló szolgál.

A BASIC program viszonylag egyszerű, ezért némi tanulmányozás után bárki kedve szerint egyszerűsítheti vagy bonyolíthatja.

A rendszerváltók területe

D000 } az inicializált EPROM kez-
D001 } dőcíme beolvasáskor
D002 } az EPROM típusa
D003 } a program hossza
D004 } ellenőrzéskor a hibabit helye

A gépi kódok kezdete D010 = 53264
A memóriaterület kezdete E000 = 57344

A bemeneti és kimeneti címek

FF1F kimeneti címen az égetendő bájtt jelenik meg

FF3F kimeneti címen az alsó (A₀-A₇) bájtt jelenik meg

FF5F kimeneti címen a felső címbájtot (A₈—A₁₁), valamint a CE, OE, a 2732 égetőfeszültség engedélyezése és az adattároló engedélyező jele

FF9F bemeneti címen olvasható az EPROM tartalma

FFBF kimeneti címen engedélyezi a 2716 21-es lábára az 5 V-ot

FFDF kimeneti címen engedélyezi a 2716 21-es lábára az égetőfeszültséget (25 V)

FFF7 kimeneti címen tiltható a feszültség engedélyezése 2732 esetén

Az olvasási folyamat

FF5F címre küldjük a felső címeket és beállítjuk a CE OE jeleket,

FF3F címre küldjük az alsó címeket,

FF9F címről az akkumulátorba töltjük az adatot, az akkumulátorból kiküldjük egy memóriacímre, amit a HL értéke határoz meg. Ez a művelet ciklusba van szervezve és addig tart, amíg a "Program hossza" rendszerváltó értéke egyenlő a HL értékével.

Az írás folyamata

FF5F címre küldjük a felső címeket és beállítjuk a CE OE jeleket,

FF3F címre kiküldjük az alsó címeket,

FF1F címre kiküldjük az égetendő bájtot, CALL égető rutin.

Ez a folyamat is ciklusba van szervezve és addig tart, amíg a "Program hossza" rendszerváltó értéke egyenlő a HL regiszter értékével.

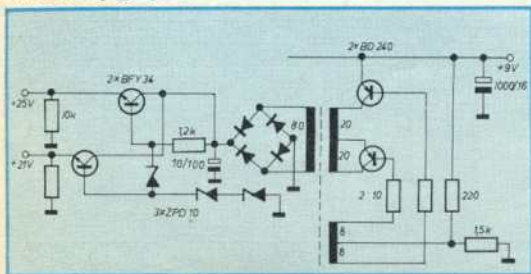
Fontos! Bekapcsoláskor és kikapcsolás előtt nem lehet EPROM az égetőben, továbbá nem ajánlott a bekapcsolt Spectrum-ról lehúzni vagy arra felhelyezni az EPROM-égetőt!

Az összeszerelés

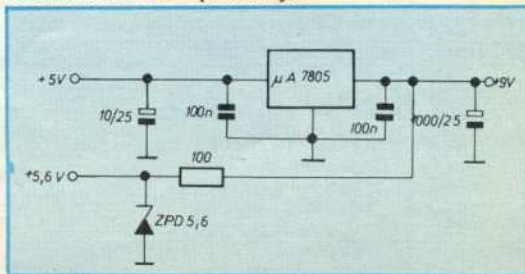
Az összeszerelést célszerű univerzális NYÁK-lemezen végezni. Az IC-eket az EPROM kivételével érdemes beforrasztani, mert a jó foglalat drága, az olcsóbbal pedig sok bosszúságot okozhatunk magunknak. Az összehozalozásnál a Wire-Wrap technológiát használtam, de természetesen forrasztani is lehet. Vigyázzunk a zárlatra és az elkötésre, ellenőrizzük többször is, mert nagy károkat okozhatunk. Az égetőt DS 868 FC 64 típusú Euro-csatlakozóval látam el. A Spectrum és az égető közé egy közdarabot tettem, amely a gép felől egy levágott DS 2582—296 51-es csatlakozó, az égető felé pedig egy DS 868 EC 64 Euro. Azért választottam ezt a megoldást, mert az Euro-csatlakozó olcsóbb is és könnyebben beszerezhető. Ezután ha bővíteni akarjuk rendszerünket, már csak Euro-csatlakozóra van szükségünk. Az összehozalozás előtt csipjük ki az Euro-csatlakozó A 1, 2, 31, 32 és C 1, 2, 31, 32 lábait, mert ezeket nem használjuk. A számítógépből az alábbi jelvezetésekre van szükség:

A Spectrum kivezetései	Az Euro kivezetései
D0 = 6A	C25
D1 = 7A	C24
D2 = 8A	C23
D3 = 11A	C20
D4 = 12A	C19
D5 = 10A	C21
D6 = 9A	C22
D7 = 3A	C28
IORQ = 17A	C14
RD = 18A	C13
M1 = 24A	C7
A5 = 23B	A8
A6 = 22B	A9
A7 = 21B	A10
"O" = 7B	A24

3. ábra. A tápegység



4. ábra. A transzverter kapcsolási rajza



```

100 REM
110 REM ++++++
120 REM +
130 REM + HARAY GYONGVI +
140 REM +
150 REM ++++++
160 REM
170 CLEAR 53247: REM CFFF-H
180 POKE 23658,8: REM NAGYBETU
190 PRINT AT 5,0:"ADATHALMAZ TO
LISE A MEMORIABA"
200 LET B=53264: REM D010-H
210 FOR A=1 TO (20+34+6+16+37+8
+77+70)
220 READ C
230 POKE B,C
240 LET B=B+1
250 NEXT A
260 GO SUB 960
270 RANDOMIZE USR 53264
280 CLS
290 PLOT 30,16S: DRAW 200,0: DR
AW 0,-116: DRAW -200,0: DRAW 0,1
16
300 PRINT AT 2,5:"MAGNORA ME
NIES (S)"
310 PRINT AT 4,5:"BEOLVASAS MA
GNOROL (L)"
320 PRINT AT 6,5:"EPROM BEOLVAS
AS (B)"
330 PRINT AT 8,5:"EGETES
(E)"
340 PRINT AT 10,5:"ELLENORZES
(U)"
350 PRINT AT 12,5:"PROGRAMOZAS
(P)"
360 PRINT AT 14,5:"EPROM TIPUSA
(2-4)"
370 BRIGHT 1: FLASH 1
380 IF H1=2 THEN PRINT AT 20,12
"2-KBYTE": BRIGHT 0: FLASH 0: G
O TO 400
390 PRINT AT 20,12:"4-KBYTE": F
LASH 0: BRIGHT 0
400 IF INKEY{<}="" THEN GO TO 40
0
410 IF INKEY{<}="" THEN GO TO 410
420 IF INKEY{<}="S" THEN GO TO 51
0
430 IF INKEY{<}="L" THEN GO TO 55
0
440 IF INKEY{<}="B" THEN GO TO 58
0
450 IF INKEY{<}="2" THEN GO SUB 9
60
460 IF INKEY{<}="4" THEN GO SUB 1
050
470 IF INKEY{<}="E" THEN GO TO 71
0
480 IF INKEY{<}="P" THEN GO TO 80
0
490 IF INKEY{<}="V" THEN GO TO 65
0
500 GO TO 370
510 CLS
520 IF H1=2 THEN SAVE "2-KBYTE"
CODE 57344,2048: GO TO 280
530 SAVE "4-KBYTE"CODE 57344,40
96
540 GO TO 280
550 CLS
560 LOAD ""CODE 57344
570 GO TO 280
580 CLS
590 RANDOMIZE USR 53264
600 PRINT AT 8,0:"HELYEZD AZ EP
ROMOT A FOGLALATBA"
610 PRINT AT 10,8:"ES GOMBNYOMA
S"
620 PAUSE 0
630 RANDOMIZE USR 53284
640 GO TO 280
650 CLS
660 RANDOMIZE USR 53340
670 IF PEEK 53252>=1 THEN PRINT

```

```

AT 8,5:"EGETES ";PEEK 53252:"
HIBAUÁL !": POKE 53252,0: GO TO
690
680 PRINT AT 8,4:"AZ EGETETT PR
OGRAM JO !!"
690 PAUSE 0
700 GO TO 280
710 CLS
720 PRINT AT 6,5:"EGETNI AKARSZ
? (I)"
730 IF INKEY{<}="" THEN GO TO 73
0
740 IF INKEY{<}="" THEN GO TO 740
750 IF INKEY{<}="I" THEN GO TO 2
80
755 CLS : PRINT AT 10,10:"EGETE
S"
760 RANDOMIZE USR 53385
770 RANDOMIZE USR 53264
780 BEEP 1
790 GO TO 280
800 CLS
810 LET CIM=57344
820 DEF FN A(A1,B)=CODE A{B}-4
8-7*(CODE A{B}>57)
830 DEF FN C(A1)=16*FN A(A1,1)+
FN A(A1,2)
840 DEF FN G(F)=CHR# (F+48+7*(
F/9))
850 DEF FN H(E)=FN G{INT (E/1
6)}FN G{(E-16)*INT (E/16)}
860 DIM A{2}
870 PRINT FN H{PEEK CIM}:TAB 7
:FN H{INT (CIM/256)}:FN H{CIM-
256*INT (CIM/256)}:
880 INPUT A{
890 IF A{1}="P" THEN GO TO 280
900 IF A{1}="0" THEN LET CIM=C
IM+15: GO TO 930
910 IF A{1}="1" THEN LET CIM=C
IM+255: GO TO 930
920 IF A{1}<>CHR# 32 THEN POKE
CIM,FN C(A{1})
930 PRINT TAB 16:FN H{PEEK CIM
}
940 LET CIM=CIM+1
950 GO TO 870
960 BRIGHT 1: FLASH 1
970 PRINT AT 20,12:"2-KBYTE": F
LASH 0: BRIGHT 0
980 LET H1=2
990 POKE 53248,0
1000 POKE 53249,136
1010 POKE 53250,2
1020 POKE 53251,232
1030 POKE 53252,0
1040 RETURN
1050 FLASH 1: BRIGHT 1
1060 PRINT AT 20,12:"4-KBYTE": F
LASH 0: BRIGHT 0
1070 LET H1=4
1080 POKE 53248,0
1090 POKE 53249,128
1100 POKE 53250,4
1110 POKE 53251,240
1120 POKE 53252,0
1130 RETURN
1140 REM ALAPHELYZET
1150 DATA 62,0,1,127,255,237,121
,63,255,237,121,62,178,1,95,25
5,237,121,201
1160 REM BEOLVASAS
1170 DATA 243,33,0,224,237,91,0
,208,58,2,208,254,2,204,70,208,20
5,76,208,119,35,19,58,3,208,188,
194,62,208,205,16,208,251,201
1180 REM OUTPUT (2-K)
1190 DATA 1,191,255,237,121,201
1200 REM INPUT OUTPUT
1210 DATA 1,63,255,237,89,1,95,2
55,237,81,1,159,255,237,120,201
1220 REM ELLENORZES
1230 DATA 243,58,2,208,254,2,204
,70,208,33,0,224,237,91,0,208,20
5,76,208,190,196,129,208,35,19,5

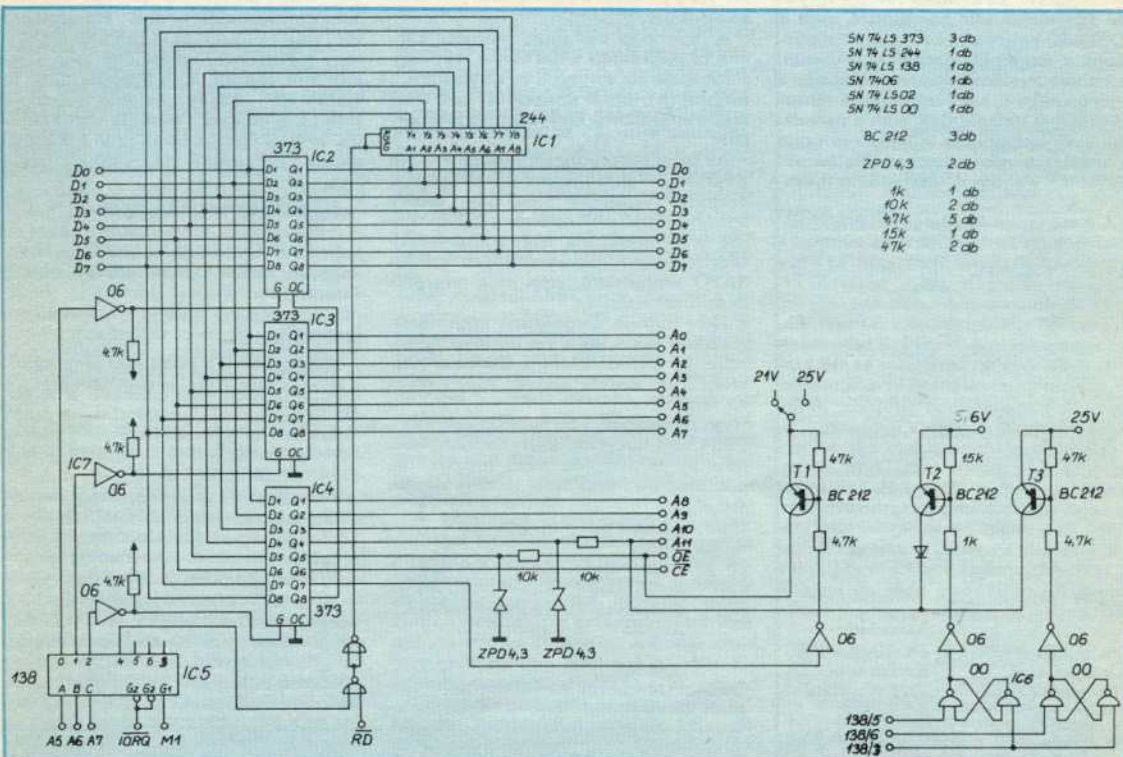
```

```

8, 3, 208, 188, 194, 108, 208, 205, 16, 2
08, 251, 201
1240 REM HIBA SZÁMLALASA
1250 DATA 58,4,208,60,4,208,2
01
1260 REM 2-K EGETO
1270 DATA 243,58,2,208,254,2,194
214,208,33,0,224,17,0,24,1,65,2
55,237,81,1,65,255,237,89,126,1
31,255,237,121,208,208,19,36
62,255,188,194,152,208,251,201
1280 REM EGETO RUTIN
1290 DATA 1,223,255,237,121,62,3
2,178,1,95,255,237,121,1,0,29,11
    
```

```

121,176,194,197,208,1,95,255,23
7,81,1,127,255,237,121,201
1300 REM 4-K EGETO
1310 DATA 33,0,224,17,0,48,1,95,1
255,237,81,1,65,255,237,89,126,1
31,255,237,121,208,249,208,19,3
5,62,240,188,194,220,208,251,201
1320 REM EGETO RUTIN
1330 DATA 1,22,246,64,1,95,255,23
7,121,230,237,237,121,845,1,0,29
11,121,176,194,9,209,241,246,32
01,95,255,237,121,230,191,237,1
01,201
9999 SAVE "HARAY" LINE 170
    
```



- SN 74 LS 373 3 db
- SN 74 LS 244 1 db
- SN 74 LS 138 1 db
- SN 7406 1 db
- SN 74 LS 02 1 db
- SN 74 LS 00 1 db
- BC 212 3 db
- ZPD 4,3 2 db
- 4k 1 db
- 10k 2 db
- 47k 3 db
- 15k 1 db
- 47k 2 db

5. ábra. Az EPROM-égető elvi kapcsolási rajza

"O" = 6B A25
9V = 4B A27

Az IC₁ olvasáskor az EPROM egy adott cím szerinti tartalmát az adatbuszra teszi. Az IC₂ íráskor az égetendő adatot az EPROM adatbuszára teszi. Az IC₃ feladata az alsó címbytek tárolása (A₀–A₇). Az IC₄ feladata a felső címbytek (A₈–A₁₁) az \overline{OE} , \overline{CE} , a 2732 égetőfeszültsége, valamint az IC₅ engedélyezése. Az IC₅ végzi a címzéseket. Az IC₆ biztosítja íráskor a 25 V, olvasáskor pedig az 5 V feszültséget a 2716 részére.

A Zener-diódák megakadályozzák az égetőfeszültségnek az IC₄ kimeneteire jutását. A T₁ kapcsolja a 2732 égetőfeszültségét, a T₂ kapcsolja a 2716 5 V-ját olvasáskor, a T₃ kapcsolja a 2716 25 V-ját égetéskor. A tápegység egy lehetséges megoldása, hogy a Spectrum 9 V-os tápjából transzverter segítségével állítjuk elő az égetőfeszültségeket. A vas-AL 100 fazékmag mérete 22 x 17 mm. A transzverter adatai: 2 x 20 menet 0,5 mm rézhuzal

2 x 8 menet 0,2 mm rézhuzal
1 x 80 menet 0,2 mm rézhuzal.
Az 5 V-os tápot átérésztővel, az 5,6 V-ot Zener-diódával állítjuk elő. A diódákat tűrésük miatt válogatni kell:
 $U_{VI} = 25 \pm 1 V$
 $U_{VII} = 21 \pm 1 V$
Hűteni csak a 7805-ös stabilizátort kell: erre a célra megfelel egy 60 x 40 x 1 mm-es alulemez. A béméréshez jól szolgál egy csövmérő és egy EMG 1555 vagy EMG 1567 oszcilloszkóp.

Integrált szoftver

Fájlok összerendelése

A cikksorozatunk előző részében közölt program az adatok vagy szövegfájlok egyesítését, összefűzését végzi, tehát a SZÖVEGSZERKESZTŐ és a POSTÁZÓ „termékeivel” dolgozik.

A POSTÁZÓ a címjegyzéket tartalmazó fájlt állítja elő. Ez azonban ebben a formában közvetlenül nem használható, ezért a POSTÁZÓ program <E> opciójával elkészítjük a megfelelő, kivonatolt változatot. A kivonat természetesen tartalmazhatja a teljes rekordot is, ha ez szükséges. A választás menüből történik, úgy, mint a nyomtatási opció esetében. Az új fájlban az egyes rekordok szövegfórmátumban tárolódnak; így például a kódmező nem jut át az új fájlba.

A menü utolsó kérdése: FÁJLNÉV? Válaszként azt a nevet adjuk meg, amelyen a kivonatolt fájlt tárolni akarjuk. Jó, ha a név emlékeztet az eredeti fájlra, amelyből készült. A program kivonatoláshoz egy új, még szabad területre tekéri a szalagot. Ehhez természetesen előbb be kell helyeznünk a magnetofonba egy kazettát és be kell kapcsolnunk a magnetofont. A kivonatolás folyamatát a képernyőn is láthatjuk. A fájl végén a program visszatér a főmenübe.

A kivonat készítésénél még egy dolgra felhívjuk a figyelmet. A POSTÁZÓ programmal a szöveges címjegyzék bevitelkor az egyes rekordok csupa nagybetűvel is írhatók. Ez azonban néhány nyomtatási képnél — például az levélnél — nem szerencsés. A POSTÁZÓ program <T> opciójával a szöveg visszaalakítható nagy- és kisbetűs formára:

DR. NAGY VAZUL Dr. Nagy Vazul
KARIKA U. 6. Karika u. 6.
7777 NAGYFALU 7777 Nagyfalu

használt /m vagy /ms jelölés csak egy sort vesz igénybe, szemben az összefűlésnél szükséges 4—7 sorral, ügyeljünk arra, hogy elég hely legyen a papíron!

A /m opció sort emel, beolvassa a címet, és kinyomtatja a cím három vagy négy sorát. Ha a program a címben üres sort talál, kihagyja azt.

A /ms opció sort emel, beolvassa a címet, és kinyomtatja a cím három vagy négy sorát, majd két sort emel és kinyomtatja a megszólítást. Ennél az opciónál nem lehet csak a megszólítást kinyomtatni a levél elejére!

Az összefűlésű program minden esetben a POSTÁZÓ által készített fájl kivonatával dolgozik, függetlenül attól, hogy esetleg minden címet felhasználunk a fájlból. Ennek az a magyarázata, hogy a levél megkívánt formátuma rendszerint elért a POSTÁZÓ programtól, ezért ez a megoldás látszott célszerűnek.

Nem történik automatikus megszólítás-generálás akkor, ha a /m opcióval dolgozunk. Ez esetben írhatunk általános megszólítást is: Kedves Vásárló! vagy Kedves Kolléga! stb.

A /ms opciónál a megszólítás-generálás a POSTÁZÓ program által használt első, második és harmadik mező (megszólítás, cím, rang stb., családnév, utónév) alapján alakul ki:

Címe,	Családnév,	Megszólítás
rangja	utónév	
Dr.	Nagy Vazul	Kedves Dr. Nagy
Prof.	Szabó Alajos	Kedves Prof. Szabó
	Szivárvány	Tisztelt Vásárlónk
	Áruház	

Láthatjuk, hogy egy kis ötletességgel változatos formátumokat állíthatunk elő.

A SZÖVEGSZERKESZTŐ-ben a kívánt levélfórmátumnak megfelelően a /sn (=soremelés n-szer) utasítás megelőzheti a /m vagy /ms utasításokat. Ezeknek az utólagos szerkesztéseknek az eredményét célszerű nyomtatásban is ellenőrizni, mert az összefűlésű programban szöveg- vagy formátummodosításokat nem végezhetünk; ezt csak a SZÖVEGSZERKESZTŐ-vel vagy a POSTÁZÓ-val tehetjük meg. Gyakori hiba például, hogy a címzés miatt az eredeti levél néhány sora átkerül a másik oldalra, és ezt csak a második vagy harmadik levélnél vesszük észre. Ezért a többpéldányos nyomtatásnál mindig ellenőrizzük a formátumot!

A munka menete a következők:

1. Készítjük el a szövegfájlt (például levelet) a SZÖVEGSZERKESZTŐ-n és /m vagy /ms jelölést írunk a cím és a megszólítás helyére.
2. Rögzítjük a szövegfájlt a kazettán a szokásos módon.
3. Készítjük el a POSTÁZÓ programmal a kívánt címjegyzék-kivonatot egy másik kazettán.

4. Töltjük be és futtassuk a POSTÁZÓ programot, adjuk meg a fájlneveket, ahogyan a program menüje kívánja.

Ne vegyük ki a címjegyzékfájl-kazettát addig, amíg az összefűlés tart! A címre-kordokat hármas blokkban olvassa be a program. Amint az összefűlés megtörtént, a programkazetta automatikusan leáll.

Ha a SZÖVEGSZERKESZTŐ-nél a /K opció választottuk (dátum, név és cím utólagos beírása nyomtatáskor), az összefűlésű program ezt figyelembe veszi, de ilyenkor a nyomtató mellett kell ülni és várni a gépeles kezdetét adó jelre. /K opciónál egyszerűen

?	[BH] (V) 3350		
A.....B.....C.....		
A	Havi kiadások		
B	Megnev.	összege	Százalék
C			
D	Lakás	1250.00	10.13
E	Közmű	845.00	6.85
F	Hitel	2400.00	19.46
G	Ruházat	1330.00	10.84
H	Élelem	> 3350.00	27.15
I	Közleik	1375.00	11.14
J	Egyéb	680.00	5.51
K	Takarék	1100.00	8.92
L	=====		
M	TOTAL	12338.00	100.00

Az átalakításnál minden mező első betűje nagybetű, a többi pozíció kisbetű lesz. A konverzió lassú folyamat, ezért csak a kivonat készítésénél (<E> opció) célszerű alkalmazni. Ahhoz, hogy pontos működését megismerjük, érdemes kipróbálni.

Mielőtt a szövegfájlok összefűzését elindítjuk, a SZÖVEGSZERKESZTŐ-vel elő kell állítani a szöveget, úgy, mintha normál szövegszerkesztő üzemmódban lennénk, csak a név és cím helyére a /m vagy /ms jelölést írjuk be. Az összefűlésű program a /m-et a POSTÁZÓ által készített kivonatolt név- és címfájlból található szöveggel helyettesíti. A /ms jelöléskor a név és cím mellett automatikusan generálódik a megszólítás is, például: Tisztelt Dr. Nagy! Mivel a SZÖVEGSZERKESZTŐ-ben

re csak egy címsor vagy mondat írható be, és ez azonnal nyomtatásban is megjelenik.

TÁBLÁZATKEZELŐ (előzetes)

Ezek után már csak egyetlen program maradt hátra: a TÁBLÁZATKEZELŐ, amely a legbonyolultabb, legösszetettebb ebben a sorozatban. Előjáróban tekintünk át feladatát és működését.

A táblázatkezelő programok elsősorban a könyvelésekben a pénzügyi mérlegek és az iparban az anyagmérlegek készítésénél használt nagyméretű adattáblázatok feldolgozását könnyítik meg. Ezek az előre nyomott táblázatok arra szolgálnak, hogy a számok (például mérési eredmények) az összefüggéseknek megfelelően szervezett formában kerüljenek feldolgozásra. A tartalom egyértelmű azonosítására a sorokat és oszlopokat címkével látjuk el. Egy-egy speciális célú táblázat létrehozása tulajdonképpen próbálkozásokon alapuló hosszú és fázadságos munka eredménye. A sorokat, oszlopokat sokszor kell újrainni, újraszámolni. Az utólagos igényeket elég nehéz figyelembe venni anélkül, hogy az egész táblázatot újra kelljen számolni, szerkeszteni.

A számítógépes táblázatkezelő programok megkönnyítik a táblázatkezelést; elvégzik a szükséges számításokat, nyomtatják az eredményeket, és az elkészült táblázatot későbbi felhasználás céljára eltárolják. Az ismertetendő programban megtalálhatók a „nagygepés” táblázatkezelők legfontosabb tulajdonságai, de természetesen a gép korlátaiból adódóan lényegesen kisebb méretekben.

A program ismertetése során néhány új fogalom is szóba kerül majd. A SOR és OSZLOP elnevezést sokan ismerik: az előbbi számok, kifejezések, címkek vízszintes elrendezése, az utóbbi pedig ennek függőleges megfelelője. A bemutatandó BASIC program 26 sor és 26 oszlop egyidejű kezelésére képes. Az így kialakuló háló vagy más néven rács 676 helyére lehet számot vagy címkét írni. Ezeket a helyeket a továbbiakban meznönek nevezzük. Minden sort és oszlopot az angol ábécé egy-egy betűje jelöl, A-tól Z-ig. Minden meznök van egy címe, amely az oszlopot és a sort jelölő betűből áll. Például a táblázat bal felső sarkában lévő mező címe AA lesz, a bal alsó sarkok AZ, a jobb felső sarkok ZA stb.: az első helyen álló betű az OSZLOP-ot, a második helyen álló a SOR-t jelöli ki.

A táblázatkezelő program Dragonon készült verziójával 3 oszlop és 13 sor jelenik meg egyszerre a képernyőn. A mindenkori ablakokat a bal oldalon és a táblázat tetején található címek, betűk jelölik ki.

A képernyő tetején két sor az ún. munkaterület, amelyen keresztül beírhatók az információk a táblázatba. A mezőkbe szám-

értékek, számítási formulák vagy címkek kerülhetnek.

Nézzük meg az ábrán látható példát. A számértékek (VALUES), amelyeket a képernyőn (V) betű jelöl, a számítógép gyártója által megadott értékeken belül bármelyek lehetnek. Jelen esetben 9 jegyre számolunk. Az értéket a kalkulátor üzemmódban közvetlenül be lehet írni, de egy számítási formula eredményeként kapott érték is kerülhet a megfelelő mezőbe.

A kifejezés (FORMULA) — a képernyőn (F) jelöli — számok és/vagy mezőnevek sorozata, amelyeket aritmetikai jelek választanak el. Például: +AA/AB*AC+200. Itt szigorúan érvényesül a balról jobbra precedenciaszabály, tehát: az AA mező tartalmát elosztja az AB mező tartalmával, az eredményt megszorozza az AC mező tartalmával és hozzáad 200-at. Zárójelke használható a jobb áttekinthetőség kedvéért, de ezeket a program a számolás során figyelmen kívül hagyja! A számítási kifejezés eredménye mindig egy szám lesz, amelyet a program a kijelölt mezőben megjelenít, kivéve ha azon a címen egy címke takarja az értéket.

Egy címkét (LABEL) — a képernyőn (L) — mindig üres helyet kell kezdeni; a program erről ismeri fel, hogy címkét kezdünk beírni. Ha egy szám elé szóközt teszünk, a szám címkeként fog szerepelni. Miután a program a címkét eltárolta, eltávolítja a szóközt. Minden oszlop 10 karakter széles, de egy címke „átszordulhat” a szomszédos mezőbe is. A címke egyszerűen eltávolítható: mindössze egy szóközt kell beírni a megfelelő pozícióra.

A képernyőn megjelenítéskor főlegreldeliséget tapasztalunk: az azonos mezőre megadott címke mindig fedi az értéket. Ezáltal munkaterületeket tehetünk el például aláhúzások, magyarázó szövegek mögé. Nulla számértéket a program nem jelenít meg, kivéve, ha a nulla egy számítás eredményeként került a cellába. A számítási kifejezések nem jelennek meg a táblázatban.

Mint már említettük, a képernyő felső két sora munkaterület, amelyen keresztül információkat vihetünk be a táblázatba. Az első sor adatbevitel kérésére utaló jellel (?) és a feladat kijelölését tartalmazza. A ! jel például az újraszámolást jelenti, a : pedig a nyomtatást a képernyőre.

A második sorban a kurzorpozícióban található mező legfontosabb információi írtnak ki: ez rendszerint egy kifejezés vagy számérték, de lehet címke is.

A valóságban a táblázat három szint mélységű. A számértékek, a kifejezések és a címkek egymástól elválasztva tárolódnak, ami által a program meglehetősen rugalmasá válik.

Legközelebb részletesen ismertetjük a programot és használatát, majd néhány példával szemlélítjük a működését.

GALINA FERENC

A tartalomleírások az alábbi folyóiratokban megjelent programlistákról készültek:

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

A tartalomleíró szövegeket permutáljuk, a szölcvariansokat pedig alfabetikusan rendeztük.

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

A forráshely a karaktorsorozatát nyílvézi be, melyet a / jelleg a folyóiratok azonosítója, a két / jel között az évszám, folyóirat szám 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 fel 5-ig. Tel.: 855-111/251) 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: _____ Pédányszám: _____
Kód: _____ Pédányszám: _____
Kód: _____ Pédányszám: _____

A megrendeléshez csatolom az oldalankénti B₁- Ft-os szolgáltatási díj befizetését igazoló csekszelvénynt.
Dátum, név, pontos cím.

```
PROGRAMLISTA
anímico@commodore 120@grafica@magi
c lantern' -'comi'06.03.104 0
PROGRAMLISTA
serie lic@parameter beallitas a sor
a illesztés -hc 06.10.77 4
PROGRAMLISTA
assembler programozas@cifrisorozat@co
mmodore 04@later@program forrasiz
-napp'06.11-70 4
PROGRAMLISTA
assembler programozas@cifrisorozat@co
mmodore 04@later@ateres@basicrol
-64er'06.10.152 5
```

INFORM

PROGRAMLISTA
starti 800 -> programlevedes listaves
et modositas automatikus torleshez
->happ '86.10.104/1

PROGRAMLISTA
starti xl file256-karakteres256-karak
teres futozosveg generalasa
->happ '86.10-04/1

PROGRAMLISTA
starti xl filegrafika'dis-show'
->happ '86.11-65/2

PROGRAMLISTA
barKacolas Commodore 64 fotozas mer
ez technika kameran szereplo epozico
c idop ellenorzes
->64er '86.12-46/2

PROGRAMLISTA
barKacolas Commodore 64 illesztore
ntronics nyomatatas Keszites utmuta
to ->64er '86.12-57/8

PROGRAMLISTA
cikksorozat Commodore 128 idc regisz
terek Kezelese ->64er '86.11-00/3

PROGRAMLISTA
cikksorozat Commodore 64 jatekprog
ram Keszites 'moon base' mint példapro
gram ->comi '86.09-84/4

PROGRAMLISTA
cikksorozat Commodore 64 128 grafika
matriszmitas transformacioi
->64er '86.12-146/5

PROGRAMLISTA
Commodore 128 cp/m formatumtervez
es
->64er '86.10-90/2

PROGRAMLISTA
Commodore 128 jatekprogram 'sprite o
uzle' ->comi '86.09-112/3

PROGRAMLISTA
Commodore 16:64 plus 'draustick' on
alrajzolas botkormannyal
->run '86.11-90/2

PROGRAMLISTA
Commodore 64 'joystick mas'
->hc '86.10-44/1

PROGRAMLISTA
Commodore 64 autostart rutin a 1541
hez ->happ '86.11-78/3

PROGRAMLISTA
Commodore 64 basic precompiler
->64er '86.12-69/5

PROGRAMLISTA
Commodore 64 basic program valtozo
v Bevezetesi lista
->run '86.12-106/3

PROGRAMLISTA
Commodore 64 helso ora assembler pro
gramozasa ->happ '86.10-59/4

PROGRAMLISTA
Commodore 64 giga cad grafika nyomt
atas ->64er '86.12-97/2

PROGRAMLISTA
Commodore 64 giga cad grafika rajze
p (150) vezelise ->64er '86.11-74/2

PROGRAMLISTA
Commodore 64 gorgeto szubrutinok
->run '86.10-11/1

PROGRAMLISTA
Commodore 64 grafika '64 grafic' hir
es gepi fodto utasitasi
->run '86.10-68/4

PROGRAMLISTA
Commodore 64 grafika monitorkep kin
omtatasa ->run '86.11-69/2

PROGRAMLISTA
Commodore 64 grafika sprite Kezela
s atkapolthato 30/40 oszlopas abzolas
->64er '86.12-91/2

PROGRAMLISTA
Commodore 64 grafika tengelyforuli o
rgasztas ->64er '86.11-54/7

PROGRAMLISTA
Commodore 64 'o gorsito' lemezegys
e

g (1541) muveletgorzitas basic-rom a
latti ram hasznalata
->64er '86.12-54/3

PROGRAMLISTA
Commodore 64 jatekprogram cave raid
3 ->happ '86.10-63/3

PROGRAMLISTA
Commodore 64 jatekprogram orokelet p
oke keresesi modszerek
->happ '86.11-155/4

PROGRAMLISTA
Commodore 64 mutato generalas menuho
z pozicionalas botkormannyal
->run '86.10-78/2

PROGRAMLISTA
Commodore 64 nyomatatas mes001 ha
rom szinu masolat indigopapirral
->64er '86.10-105/4

PROGRAMLISTA
Commodore 64 nyomatatas mes001 ke
rmet csokkentese ->64er '86.10-95/1

PROGRAMLISTA
Commodore 64 oktatas az amitas tech
nika logikai operatorok Kezelese
->run '86.11-90/2

PROGRAMLISTA
Commodore 64 prodisc kiegezesites a
(86.06) ban kozotti listahoz
->64er '86.11-89/1

PROGRAMLISTA
Commodore 64 programozasi tippek ha
ladoknar ->64er '86.10-01/4

PROGRAMLISTA
Commodore 64 programozasi tippek ha
ladoknar ->64er '86.11-76/3

PROGRAMLISTA
Commodore 64 programtorok nyelvi k
atalogizalasi 108 byte-os parancs
utin ->happ '86.10-72/1

PROGRAMLISTA
Commodore 64 tukorkeppajzolas
->run '86.10-86/2

PROGRAMLISTA
Commodore 64 128 eger muvodesi el
ncsopoltasi rajz ->64er '86.11-44/4

PROGRAMLISTA
Commodore 64 128 nyomatatas 'epson' ke
pment csokkentese ->64er '86.10-94/1

PROGRAMLISTA
Commodore 64 128 oktatas (morse-abc)
jelkonvertalasi ado/vevo uzemmodhoz
->run '86.11-74/4

PROGRAMLISTA
Commodore 64 128 jatekprogram oktat
as (matematika) 'math scramble'
->run '86.11-82/3

PROGRAMLISTA
Commodore 64 128 programmoduloz oss
zekapcsolasa ->run '86.10-88/3

PROGRAMLISTA
Commodore 64 128 visszamentes scrat
ch utan ->comi '86.09-82/2

PROGRAMLISTA
helps bites (86.08) 76
->hc '86.10-50/1

PROGRAMLISTA
helps bites (86.09) datazette-egedp
rogramform-killer ->happ '86.11-130/1

PROGRAMLISTA
sakhatari xl Commodore 64 ket sz
amitogezes jatek ->happ '86.11-61/3

PROGRAMLISTA
sincclair spectrum decimalis-lebegpo
ntos atalkitas (automatikus)
->happ '86.11-70/1

PROGRAMLISTA
sincclair spectrum eger szimulacio bot
kormannyal ->happ '86.10-86/2

PROGRAMLISTA
sincclair spectrum gorgeto rutin
->inc '86.12-125/1

PROGRAMLISTA
sincclair spectrum grafika
->hc '86.10-46/4

PROGRAMLISTA
sincclair spectrum jatekprogram Kesz
ites high-score lista kiiratas
->hc '86.10-45/2

PROGRAMLISTA
sincclair spectrum jatekprogram 'cave
hero' ->inc '86.11-106/2

PROGRAMLISTA
sincclair spectrum jatekprogram 'back
to font' ->inc '86.12-129/2

PROGRAMLISTA
sincclair spectrum jatekprogram 'bogg
le' ->inc '86.11-105/2

PROGRAMLISTA
sincclair spectrum jatekprogram 'fore
st of the long shadows'
->inc '86.10-100/4

PROGRAMLISTA
sincclair spectrum jatekprogram 'mou
se all' ->inc '86.11-100/1

PROGRAMLISTA
sincclair spectrum jatekprogram 'pro
ced word' ->inc '86.12-132/1

PROGRAMLISTA
sincclair spectrum jatekprogram 'pos
t-district' ->inc '86.12-129/1

PROGRAMLISTA
sincclair spectrum jatekprogram 'tree
builder' ->inc '86.12-130/2

PROGRAMLISTA
sincclair spectrum tasuord gyors alf
betikus rendezes ->happ '86.10-88/1

PROGRAMLISTA
sincclair spectrum tasuord pascal pro
gramozasi file-formatum atalkitas
->happ '86.11-68/1

PROGRAMLISTA
spor Commodore 64 'bundesliga-manag
er' ->hc '86.10-35/3

PROGRAMLISTA
startexter szovegfeldolgozas Commod
ore 64 monitorkep formalas es lemeze
iras ->64er '86.11-172/1

PROGRAMLISTA
szerviz barKacolas cikksorozat leme
zseg (1541) hibakereses fejezocico
nalas ->64er '86.12-48/3

PROGRAMLISTA
szovegfeldolgozas Commodore 128 (run
script 128) ->run '86.12-61/26

PROGRAMLISTA
szovegfeldolgozas Commodore 64 maste
r text formatum atalkito idegen fil
ekhoz ->64er '86.11-96/2

PROGRAMLISTA
szovegfeldolgozas Commodore 64 nyomt
atas Ascii konverzio karaktereszlet
modositasi ->64er '86.10-28/4

PROGRAMLISTA
szovegfeldolgozas vizarute cikksoro
zat Commodore 64 helyesiras ellenor
zes ->64er '86.11-65/6

PROGRAMLISTA
terminal uzemmod Commodore 128 (mini
term 128) ->comi '86.09-110/2

PROGRAMLISTA
vizaruter cikksorozat Commodore 64
giga cad grafika szoveg-kep szerkes
tes ->64er '86.10-179/6

PROGRAMLISTA
zene barKacolas Commodore 64 hangd
igitalizalo Keszites utmutato
->64er '86.10-65/7

PROGRAMLISTA
zene Commodore 64 'sound monitor' al
kalmazasi utmutato
->64er '86.10-53/3

PROGRAMLISTA
zene Commodore 64 hyper-basic id-pr
ogramozasi poke nalkul
->64er '86.11-85/2

Az Olivetti nem hibázik! Jobb kettő, mint egy sem



Az olaszországi Olivetti cég is magáévá tette e szólás alapigazsá-
gát, és megkezdte egy új gépcsalád gyártását. A katonai, űrhajózá-
si, pénzügyi stb. rendszerekben alapvető követelmény a minél na-
gyobb biztonság és a folyamatos űzem. Az új gépcsalád, amely
ezen elv szerint működik, az OLIVETTI CPS 32 nevet viseli.
A CPS a Continuous Processing System (folyamatos feldolgozási
rendszer) rövidítése.

A rendszer alapkonceptója nem új: duplázzunk meg minden
hardveregységet, így egy rendszer helyett kettő fog egyszerre dol-
gozni. Ha valamelyik egységben hiba jelentkezik, a másik folytatni
tudja a munkát. Az elképzelés azonban mindeddig mégsem valós-
ult meg, mert a megduplázott áramkörökkel működő gépek igen
költségesek lettek volna. Mégis, az elmúlt tíz évben jelentős válto-
zás történt ezen a téren.

A hardveremelés olcsósága, méretük és energiaigényük csökke-
nése lehetővé tette az elgondolás megvalósítását. A szoftverköltse-
gek emelkedése viszont a különálló gépek összekapcsolása ellen
szólt. Korábban próbálkoztak ugyanis különálló, hagyományos
nagygepek „párhuzamosításával”, ami az éppen futó programnak
megfelelő ellenőrzőprogramot is igényelt.

A CPS tehát egy gépen belül duplázza meg az áramköri eleme-
ket, és az ellenőrzést is saját beépített monitorprogramjával végzi.
Használtsuk össze a hagyományos rendszereket a CPS-sel! Ha
egy hagyományos rendszerben valamilyen hiba keletkezett, azt
nem mindig lehetett azonnal észlelni. A feldolgozás tovább folyta-
tódott, gyakran hibás eredményeket szolgáltatva.

A CPS 32 rendszer minden egysége tartalmaz egy hardvermoni-
tor, amely a feldolgozott adatok folyamatos ellenőrzését végzi. Az
önellenőrzés másodpercenként nyolcmilliószor megy végbe, így az
esetleges hibát a rendszer gyakorlatilag keletkezésének pillanatá-
ban észleli. Ez igen jelentős előnyökkel jár:

- a feldolgozott adatok garantáltan hibamentesek,
 - nincs szükség hosszadalmas adathiba-feltárára és újrafel-
dolgozásra,
 - elhagyható a bonyolult áramköri hibakeresés, mivel a rend-
szer jelzi, hogy hol történt a meghibásodás,
 - a hibajavítás időtartamára nem kell a rendszert leállítani.
- A CPS 32 család az FT 200, az XA 400 és az XA 600 típusú gé-
pekből áll. Minden gép virtuális operációs rendszer (Virtual Oper-
ating System) alatt dolgozik.

Mivel az elemek duplázták, ugyanazon a feladaton egyszerre
két rendszer dolgozik. Ha egy alkatrész meghibásodik, a párja a
felhasználó zavarása nélkül folytatja munkáját. A feldolgozás se-
bessége ekkor sem csökken.

Az előfordult hibáról külön jelzést ad. A hibás egységet a rajta
kigyulladó piros színű világítódióda jelzi. A hibás egység a kigyul-
lás után, annak megszakítása nélkül kicserélhető. Az új egység
behelyezését észleli a rendszer, az egységet automatikusan ellenőr-
zi, majd újra beállítja működési feltételeit, és a feldolgozás már is-
mét a dupla rendszeren fut anélkül, hogy a felhasználó ezalatt egy
másodperc kiesést is tapasztalt volna.

ZSOM BÉLA

ADOK-VESZEK-CSERÉLEK

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

ADOK

Angol nyelvvizsga-előkészítő szótári program C64-re. 11 témakörben 1400 szót tanít, kiejtéssel. 2-3-szor gyorsabban tanulhat, mint szótárból. Ára magánzemélyeknek és közületeknek floppyn 3600 forint. Kivánságra részletes tájékoztatót küldök. Ióth András számítástechnika, Post Restant, 1364 Budapest 4.

C64-re angol nyelvet oktató programot - összesen öt lemezzel rendelni anyagot - eladóék vagy elcserélnek. Egy oldal ára 100 Ft + a lemez ára (amennyiben szükséges). Eladó továbbá a Flight Deck nevű játékgépprogram, ára 200 Ft. Illés István, Cikó, Parczel M. u. 44. 1161

C64 programkettőn eladóék. Kazettánként 50-60 program. Egy kazetta 500 Ft (+ utavétel). Ióth József, Hajdúbobos-zsornyi, Terekeskova u. 22. 4220

C64 magndal, joystickkal, programkettővel és bő szakirodalommal eladó. Irányár: 28 000.- Ft. Spivák Zoltán, Nyireghyza, Május 1. tér 11. 4400

Commodore 1901-es színes monitor (új) PAL és RGB csatlákozával olcsón eladó. Tokaji Zoltán, Budapest, Petőfi S. u. 5. 1052 vagy: 18-77-48 telefonúzenet.

Commodore 64 programok titkosítását floppyn vállalom magánzemélyeknek és közületeknek. Minden másoldal szomban biztos védelem, igen nehéz visszafejtethe-
tőség. Levélben részletesen tájékoztatom. Bálldi András, Post Restant, 1364 Budapest 4.

Programok C Plus/4 és C16 szériájú gépekre: FORTH nyelv, Turbo file kezelés fejlesztés, Disk system BASIC és assem-
bler fejlesztőprogram, Copy 226 lemez-es/szalagos másoló, C64-C Plus/4 turbo, C64-C Plus/4 rutin és rendszerértelmező összehasonlító táblázat, Profians C Plus/4 kétfenestű assemblert fordító. A programok ára 180-600 forint, kazettán, utavétellel, részletes kezelési utatitással. Kérésre ismertetőt küldök. Pelsőczy Gyula, Szilasliget, Ady Endre u. 36. 2145

C64 és C128-tulajdonosok figyelme! János-
sity televíziót videomonitorra alakít-
tok (képhang), az eredeti funkciók megtar-
tása mellett. C128 esetén 80 csoplosz megjelölést is alkalmas. Páll Miklós,
Budapest, 11. Rákóczi Ferenc u. 345. k.
7-10 C lépcső, 1214

Sony személyi számítógép (MSX rendze-
rű) - 280A, 64 k RAM, 448 k ROM,
256x192-es felbontású grafika, 3 hang-
szatona, személyi adatbank -, 2 db
joystickkal, cartridge-zsel, dokumenta-
cióval eladó. Tel.: 26/49-051

Vidocron TV-Computer olcsón eladó.
Tel.: 647-887

X861 (64 kbjt) sok programmal és joystickkal eladó. Árjajánlatot levélben kérek. Vigh György, Budapest, Margit u. B. 1165

ZX-Spectrum (48 k) olcsón eladó. Vadon Zoltán, Budapest, Torcsavár u. 28. 1112. Tel.: 669-539

VESZEK

Vennék üzempételem VC/VIC 20, HI 1080Z, ABC 80, Prime, ZX80, ZX81, TIMEX 1000, TIMEX 2068, C16, AIRCOMP 16 számí-
tógépet és tartozékait. (Darabok is érdekelnek.) Árjajánlatokat a következő címre kérem: Maróti Gyula, Bördányi, Zákányzéki u. 26. 6795

CSERÉLEK

C16 és C Plus/4 programokat cseréllek. Surányi Péter, Miskolc, Körösi Cs. u. 7. 3527

C16 és C Plus/4 programokat cseréllek. Gorenóser Tamás, Zalaegerszeg, Lander-
ngy ut 18 B. III/9. 8900

C64-re készült programokat cseréllek (elsősorban kazettán, de lemezen is). Hajdu László, Budapest, Szigetzi József u. 17. V/32. 1041

C64 programokat kazettán cseréllek. Bodai József, Pápa, Fő tér 14. 8500

Commodore 64-es számítógépre játé-
k-programokat cseréllek kazettán. Válasz-
okat listával a következő címre: Banyák Mihály, Mosonmagyaróvár, Széchenyi u. 19. II/8. 9200

ZX-Spectrum felhasználói és színvonal-
s játékgépprogramokat cseréllek. Szarka
Endre, Veszprém, Stadion u. 22. 8200

Milyen feladatok oldhatók meg számítógéppel?

Képes-e egy számítógép például árkot ásní, zoknit mosni vagy egy nehéz ládat felcipelni az első emeletről a harmadikra? Általában úgy tartjuk: nem, de a válasz attól függ, hogyan definiáljuk a számítógépet, hol húzzuk meg a határt a számítógép és a környezet között.

Másfelől: miért nem nevezünk egy egyszerű zsebkalkulátort számítógépnek, csak (megkülönböztetésül) számológépnek? Általánosabban: mi az a *minimális tudás*, mellyel egy eszköznek rendelkezni kell ahhoz, hogy a számítógép nevet kiérdemlje?

A számítógép meghatározásával kapcsolatos kérdések láthatóan két csoportra oszthatók: az egyik csoport *felülről*, a másik *alulról* próbálja behatárolni a gépet.

Meghatározásaink, fogalmaink, elnevezéseink

Gyakran árnakdesnek tűnhetnek — mögöttük azonban rendszertírelmely társadalmi *tapasztalat* hűződik meg. Andor Csaba könyvének (Jel-Kultúra-Kommunikáció, Gondolat, 1980.) példáját idézve: megtehetjük-e, „hogy a lovak egy részét — mondjuk a kancákat — a zsiráfok osztályával egyesítve, ezentúl ezt az osztályt jelöljük egy meghatározott szóval? A választ nem is olyan könnyű megadni, mert egyfelől valóban nincs *elvi* akadályja egy ilyen — meglehetősen „bizarr” — osztályozásnak, másrészt viszont semmiképpen sem véletlen, hogy egyetlen olyan nyelvet sem ismerünk, amelyben külön szó jelölne az állatoknak ezt a keverék osztályt. ... Ennek az az oka, hogy a nyelvhasználat során a *ló-val* jelölt osztályra gyakran, míg a „kanca vagy zsiráf” osztályra úgyszólván sohasem történik utalás, így célszerűbb az előbbi osztályt egyetlen jellel, míg az utóbbit szükség esetén valamilyen nyelvi leírással helyettesíteni. Persze az sem véletlen, hogy az egyik osztályt lényegesen gyakrabban kell megnevezni, mint a másikat: végső soron a *társadalmi gyakorlat* határozza meg, hogy az egyes természetes nyelvek milyen megkülönböztetési elvek szerint osztják fel a világot.

Ha tehát a „kanca vagy zsiráf” osztályra egyetlen nyelvben sincs külön szó, akkor ezt úgy értelmezhetjük, hogy ez az osztály egyetlen közösség életében sem játszik fontos szerepet.”

A számítógép meghatározása mögött is nyilván az hűződik meg, hogy a közösség mit tekint a számítógépek lényegének.”

A minimális tudás

A számítógépektől megkövetelt *minimális tudás* problémája érdekes módon a számítógépek *előtti* időkre nyúlik vissza, és összefügg azzal a filozófusok és matematikusok által feszegetett problémával, hogy mi a

valóság, a bizonyíthatóság és a megoldhatóság

viszonya. A történelmi vonulatot többek között olyan nagy nevek fémjelzik, mint Arisztotelész (i.e. 384-322), Hilbert (1862-1943), Gödel (1906-1978), Turing (1912-1954) — mint ahogy ezt egy 1986-ban készített és Magyarországon meg a nagyközönség elé nem került videofilm is összefoglalja (Dr. Kurt Gödel: Ein Mathematischer Mytos. P. Weibel és W. Schimanovich, 1986.) Reméljük, hogy ezt a filmet valamikor a szélesebb hazai közönség is láthatja majd.

A brünni születésű osztrák Gödel volt az, aki 1930-ban többek között bebizonyította, hogy egyetlen formális rendszer sem lehet tökéletesen zárt; a valóság erősebb, mint a *bebizonyíthatóság*. Vannak olyan feladatok, melyek egy formális rendszeren belül bizonyíthatóan *megoldhatatlanok*.

Számunkra különösen érdekes például az, hogy egy programmal nem lehet kideríteni, hogy egy másik programban vannak-e végtelen ciklusok.

Az angol Turingot is a megoldhatóság problémái foglalkoztatták, amikor 1936-ban kigondolt egy igen egyszerű szerkezetet

— egy számítógépet — (lásd az 1. ábrát), amelyről be lehetett bizonyítani (a bizonyítás jóval meghaladja e cikk kereteit), hogy vele *minden* más, *szimbólumokkal műveleteket végző gép működése elvileg tökéletesen szimulálható*. Azaz ez az egyszerű gép (megfelelően programozva) képes — a sebességtől eltekintve — *tökéletesen* úgy viselkedni, mint bármilyen más szimbólumfeldolgozó gép. (Megjegyezzük, hogy cikkünkben nem térünk ki arra, hogyan lehetett egy ilyen egyszerű felépítésű géppel [majdnem] minden, szimbólumokra visszavezethető feladatot megoldani. Az ábrával csak azt kívánjuk érzékeltetni, hogy milyen egyszerű [?] gépről van szó.) Bizonyítható továbbá, hogy az a feladat, mely ezzel a Turing-féle géppel elvileg nem oldható meg, az *semmilyen* más számítógéppel sem lesz megoldható.

Amikor azt mondjuk, hogy a számítógépek közös, lényegi vonása az, hogy (megfelelően programozva őket) minden olyan feladat megoldására képesek, melyekre a Turing-féle gép képes, akkor meghatároztuk azt a *minimumot*, mely ahhoz kell, hogy egy szerkezetnek a számítógép nevet adhassuk.

Természetesen sok olyan szimbólumfeldolgozó gép van, mely *nem* képes minden olyan feladatot megoldani, melyet a Turing-féle gép *elvileg* képes lenne. Ilyenek lehetnek például az automata háztartási gépekbe, egyes szerszámgépekbe szerelt elektronikus vezérlő, számítógépszerű célszerkezetek, de ilyenek például a már korábban említett zsebkalkulátorok is.

Szimbólumok, mintázatok, tükröztetés

Láttuk, hogy az elvárható minimális „tudást” illetően a számítógépek általános célú *szimbólumfeldolgozó* eszközöként definiálhatók. A *szimbólum* (a jelkép) szót a köznyelv is használja: olyan ábrát, tárgyat stb.-t szoktunk így megnevezni, mely valami más (fogalom, dolog, személy stb.) helyett áll, valami mást „ábrázol”. A mindennapi életben a szimbólumot és azt, amit ábrázol, összekapcsolhatja az emberi képzelet (a gondolatátvitál), de összekapcsolhatja vagy a társítást megerősítheti pontos leírás, hagyomány, akár jogszabály is (lásd például az állami jelvényekre vonatkozókat).¹⁾

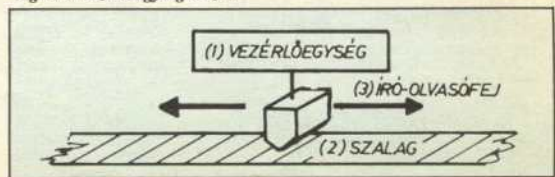
Számítógépekkel dolgozva a megoldandó feladatok világát *leképezzük* a gép által kezelhető fizikai *mintázatok* (a gép alkotóelemei állapotában megtestesülő) világára. A *fizikai* jelzővel itt arra utalunk, hogy ma a gépek alkotóelemei — a feladatok világát tükröző állapotaikat-állapotváltozásait tekintve — a *fizika* (konkrétan: az elektronika, a villamosságban és nem például a kémia vagy a biológia) törvényei szerint működnek.

A gép képes a *mintázatok*at összehasonlítani, illetve képes azokkal más műveleteket is végezni. A gép által kezelhető szimbólumok készletét (a gép ábcéjét) éppen az határozza meg, hogy a gép ezt az alapvető összehasonlítást milyen mintázatokon képes elvégezni. Ezt a tudást persze a gép tervezője építette be a gépbe.

A feladatok világának *tükröztetése* a szimbólumok világára nem új dolog azoknak sem, akik számítógépet még sohasem láttak — legfeljebb nem tudatosult bennük a művelet ezen aspektusa, akárcsak az „Úrhatnám polgár” Jourdain urában az, hogy egész életében „prózában beszélt”. Például amikor az iskolai feladatainkat „lefordítottuk” a matematikai egyenletek nyelvére, és ezeket az

1. ábra. A Turing-féle gép

- (1) A vezérlőegység véges számú állapot felvételére képes
- (2) A szalag diszkrét részekre (négyzetögre) van osztva, melyek mindegyike egy véges készletből vett, egyetlen szimbólum tárolására képes
- (3) Az író-olvasó fej a szalag menetén lépésként (négyzetenként) képes előre/hátra mozogni és információt továbbítani (oda/vissza) a szalag és a vezérlőegység között



is és eltűnődünk azon, hogy lényegét tekintve mit is nevezünk számítógépnek. Fontos ez? Megyünk-e ezzel valamire? Szoktunk-e gondolkodni például azon, hogy mi egy nadrág lényege? Felvesszük, levesszük, kimossuk, eldobjuk, satöbbi és kész. Joggal furcsállnánk, ha valaki elkezdene: a nadrág olyan háromlyukú ruhadarab, melyen az egyik lyuk nagyobb és ellenkező irányú, mint a másik kettő... , satöbbi. A számítógépeket is így vagyunk: bemegyünk a boltba, vásárolunk egyet, majd bekapcsoljuk, használjuk definíció nélkül is. A kérdésnek nincs közvetlen gyakorlati jelentősége, inkább csak ürügy arra, hogy a számítógép-fogalmat körbejárjuk.

egyenleteket leírtuk a fűzetünkbe, szintén ilyesmit tettünk. Amikor például leírjuk azt a szót egy levélben, hogy „kutya” és közben egy létező, élő, konkrét kutyára gondolunk, akkor a valóságos kutyát az ábécé betűiből összeállított karakterorsóra tükröztettük.

Hol a „felső határ”?

Szokás a „turingi” (a lényegi, a szükséges minimumot tartalmazó) részsel elektromosan összekötött szerkezetek együttesét konfigurációnak nevezni, a teljes konfigurációt pedig a hagyományos és a ma tipikus gépeknél szokás (és lehet!) ún. központi egységre (CPU-ra) és ún. perifériális egységekre (röviden csak: perifériákra) osztani. A perifériák tipikus szimbólumok be- és kivitelére, átvitelére vagy ún. háttérátrolásra szolgálnak (lásd a 2. ábrát). De hol a CPU határa, hol kezdődnek a perifériák? A számítógépek (az ún. konfiguráció) felosztása központi egységre és perifériákra egyre nehezebb lesz.

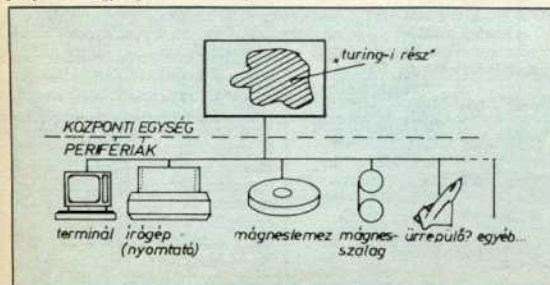
A CPU-hoz kapcsolódó eszközök halmaza elvileg korlátlan. Ténylehető-e például egy beépített számítógéppel vezérelt ürrepülőgép a számítógép különleges perifériájának? Elvileg nem lenne akadály a egy ilyen felfogásnak sem. Gyakorlati érzékünk persze tiltakozna ez ellen.

Ha a számítógépekhez a velük elvileg megoldható feladatok felől közelítünk és a számítógép-fogalomba beleértjük a perifériákat is, akkor nem jutunk messzire a lényeg megfogásában, mert (legalábbis elvben) nem fogunk találni olyan (megoldható!) feladatot, mely az ilyen tágan értelmezett géppel nem lenne megoldható. Ilyen értelmezéssel ugyanúgy nem jutnánk közelebb a lényeghez, mint ha a vasút lényegét az alapján próbálnánk megragadni, hogy „ki mindenki utazhat vasúton” (cipész, postás, polgármester, tőkmagárus).

Visszaautalva a „kanca-zsiráf” példára, a számítógép lényegét kutatva akkor járunk el célszerűen, ha azt a turingi, a minimálisan szükséges részt is tartalmazó CPU környékén keressük.

A gyakorlatban általában abból indulunk ki, hogy mindaz, ami a lényegi (a „turingi”) részsel egy mechanikai egységet (egy szerelvényt) alkot, azt nevezik központi egységnek. A központi egységek az elméleti minimumnál általában persze sokkal többet tartalmaznak, mert más dolog kigondolni egy gépet egy fontos tétel bebizonyításához, ahogy ezt Turing is tette (ez legyen lehetőleg minél egyszerűbb, az se baj, ha nagyon lassú, hiszen úgysem építik meg, csak gondolat kísérleteket végeznek vele), és megint más olyan gépeket tervezni, amelyekkel feladatok sokaságát kell majd gyakorla-

2. ábra. A számítógépeket hagyományosan ún. központi egységre és ún. perifériális egységekre szokták felosztani



ti követelményeket is kielégítő módon (gyorsan, olcsón stb.) megoldani.

A „felső határ” (azt, hogy mit ne értsünk a lényegi részbe a számítógép környezetéből) keresve meg kell elégednünk tehát azzal, hogy gyakorlati szempontok alapján leválasztjuk a CPU-ról a perifériák tarka sokaságát, tudva, hogy a CPU-fogalom is képlekeny és az is elképzelhető, hogy hosszabb távon értelmét is veszti.

Mire jutottunk?

Nem nagyon sokra. Megállapítottuk, hogy a számítógépek lényegüket tekintve általános célú szimbólumfeldolgozó szerkezetek, melyekbe legalább a „turingi tudás” be van építve. Azt, hogy ez pontosan micsoda, itt nem mondtuk meg, csak egy ábrával utaltunk a Turing-féle gépre. Szó volt arról, hogy ma a számítógépeket — az ún. konfigurációkat — CPU-ra és perifériákra szokás felosztani. Tudjuk, hogy az elhatárolás esetleges, önkényes. Csak önkényesen tudtuk elhatárolni a gépeket a külvilágtól, a környezetüktől is.

Röviden körüljártuk a szimbólumok, a mintázatok és a tükröztes fogalmait, és visszahúzódtunk a szimbólumfeldolgozás világába.

Kikerültük azt a kényes kérdést, hogy a gép—szimbólum—művelet „hármast” csak együtt értelmezhető igazán, bár aki figyelmes volt, az felfigyelhetett arra, hogy például a szimbólumok készletét éppen az határozza meg, hogy a gép milyen mintázatok megkülönböztetésére képes az összehasonlítási művelet kapcsán.

Ami az „általánosculóság” illeti, ez nem jelenti azt, hogy a számítógéppel minden olyan feladat megoldható, mely szimbólumokkal végezhető műveletre vezethető vissza. Vannak ugyanis megoldhatatlan feladatok. Beleütköztünk a „gödeli problémába”. Azt, hogy mely feladatok nem megoldhatók, azt éppen Turing-féle gépek segítségével vizsgálják.

Ezek után miben maradjunk? Először is abban, amiből kiindultunk: a kérdésnek nincs közvetlen, gyakorlati jelentősége. A mindennapi életben nem okoz gondot a pontos definíció hiánya. A számítógépeinkkel naponta megoldandó feladatok sokasága pedig megoldható! A megoldatlan (vagy ma annak látszó) problémák elsősorban a filozófusok, a matematikusok számára érdekesek. Az átlagfelhasználó ezekkel nem találkozik. Egyébként is megtanultunk olyan világban élni, tájékozódni, melyben — anélkül, hogy tudtuk volna vagy számunkra bebizonyították volna — homályosan éreztük (vagy nem történőtt vele), hogy fogalmaink nem alkotnak tökéletesen zárt rendszert. Talán ösztöneinkben érezzük, amit Gödel már 1930-ban bebizonyított, hogy a „valóság erősebb a bizonyíthatóságnál”, hogy minden formális rendszeren belül vannak, lehetnek magán a rendszeren belül bizonyíthatatlan tételek, sejtések. Ha a számítógéphez nem a velük elvileg megoldható feladatok oldaláról kívántunk volna közelíteni, talán nem is botlótunk volna ilyen elvi problémákba. Azoknak, akiket a problémákör jobban érdekel, a remélhetőleg előbb-több szélesebb körben hozzáférhető, idézett videofelvétel mellett Péter Rózsa: Játék a végtelennel című (Budapest, 1957.) könyvét ajánljuk.

Befejezésül még annyit: a számítógépek, ahogy lekerülnek a tervező asztalról vagy a gyári futószalagról, elvileg (látjuk, hogy ez mit jelent) univerzálisak, azaz a gyakorlati életben (majdnem!) minden feladat megoldására programozhatók. Egy adott pillanatban egy konkrét gép azonban csak azoknak a feladatoknak a megoldására képes, melyekre felkészítették:

— melyek megoldására a megfelelő programot elkészítették (megterveztek, kipróbálták) és

— „futásra kész” állapotban betöltötték a gépbe.

A megoldható feladatok köre egy-egy konkrét esetben tehát rögzített, és ezt a kört a tervező határozza meg. (Olyan értelemben természetesen a felhasználó is, hogy betölti a tervező által előzetesen elkészített programot.)

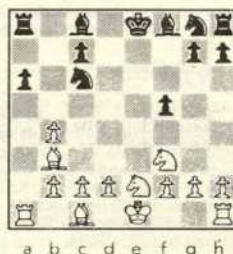
Messze vezetne annak a kérdésnek a boncolgatása, hogy mi van akkor, ha gépünk egy kiválasztott pontjára hangszórót vagy színes monitor kapcsolunk és így (előre nem tervezett módon!) érdekes — netán „művészi” — képeket vagy „zenét” is nyerünk, miközben a gép előre eltervezett módon éppen valamilyen matematikai problémát old meg. Hagyjuk ezt a kérdést is a filozófusokra!

A játéka és kiértékelése

Legutóbb az iteratív keresési eljárással foglalkoztunk, amelynek lényege, hogy az algoritmus nemcsak a végső csomópontban értékeli a játékfát, hanem a közepén is, és ennek alapján rendezzi. Így elértük, hogy nagy valószínűséggel a legjobb változatot elemezzük elsőként, és ennek eredményeként az alfa-béta algoritmus levágja a fa következő leveleit. Ezzel sok időt nyerhetünk, amit még több változat átszámítására fordíthatunk, növelve programunk játékerőjét.

Ha az iteratív eljárást az alfa-béta algoritlussal kombináljuk, annak hátrányai is vannak. Az alfa-béta algoritmus ugyanis — ellentétben a minimax eljárással — csak a legjobb lépésnek ad korrekciós pontértéket. Azt tudjuk tehát, hogy mi a legjobb lépés az adott szintű kutatás során, de a többitől nem tudunk egyebet, csak hogy azok nem a legjobb lépések. Mindössze annyi információval van róluk, hogy az adott pontérték, amit az alfa-béta keresés során kaptak, a maximális. Az biztos, hogy a lépés valódi értéke ennél nem nagyobb — ez a levágások következménye. Ha ugyanis az algoritmus — nem a legjobb gráf elemzése közben — egy elég jó lépést talál, akkor a fa többi részét levágja, hiszen annak elemzése már nincs szükség. Ilyenkor ennek az „elég jó lépésnek” a pontértéke kerül egy szinttel feljebb, annak ellenére, hogy legtöbbször esetében létezik ennél magasabb pontértékű lépés is.

Láthatjuk tehát, hogy a levágások miatt az alfa-béta algoritmus csak a legjobb lépésről ad felvilágosítást — kivéve az egy félépéses kutatást, ahol az értékelés módszere természetesen meggyezik a minimaxszal —, és emiatt az iteratív eljárásnál alkalmazott lépés-sorrendezés az egy félépésesnél mélyebb kutatás esetén feleslegessé válik, sőt általában rossz eredményt ad. Ezt



illusztrálja a táblázatokban látható pontfelvitel, amely az ábrán bemutatott hadállásban sötét lépését választja ki: minél kisebb a hexadecimális előjeles szám, az értékelőfüggvény szerint annál jobb az éppen lépésre következő fél pozíciója.

Az előbbi probléma megoldására KEMPELEN ATARI programom fejlesztése közben két ötletet támadt. Az egyik egy kézenfekvő, egyszerű eljárás. A program az első félépésű értékelés után sorba rendezzi a lépéseket, majd mélyebb kutatások esetén nem rendezi újból az összes lépést, hanem csak a legjobbakat helyezi át

Az egy félépéses kutatás eredménye

Ff8xb4	001b
Hc6xb4	0023
Hg8-f6	002b
Ff8-d6	002e
Ff7-e7	002f
Hg8-e7	0030
Ba8-b8	0031
Fc8-d7	0033
Hg8-h6	0034
a6-a5	0034
h7-h5	0034
g7-g6	0035
Ba8-a7	0035
h7-h6	0035
Ke8-d8	0036
Fc8-b7	003a
Ke8-d7	003b
Hc6-d8	003d
Ke8-e7	003d
Hc6-b8	0041
Hc6-a7	0042
Hc6-e7	0042
f5-f4	0049
g7-g5	004b
Hc6-d4	0069
Hc6-e5	0073
Hc6-a4	0074
Fc8-a6	0076
Ff8-c5	007a

A két félépéses kutatás eredménye

Ff8xb4	0022
He2-d4	ffde
Hf3-d4	ffe0
He2-c3	ffe0
He2-f4	ffe0
Fb3-d5	ffe1
c2-c3	ffe1
Fb3-c4	ffe2
0-0	ffe3
h2-h4	ffe3
Ke1-d1	ffe4
h2-h3	ffe4
g2-g3	ffe5
Bh1-g1	ffe5
Bh1-f1	ffe5
Ba1-a4	ffe6
Ba1-a2	ffe6
Hg8-h6	0022
Fb3-d5	ffde
Hg8-e7	0023
0-0	ffdd
Hc6xb4	0025
0-0	ffdb
Ff8-d6	0029
Fb3xg8	ffdf
Ff8-c5	0029
Fb3xg8	ffdf
Hc6-d4	002b
Fb3-	ffdf
Hc6-a7	a4+
002b	ffdf
Fb3-	ffdf
a4+	ffdf
002c	ffdf
Fb3xg8	ffdf
Ff8-e7	002c
Fb3xg8	ffdf
Hg8-f6	002d
c2-c3	ffdf
Fc8-d7	002e
Fb3xg8	ffdf
f5-f4	002e
Fb3xg8	ffdf
a6-a5	002f
Fb3xg8	ffdf
Hc6-d8	002f
Fb3-	ffdf
a4+	ffdf
h7-h5	002f
Fb3xg8	ffdf
g7-g6	0030
Fb3xg8	ffdf
g7-g5	0030
Fb3xg8	ffdf
Ba8-a7	0030
Fb3xg8	ffdf
h7-h6	0030
Fb3xg8	ffdf
Ke8-d8	0031
Fb3xg8	ffcf
Hc6-e7	0032
Fb3-a4	ffce
Fd8-b7	0035
Fb3xg8	ffcb
Hc6-b8	0035
Fb3-a4	ffcb

Ke8-d7	0036	Fb3xg8	ffca
Hc6-e5	0037	Fb3-a4	ffcf
Ke8-e7	0038	Fb3xg8	ffcf

az előző kutatás alapján már rendezett lépéslistán. Ennek egy továbbfejlesztett változata — amit tudomásom szerint más programban még nem alkalmaztak — abban különbözik az előbbi ötletemtől, hogy a legjobb lépés után következőknek is megközelítőleg korrekciós pontértéket ad. Ezáltal ismét úgy rendezhető az összes lépés, hogy nemcsak a legjobbnak a helye változik meg a listán, hanem a többi is, az állásértékelésnek megfelelően. Ezt a majdnem korrekciós pontértéket úgy kapom, hogy abban az esetben, ha az algoritmus egy vágó lépést talál, ennek pontértékével nem foglalkozik, hanem meghívja az egy szinttel feljebb levő értékelés pontját. Abban az esetben viszont, ha egy pozícióban nincs vágó lépés, és emiatt az összes lépés értékelése megtörtént, a minimax elvnek megfelelően a legjobbnak talál lépés pontértékét egy szinttel feljebb visszük.

Ezt a módszert elég hatásosnak tartom: a játéka megtartja teljes rendezettségét az elemzés végéig, továbbá lehetővé teszi, hogy minden esetben összehasonlíthassuk a legjobb lépést követő pontértékekkel, aminek különbségeit felhasználhatjuk az értékelésnél és a végső lépés kiválasztásánál is (lásd Mikroszámítógép Magazin, 1985/2. szám).

Két félépéses kutatás esetén egy kevésbé jó sötét lépésre egy közepes értékű világos válaszlépés is elég erős lehet ahhoz, hogy levágja a fa többi részét. Mivel világos nem az általában megtehető legjobb lépésével vágta le a fát, ezért sötét számára kedvező pontértéket ad az állásértékelés. Ennek következménye, hogy két félépéses kutatás után már nem a Hc6xb4-et tekinti a program a második legjobb lépésnek, hanem a Hg8-h6-ot. E lépés megtétele után ugyanis a program az Fb3-d5 válaszlépést értékelte, a hadállásban megtehető valóban legjobb c2-c3 lépés helyett.

KOVÁCS P. ATTILA

Aprócska játékos trükkök, melyek szórakoztatnak, és egy picit a matematikai ismereteket is felelevenítik

Petike kártyázik

Petike szeret kártyázni, de ritkán van partnere: apuka és anyuka soha nem ér rá, barátai pedig inkább számítógépes játékokkal szórakoznak. Így magára maradt hobbi-jával, a logikai feladatok megoldásával. Készített hát egy kártyázó programot. Problémáztához a „szírozást” választotta. Ennek a játéknak a szabályait bizonyára sokan ismerik, de azért szólunk róluk.

A szírozást a 32 lapos magyar kártyával játézzák. A játékban a színeknek nincs jelentőségük, csak a figuráknak: az a játékos győz, aki több „zsirt”, azaz tízest és ást gyűjt össze. Ezek a legértékesebb lapok tíz pontot érnek. Fontos szerepe van még a hetesnek is, amely nem ér ugyan pontot, de minden lapot üt. Egyébként ütni csak ugyanolyan lappal lehet, és ha a játékos bírja tartalékkal, akkor ismételhet.

Petike programja megszerkesztésénél a játék menetéből indult ki. Keverés után mindkét játékos négy-négy kártyát kap, a többit pedig lefelé fordítja az asztalra tesz ki. Az egyik játékos hív egy lapot. Ezt a partner vagy üti, vagy rátesz bármilyen másikat. Az utóbbi esetben a hívóé az ütés. Ha a másik játékos ütött, akkor az elsőnek joga van tovább ütni ugyanazzal a lappal vagy

```

10 REM SZIROZAS
20 DIM A(4),B(4),K(32),KR(8),V(32),JK(32),L(8)
30 FOR I=1 TO 32
40 K=K+1*(I)=X
50 NEXT I
60 REM KEVERES
150 PRINT "TURELEM KEVEREK"
170 FOR I=1 TO 32
180 V(I)=0
190 NEXT I
200 RANDOMIZE
210 A=INT(1200/HND)
220 IF X=32 THEN GOTO 210
230 FOR I=1 TO 32
240 IF V(I)=X THEN GOTO 210
250 NEXT I
260 FOR I=1 TO 32
270 IF V(I)<>0 THEN GOTO 290
280 V(I)=X
290 GOTO 210
290 NEXT I
300 FOR I=1 TO 32
310 H=V(I):K(K(H)=K(I))
320 NEXT I
400 REM JATEK KEZDODIK
410 FOR I=1 TO 4
420 A(I)=0:B(I)=0
430 NEXT I
440 Z=1:AD=0:BD=0:D=0
445 BEEP
450 INPUT "AKARSZ FEZDENI? I/N",W$
460 IF W$="N" THEN GOTO 570
470 GOSUB 5000
480 D=0:GOSUB 5300
490 Y=1:GOSUB 5000
495 GOSUB 5900
500 Y=2:GOSUB 5000
510 INPUT "HIVJAL LAPOT",H
520 H=H+1
525 GOSUB 5000
530 INPUT "TIOVABB,HIVAS,VELE T/H/V",R$
540 IF R$="H" THEN AD=AD+R:GOTO 470
550 IF R$="V" THEN BD=BD+R
560 D=D+R+D
570 D=0:GOSUB 5300
580 Y=1:GOSUB 5000
590 GOSUB 5900
600 Y=2:GOSUB 5000
610 REM GEP FEZD
620 NI=0:N4=0:HI=0
630 FOR I=1 TO 4
640 IF B(I)=1 THEN NI=NI+1:GOTO 670
650 IF B(I)=4 THEN NI=NI+1:GOTO 670
660 IF B(I)=8 THEN NI=NI+1
670 NEXT I
680 N4=NI+N4:HI=NI+HI
690 IF HI=3 THEN P=4:GOTO 700
700 IF HI=3 THEN P=8:GOTO 700
705 GOTO 720

```

```

700 FOR I=1 TO 4
705 IF B(I) <> P THEN GOTO 715
710 T=I:PRINT "LAPOM ",K(I)
711 GOTO 900
715 NEXT I
716 REM VALAMI HIVNI KELL
720 FOR I=1 TO 4
730 IF B(I)=1 THEN GOTO 785
740 IF B(I)=4 THEN GOTO 795
750 IF B(I)=8 THEN GOTO 795
770 T=I:PRINT "LAPOM ",K(I)
780 GOTO 900
785 NEXT I
790 FOR I=1 TO 4
800 IF B(I)=1 THEN GOTO 840
820 T=I:PRINT "LAPOM ",K(I)
830 GOTO 900
840 NEXT I
900 INPUT "MI A VALASZOD?",H
905 IF H=0 THEN GOTO 1100
910 IF A(H)=0 THEN GOTO 960
920 IF A(H)=1 THEN GOTO 960
930 GOSUB 5000
940 BD=BD+0:GOTO 570
960 IF B(T)=4 THEN GOTO 1000
970 IF B(T)=8 THEN GOTO 1000
980 A(H)=0:D(I)=0
990 PRINT "NEM TESZEK LAPOT"
995 GOTO 470
1000 GOSUB 5000
1010 FOR I=1 TO 4
1020 IF B(I)=1 THEN T=I:GOTO 1060
1030 NEXT I
1040 PRINT "NEM TESZEK LAPOT"
1045 GOSUB 5000
1050 AD=AD+1:GOTO 470
1060 PRINT "LAPOM ",K(I)
1070 GOTO 900
1100 IF Z<>0 THEN GOTO 560
1110 IF AD=BD THEN GOTO 1140
1120 PRINT "GAPUTALOK GYOZTEL"
1125 PRINT "PONTJAID",AD
1130 GOTO 150
1140 PRINT "SAJNALOM EN GYOZTEM"
1145 PRINT "PONTJAIM ",BD
1150 GOTO 150
5000 REM AZONOSITAS
5050 IF V=2 THEN GOTO 5100
5060 FOR I=1 TO 4
5070 L(I)=A(I)
5080 NEXT I
5090 GOTO 5130
5100 FOR I=1 TO 4
5110 L(I)=B(I)
5120 NEXT I
5130 FOR I=1 TO 4
5140 IF L(I)=0 THEN K(K(I))=V:GOTO 5200
5150 IF L(I)=1 THEN K(K(I))=V+1:GOTO 5200
5160 IF L(I)=2 THEN K(K(I))=V+1:GOTO 5200
5170 IF L(I)=3 THEN K(K(I))="X":GOTO 5200
5180 IF L(I)=4 THEN K(K(I))="Y":GOTO 5200
5190 IF L(I)=5 THEN K(K(I))="ALSO":GOTO 5200

```

```

5200 IF L(I)=6 THEN K(K(I))="FELSO":GOTO 5200
5210 IF L(I)=7 THEN K(K(I))="KIRALY":GOTO 5200
5220 K(K(I))="ASZ"
5230 NEXT I
5290 RETURN
5300 REM LAPOK HUZASA
5310 IF Z=32 THEN GOTO 5350
5305 FOR I=1 TO 4
5310 IF A(I)=0 THEN A(I)=K(K(2)):Z=Z+1
5320 IF B(I)=0 THEN B(I)=K(K(2)):Z=Z+1
5330 IF Z=32 THEN GOTO 5350
5340 NEXT I
5350 RETURN
5500 REM A GEP VALASZA
5501 FOR I=1 TO 4
5505 IF A(H)<>B(I) THEN GOTO 5520
5510 T=I
5515 PRINT "VALASZOM",K(I):GOTO 5750
5520 NEXT I
5525 REM ELBO LAP-E
5530 FOR I=1 TO 4
5540 IF B(I)=0 THEN GOTO 5570
5550 NEXT I
5560 GOTO 5650
5570 IF A(H)=4 THEN GOTO 5590
5580 IF A(H)=8 THEN GOTO 5590
5585 GOTO 5650
5586 REM VAN-E VII-ES
5590 FOR I=1 TO 4
5600 IF B(I)=0 THEN GOTO 5640
5610 IF B(I)<1 THEN GOTO 5640
5620 T=I:PRINT "VALASZOM",K(I)
5630 GOTO 5750
5640 NEXT I
5641 REM BARMILYET
5650 FOR I=1 TO 4
5660 IF B(I)=8 THEN GOTO 5720
5670 IF B(I)=1 THEN GOTO 5720
5680 IF B(I)=4 THEN GOTO 5720
5690 IF B(I)=8 THEN GOTO 5720
5700 T=I
5710 PRINT "VALASZOM",K(I):GOTO 470
5720 NEXT I
5725 FOR I=1 TO 4
5730 IF B(I)=0 THEN GOTO 5740
5731 T=I
5735 PRINT "VALASZOM",K(I):GOTO 470
5740 NEXT I
5750 RETURN
5900 REM PONTOK SZAMOLASA
5810 IF A(H)=4 THEN D=D+10
5820 IF A(H)=8 THEN D=D+10
5830 IF B(T)=4 THEN D=D+10
5840 IF B(T)=8 THEN D=D+10
5850 A(H)=0:B(T)=0
5860 RETURN
5900 REM LAPOK KIJELESE
5910 FOR I=1 TO 4
5920 PRINT
5930 PRINT I,"SZAMU LAP",K(I)
5940 NEXT I
5950 RETURN

```

hessel. A játék egy-egy fordulója alapján legfeljebb négy ütés lehet, mert ennyi a kézben levő lap. Ha valamelyik játékos nem üt tovább, akkor kiegészítik a kézben levő kártyák számát négyre. Ha az utolsó menetben a lefordított lapok száma kevesebb, mint kellene, akkor annyit vesznek fel belőle, hogy lapjaik száma egyenlő legyen. A játék végén megszámozzák „zsírjaikat”.

Érdeemes néhány partit kártyával lejátszani, hogy a program logikája érthetőbbé váljon.

Petikének először a keverést kellett megoldania. Ezt végzik el a 170—320-as sorok: a K tömb elemeit véletlenszerűen feltöltik 1—32-ig terjedő számokkal, és elemeihez rendelik a magyar kártya lapjait. A „keverés” viszonylag hosszú időt igényel, amely csökkenthető, ha gépünk véletlenszám-generátorát az 1—32 értékhatár között állítani tudjuk.

Petike a gép „gondolkodásának” programozásánál az élő játékos logikájából indult ki. Amikor a játékos hív, a gép az 5500-as programsortól kezdődően a következőket teheti. Ha van azonos lapja, akkor üt, és a játékosé a döntés joga. Ha nincs, megvizsgálja, hogy értékes-e a hívott lap. Ha igen, megpróbál hessel ütni (5570—5640-es sor). Egyéb esetben nincs mit tenni, lerak valamilyen lapot, de lehetőleg nem „zsírt”. Továbbtételnél ez az eljárás ismétlődik.

Híváskor a számítógép úgy „gondolkodik”, hogy csak akkor célszerű értékes lappal indulni, ha abból a játékosnak hetesekkel együtt legalább három van a kezében, mert a partnernek is lehet belőle; egyébként értéktelen lappal kell kezdeni (620—710-es sorok). Az ütés után az 5300-as sorral kezdődő szubrutin kiegészíti a játékosok lapjait négyre, az 5800-as sor pedig megszámozza pontjaikat. Az 5000-es sorral kezdődő szubrutin a partnernek lapjait azonosítja a dobások és húzások után.

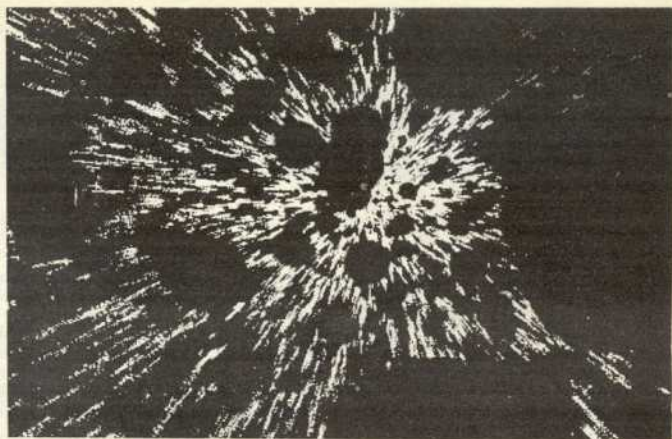
A program MS BASICA nyelven készült, a lehetőség szerint egyszerű utasításokkal. A játék hardver megvalósítása megtalálható Kovács Mihály Néhány kibernetikai játék című könyvében (Budapest, 1969. Tankönyvkiadó).

Az alkalmazott jelölések a következők. A K tömb a magyar kártya kódolt lapjait tartalmazza: VII=1, VIII=2, ... ász=8. A KK tömbben a már megkevert pakli, az A és B tömbben a játékosok kezében lévő kártyák vannak.

A program szubrutinokra épül. Ez a rövidítésen kívül lehetővé teszi a játékszabályok változtatását is: például bővíthető a játék azzal az engedménnyel, hogy nem kötelező azonos lapot tenni, ha van ilyen. A program keverést végző része más kártyajátékoknál is felhasználható.

Szívesen fogadunk ötleteket és programokat egyéb kártyajátékokról.

PINKE GYÖRGY



A természet vigyáz ránk

— Sietned kell, halld! Igyekezz!
2100. 11. 16
Híreket mondunk.

A legutóbbi felmérések szerint a világ népessége elérte a 8 milliárdot. A Biológiai Kutató Intézet közlése szerint elenyésző, 0,1 százalék azoknak a születéseknél a száma, melyek során csak egyetlen gyerek születik. Ezzel szemben 50,7 százalék a kettős, 30,8 százalék a hármas ikrek születési aránya. A fennmaradó szüléseknél háromnál több gyermek látja meg a napvilágot.

Érdekes eseményekről számol be a UPI hírügynökség. Közlése szerint a Sziklás-hegységben felfedeztek egy törzset, tagjai különös képességgel rendelkeznek: talpuk felé tudják fordítani a lábujjaikat, kapaszkodnak velük.

— Gyerünk, lassú vagy!
2105. 11. 16.
Híreket mondunk.

A legutóbbi felmérések szerint a világ népessége 10,5 milliárdra tehető. A Biológiai Kutató Intézet felmérése kimutatta, hogy már a nyolcéves gyermekek nemzőképesek.

A TASZSZ közlése szerint olyan új vércsoportba tartozó emberekre bukkantak, akiknek szervezete ellenálló a sugárfertőzésnek.

— Megint elkésem, siess!
2120. 11. 16.
Híreket mondunk.

A legújabb kutatások szerint a világ népessége elérte a 20 milliárdot. A két szemben álló ország között a helyzet egyre feszültebb.

Budapest. Az MTI idegen forrásra utalva közli, hogy egy Csendes-óceánban fekvő szigetben olyan embercsoport

találtak, amelynél kettős légzés alakult ki. Kopoltyúval és tüdővel is tudnak lélegezni.

A Biológiai Kutató Intézet felmérése szerint az embereknek egyre javul a szaglásuk, de ezzel párhuzamosan romlik a látásuk.

— Siess!
— Ne dumálgj bele, mindig miattad rontom el!
2200. 11. 16.
Híreket mondunk.

Készültségbe helyezte nukleáris elhárító rendszerét az egyik ország. A közvélemény-kutató irodák szerint az atomháború kitörésének lehetősége 97 százalék. A Biológiai Kutató Intézet szerint az emberi túróképesség határa jelentősen kitolódott. Képesek vagyunk például, három órán keresztül 70 Celsius-fokot elviselni, és négyheti szomjazást.

— Megint buksz!
— Kuss!
2225. 11. 16.

Híreink előtt hallgassák meg szerkesztőnk jegyzetét, A természet vigyáz ránk — címmel.

A természet vigyáz ránk. Igyekszük megóvni bennünket az atomháború borzalmaival. Ellenállóvá teszi a növényeket, az állatokat és...

Elnézést, de meg kell szakanunk adásunkat.
Fontos közleményt kell beolvasnunk. Kitérő a háború!
A képernyőn felirat jelenik meg. Ketten figyelték.

Az atomháború lefolyt.
Túlélők száma=0
Vesztett!

Akar még egyet játszani? I/N
HARANGOZÓ ANDRÁS

Már a μ M előző havi számában is közöltünk egy levelet, amely a programok illegális másolásával és terjesztésével foglalkozik. Az egyik szerkesztőségi írásom (Copyright — μ M 1985/2. szám) is elég nagy visszhangot váltott ki, a cikk megjelenése után meglehetősen sok telefont kaptam. A telefonálók egy részének a stílusa elég lekezelő volt („mit ugrál, ez ma a legnagyobb problémája a hazai számítástechnikának?”), de sokan hasonlóan gondolkodtak, mint a már megjelent és a most következő levél írója. Ha olvasóink részának egy levél megírására való időt, szívesen közölnék a problémáit (vagy az álproblámát?) kapcsolatos véleményüket.

Imhof Iván, Dombóvár,

Móricz Zs. u. 11. 7200

Egy komoly problémával kapcsolatban szeretném a véleményét és tanácsát kérni. Azt hiszem — bár ez esetben személyes jellegű — a számítástechnikát kedvelők és művelők népes táborát is érinti. Konkrétan a programok „illegális” másolásáról és a szoftvervédelemről van szó.

Jelenleg főiskolás vagyok, előtte egy évig operátorként dolgoztam, így az élet több területéről rendelkezem némi tapasztalattal a témát illetően. Azt hiszem köztudott, hogy az amatőr — hobby és tanulási célból — programozók körében a programok másolása és cseréje elfogadott és természetes dolog, bár jogilag talán vitatható. Még inkább az, ha a jelenség egy profi programozók-nál találkozunk, ami (szerintem) szintén nem ritka. Nálunk a főiskolán a programok úgy terjednek, mint az influenza. C64 gépre szinte az összes komoly program könnyedén megszerezhető (GEOS, Art Studio, Newsroom stb.). Azt hiszem, az említett körökben senkinek nem okoz lelkiismereti válságot a tudat, hogy kalózmásolat van a birtokában. Eddig magam is így voltam ezzel, bár tudtam, hogy a szoftver éppen úgy szellemi termék, mint a könyv, vagy egy zenemű.

Ennyit bevezetésül, a most következő rövid történetben pedig szándékosan nem akarok nevet említeni.

Nemrég másoltam egy rendszerprogramot, mellyel nyári gyakorlatomon szerettem volna dolgozni. Leírásom nem volt hozzá, de emlékeztem, hogy egy ismert szoftverforgalmazó cég hirdette a Mikromagazinban. Meg is találtam a hirdetést, majd felkerestem az illető céget. Elmondtam, hogy egy bizonyos programhoz leírást szeretnék venni. Teljesen jöhözémien úgy gondoltam, hogy ha a könyvesboltokban számtalan felhasználói programról kapható leírás, ez egy ilyen cégnél is lehetséges. A céget képviselő úr megkérdezte, miért nincs leírás, hiszen őnk minden eladott programhoz adnak. Megmondtam, hogy nem tölül vettem, másolással került hozzám. Erre az úr felháborodva, de udvarias hangnemből közölte, hogy szó sem lehet róla, ő nem segédkezhet egy szoftverlopásban, és egyáltalán nem vagyok képzeltem, hogy a forgalmazó céghez jövök a lopott szoftverhez leírást kérni. En kövé

dermedve álltam, hiszen bennem ez még soha sem vetődött fel ilyen komolyan. Kinos magyarázkodásba kezdtem: tudom, hogy a szoftver jogvédelem, és jogi-erkölcsi szempontból helytelen a másolás, de ha már megvan, szeretnék dolgozni vele. Az úr megjegyezte, hogy ez teljes képtelenség, és neki tulajdonképpen jegyzőkönyvet kellene felvenni, miképp jutottam én illegálisan egy ilyen programhoz. (Csak mellékesen mondom el, hogy vita közben elismert: néhány programhoz ő sem pénzért jutott hozzá. Ez számomra nem lenne szilárd erkölcsi alap a hasonló eljárások kritizálására.) Rövid vita után, melyben szavai nyomán szinte veszélyes bűnözőnek éreztem magam, távoztam.

Ennyi. Természetesen nem tudtam gyorsan napirendre térni a dolog felett, megpróbáltam több szempontból is végiggondolni.

A szoftvervédelem jogi szabályozásával nem vagyok tisztában — gondolom, sok amatőr programozóhoz hasonlóan. Így nem tudom, mit gondoljak: tényleg jogilag büntetendő lopást követtem el, mikor lemásoltam egy programot? Az számomra is világos, ha valamegy nem kereskedelmi úton szerzett szoftver segítségével netán én készítek egy eladható programot, az ebből származó hasznomból végül is az illető szoftverkészítőt károsítanám. Sőt, már az is, hogy nem pénzért juttattam hozzá. Valószínűleg én is így érezném, ha egy általam készített, kereskedelmi forgalomba hozott program (sajnos, ilyen még nincs) kalózmásolatával találkozok.

Ez az egyik oldal. A másik: a szoftverek másolás elleni védelme technikaiak lehetnének nehezebb, mivel majdnem minden kód megfejthető; ha azonban mégis sikerülne, annak kellemetlen következményei is lennének. A nem szexiális programokhoz nem jutnának hozzá a nélkülözhetetlen segédprogramokhoz (fordítók, belvők stb.), az egyéni felhasználók pedig a felhasználói programokhoz (szövegszerkesztők, adatkezelők stb.), hiszen az árak a többszer Ft-os nagyságrendnél kezdődnek. Én sem irhatnám levelem Easy Script-tel (ha jól tudom, kb. 10 000 Ft lehet). Szerintem a számítástechnika fejlődésében a fiatalok — elsősorban a diákok — rendkívül nagy szerepet játszanak új iránti fogékonyságuk, ötleteik miatt, ezért pedig pont ezt a réteget fosztanák meg sok lehetőségtől.

Visszatérve a velem történtek: ha a cég a másolást megakadályozni nem tudja, az lenne a helyes út, hogy „zárolja” a rá vonatkozó információkat? Ezzel lassítja ugyan a programmal való munkát, de megakadályozni nem tudja, mert akít igazán érdekel, az nem rád vissza, és előbb-utóbb magától kiismeri a rendszert. Így viszont csökken a programírásra fordítható idő. Számomra ez kissé a cég „bosszújának” tűnik, és nem értem, miért jobb így nekik. Talán ez is jogi kérdés lenne? Úgy vettem észre, könyvesboltjainkban a program megvásárlása nélkül is vehettek számtalan programleírást (Simon's, Help+u, Easy, Master stb.).

A Mikromagazinban is volt egy rövid ismertetés erről a programról. A cikk végén — „A téma iránt érdeklődőknek a következő szakirodalmat ajánljuk” címszóval több külföldi és magyar szakkönyv között megtalálható volt a fejlesztés és a forgalmazó cég által kiadott kézikönyv is. Ezek szerint az érdeklődőnek a programot is meg kell vennie, ha részletes információhoz akar jutni róla?

Összefoglalva: mennyiben törvénytellenes a szoftver ilyen formájú másolása? Lehetséges-e olyan megoldást találni, ami sem a szoftverkészítők, sem a kispénzű felhasználók érdekeit nem sérti? Van-e kitalálás rá, hogy a jövőben programot, szakirodalmat — esetleg még számítógépet is — nem a nyugati ár három-négyszereséért lehessen megvenni?

Most az egyszerű írt nem válaszolok a levélre, mert azt szeretném, ha olvasóink küldenék el véleményüket és válaszaikat.

Mocsary Gábor, Budapest,

Tűzoltó u. 8. 1094

Azt hiszem, hogy találtam egy tartósított „NO-VOTRADE gumicsont”-ot. Ebből rengeteg van, bárki meggyőződhet róla, ha megveszi a DATA BECKER — NOVOTRADE fordításokat. Azért nem nevezem könyveknek, mert a könyvek általában olvasni és érteni lehet, a fordítások meg olyanok, amilyenek.

A történetem röviden a következő. A COM-MODORE 64 létezésétől fogva keresem hozzá az irodalmat, így az említett nyomdaipari termékeket is megvettem. Ezeknél jobb programcsickek és fejtörők még kitalálni sem lehet. Kezdetben nagyon bosszantott a dolog, de később rájöttem — és kedvenc időtöltésemé vált —, hogy a hibák kerégselése közben lehet a legjobban tanulni. Most azonban megakadtam.

A „Tippek és trükkök a C64-hez” 55. oldalán és a „A VC—1541-es lemezegység programozása” 255. oldalán van két azonos elvű, a MIDS-et átalakító gépi kódú program. A jól beépített hibák miatt gyakorlatilag egyik sem működik, sőt mindkettő alkalmas arra, hogy a gépek azonnal „agygörcsöt” okozzon. A hibákat nem sorolom fel, mert túl hosszú volna. Mivel a programok alapötletét jónak találtam, nekialtám, hogy „beindítom” őket. Ekkor értek a meglepetések.

A javítások után a fordítást elsőnek a HELP PLUS-szal, másodikként a PROFI—ASS 64-gyel próbáltam. Amíg maga a lista hibás volt, addig a fordítókkal nem volt baj, csak a programok nem működtek. Ahogy az utastások és a címek a helyükre kerültek, a két fordító megbolondult. Például a mellékelt, a HELP PLUS-szal készített listán a 240-cédes sorig nincs baj. Ettől kezdve a fordító a 240-es sor BEQ, a 359-es sor BEQ, a 400-as sor BNE, a 428-as sor BNE, az 530-as sor BCS, a 680-as sor BEQ, a 700-as sor BCS és a 830-as sor BNE utastásainál — mindegyik észszezahasonlít! — a címeket rendre rosszul számította ki. A listán a címek jó helyen vannak, az ugrási címekre gondolok, az elvileg hibátlan listákon viszont már nem. A jelenség azért is furcsa, mert a PROFI—ASS 64 listája is teljesen azonos bolondéria szerint alakul.

Mivel a fordítók nem működtek jól, elkészítettem a BASIC betöltőt, a helyes ugrási címekkel.

A program így betöltve hibátlanul működik, amit a mellékelt REC/2-TEST programmal ellenőriztem.

A tudományom itt megállt! Sajnos nem jöttem rá, mi okozhatja a rejtélyes eseményeket. Ha valaki megfejtene a talányt, annak nagyon örülnék. Bizonyára sokan dühöngenek a NOVOTRADE program vicei miatt. Azt már meg sem merem kérdezni, hogy mivel készítek a hitetlenek eladott listákat (lásd a Tippek és trükkök 57. oldal 860-as programosát).

Kérésem az, hogy ha a témával nem foglalkoznak, erről egy rövid levélben értesítsenek. Amennyiben a megoldás érdekelne Önöket, és további adatokra volna szükség, úgy állok rendelkezésükre.

Levél elküldtem a Novotrade igazgatójának, aki mindig gondoskodik arról, hogy a levele munkatársai válaszoljanak. Íme a könyv lektorának válasza: Sajnos ki kell ábrándítanom, mert a „tartósított NOVOTRADE gumicsont” nem sikerült megtalálnom. Ellenben nekem okozott egy kis fejtörést az a lista, amelyet levelemben mellékeltem. A program én szerint a „Tippek és trükkök a C64-hez” c. könyvből való. Nos nem élteri tapasztalható a mellékelt és a könyvből program között. Valószínűleg figyelmenlenségéből ered, hogy programjában a 150. sor (835A címen): JSR \$B88D utastás szerepel, míg ha megnézi, a könyvből ez az utastás: JSR \$B88B! Nos ez az egyik oka annak, hogy programja nem működik. Sokat szoktam én is bosszankodni a figyelmenlenség begépelések okozta

„programhibák” miatt, ezek nem „programviccek”, hanem emberi tévedések. Annak érdekében, hogy számítógépe nem kapjon „azonnali agyöröcsőt”, a következőket tudom tanácsolni: az utasítások bejegyzésénél kövesse a könyvből megadott sorrendet. Tehát a programjában a 858. soról (→ 863. → stb.) kezdődő utasításokat helyezze a program elejére, mint ahogy azt a szerzők is tették, így a HELP PLUS is jól megbirkózik ez címek helyes kiszámításával. Ennen a problémán nem kell „dühöngeni”, „bosszankodni”, ez a HELP PLUS ill. a PROFIT—ASS 64 „hibája”.

Pataki János, Kiskunfélegyháza,

Horráth Z. ú. 2. 6100

A spectrumosok széles táborához tartozom, és gondjaim hasonlóak a többiekéhez. Én nem kevesleim a Spectrum cikkeket, inkább azzal vagyok elégedetlen, amikoről szólnak. Véleményem szerint programokat főlegesen közölni két okból is. Egyrészt egy valamire való program terjedelméért miatt nemigen való újsg hasábjaira, másrészt nem valószínű, hogy valaki begépelje. Ezenkívül bármilyen problémára több száz profi program forog közökben. A baj az, hogy megfelelő leírás hiányában nagy részüket lehetetlen használni. Ezt kellene a lapnak fővállalnia, és közreadni programok leírásait. Tudom, hogy a Spectrum Játék és Program című könyv 3 kötetét ezt a feladatot vállalta, de mivel ismertek a könyv megjelenési idők, az áttűtás nagyon hosszú, sok program elvesztéi aktualitását. Főleg felhasználói programok leírására gondolok, pl. ANIMATOR I., GRAPHICS ADVENTURE CREATOR, WHITE LIGHTNING, HP—80, BETA BASIC 3.2, ARTIST—II., de lehetne a sort nagyon sokáig folytatni. Ezenkívül sokan örülnének különböző játéktérképeknek, akár valamilyen külföldi lapból kölcsönözve, akár az olvasók által készített. Továbbá közzé lehetne tenni a legújabb öröklet POKE-okat.

Javasolom továbbá, hogy a két legelterjedtebb gép, a C64 és Spectrum legújabb programjairól legyen TOP lista. Gondolom ez igen egyszerűen megvalósítható — és szinte mindenki szívesen látta — hiszen szinte minden külföldi számítógépes lap közöl ilyeneket. Ez már csak azért is célszerű lenne, mert innen lehetne informálódni a legújabb programok megjelenéséről.

Másnak más a véleménye, nagyon sok olvasónk megírta, hogy bizony begépelik az általunk közölt programokat. Ezt persze abból is észleljük, hogy nagyon sokan elvlik a port rajtuk, ha a közölt programokban hiba van. A többi javaslatán elgondolozunk.

Iff. Szabó József, Oroszlány,

Haraszthygyi út 1/C 2840

Hadd kezdjem leveletem azzal, hogy ZX81-em hősri halált halt. Mégpedig úgy, hogy a modulátor ki kellett szedni azért, mert csak így tudtam megmenteni a ZX SPECTRUM-omat, mert a speci-é elromlott, és csak drága áron kaptam volna egy új modulátort. Szóval a speci — már működik (egy élet árán!), csak a tv-zsinór szakadt el, és javíthatatlan lett. Kérem Önt, hogy szólna a Sinclair Service-nek, hogy küldjenek nekem 200 Ft ellenértékben, utánvéttel egy gyári kábel. Spenci-ér. Nagyon megköszönöm, ha szólna nekik. Vagy valamelyik olvasóval lerendezném, hogy ha nem tudom Ön lerendezni.

Vannék Mega Basic 4.0 és Beta Basic 1.8 programra könyvet. Mert könyv nélkül nem tudom használni ezeket a programokat. Ugyanakkor levezeték program- és tapasztalatcsere céljából is.

Szívesen küzölöm a levélét, de megrendelések elintézését nem vállaljuk, túl sok a dolgunk az újsg-szerkesztéssel.

Avorosi László, Budapest,

Hámán Kató u. 4. 1096

Megköszönöm a levelemer adott választ (Mikroszámítógép Magazin 1987/6.).

Úgy látszik, jócskán illúziókat tápláltam, amikor feltételeztem, hogy sokak rendelkeznek nyomtatóval. Bocssá meg nekem, de én a folyóirat írásai mögött sejtettem színvonalat általánosítottam. Ezért tápláltam magamban egyfajta irigységet, úgy éreztem, hogy „mindenkinek” van nyomtatója, csak nekem nincs. Ez a hitem most megdőlt.

Korábban a Z80-as mikroprocesszorral kívántam foglalkozni, ezért vásároltam a ZX Spectrum-t. Időközben beláttam, hogy roppant előnyös volna hozzá egy nyomtató. Lélekben beletörtém még az árakba is. Elvégre olcsóbb hobbim, mint a kertészkedés vagy autózás. Azonban a vásárlás megoldhatatlan ideháza, mindenféle zugkereskedésben feltűnnek ugyan gyánús portékák, melyekről többnyire még műszaki információ sem képesek mondani, nemhogy interface-t kínáljanak. Gondoltam, hogy valamely nyugatra utazó ismerősömet megkérem a vásárlásra, de kiderült, hogy a devizaszámámlról nem adhatók át összegzet másnak. Jőmagam közlekedési viszonyból eredően nem utazok.

A nehézség ellenére remélem, hogy előbb-több lesz nyomtatóm, és akkor az Ön válaszában említett segítséget nyújthatom. A beszerzést illetően egy jó tippet örömmel fogadok.

Talán nem haraszlik, ha megragadom az alkalmat két apróság visszajelzésére. A '87/6. szám 6. oldalán tapasztaltam eltéréseket a programlistában. Szerintem a 80-as DATA sorban levő 133-as szám helyesen 33, és a 89-es DATA sorban 60 helyett 80-as szám kell. A lista 10-es és a 40-es sorából a LET értékadó utasítás hiányzik.

A másik észrevételém a '87/7. szám 31. oldalán látható ZX81 soroltó program vonatkozás. Ez a frappáns kis segédlet jól alkalmazható a ZX Spectrum gépen is, hiszen a Basic utasítássor formátuma e két géptípusnál azonos, csupán a memóriacím kezdőértéke tér el. Tehát Spectrum esetén, a lista 6-os sorában C-nek 23 755 értékkel kell adni (microdrive alkalmazásakor megint más értéket), s máris jól használható.

Végül hadd lapozom át ismét a MM legutóbbi '87/7. számát, megmutatva, mi érdekset találtam.

2. old. — Szerintem a távbeszélő-hálózat hiánya még súlyos károkat és feszültségeket fog okozni az informatika terén. Nyilvános adatbázisok terén a lemaradás már elkepeződött.

21. old. — görögországi beszámoló, tanulságokkal

23. old. — gondolatébresztő ez a sorozat a mesterséges értelemről

27. old. — az „implementáció”-t nem sikerült megfejtenem, szótárban sem találtam

36. old. — megkeresem a Video Elektronika gmk-t, hátha tudnak segíteni nyomtató ügyében

38. old. — végre megértettem a lapkezelési technikát!

41. old. — Nagyszerű! Szókimondó, igaz. A hazai számítógépes kultúráról

43. old. — élvezetes beszámoló az „Atari Kempelen” küzdelméről.

Ezzel a kis szemlélet igazolom, hogy a magazin ismét tartalmaz volt.

Egyetértünk, hogy a hazai nyomtató-vásárlási lehetőségek ma egyre több problémát jelentenek nemcsak az amatőr mozgalomban, de pl. az iskolai számítógépek használatában is. Egy kevesebb, mint tízezer forintot gép mellé egy többször tízezer forintos nyomtató megvásárlása meggondolandó. Ugyanez a helyzet persze a floppy-val is, ennek a két periferiának a hiánya — véleményem szerint

— egyre jobban akadályozza a házi számítógépek valóban hasznos alkalmazását (pl. könyvszerkesztés, adattárolás stb.). A megjegyzéseit és különösen a 7. számról írt szemléjét a szerkesztésé válamennyi munkatársra nevében köszönöm.

Nagy Zoltán, Budapest,

Árpád út 103. III. 3. 1042

Régi olvasója vagyok kitűnő lapjaink annál ellenére, hogy talán én vagyok az egyedüli magyar, aki nem ért a számítástechnikához.

Viszont szeretném, ha gyermeikem komolyabb kapcsolatba kerülhetnének ezzel a szép tudománnyal. Mint „határidő-nélküli előjegyző” hiába vártam C 16-os vagy C plus 4-es gépekre, — a forgalmazó még azt sem árulta el, hányadik vagyok hasonló sorsú társaim között. Ezért örömmel fogadtam a Centrum Áruházak akcióját, és vettem egy ENTERPRISE 128 típusú gépet. Azóta nem tudom, helyesen cselekedtem-e, mert a hozzátérők sajnálkoznak és becserlőien nyilatkoznak a gépről.

Azt hiszem többedmagammal szeretnék olyan objektív értékelést megismerni, mely helyre teszi az Ep 128-at. A használatnál is fellépnek olyan problémák, melyet nem tudok megoldani. Ilyen gond például az, hogy nem tudjuk a színes tv-video üzém éleonyeit hasznosítani, mert a tv-video csatlakozóháza bevitit számítógép jel nem ad színes képet.

A szakajót ezzel a géppel egyáltalán nem foglalkozik. Kérem ezért tisztelt Főszerkesztő Urat, hogy nagybecsűlt lapjainkban — pár sor erejéig — szíveskedjék foglalkozni az Ep 128 géppel, hiszen a hazai piacon ez az egyedüli, viszonylag hozzáférhető áron kapható számítástechnikai eszköz.

1. Nem Ön az egyedüli.

2. A µM 87/9. számában jelent meg a µ Klub rovatban a két írszemertő „Szinkronban az idővel” címmel.

3. Ami az Ep-t illeti, néhány barátom a gépet nagyon dicséri, különösen a szövegszerkesztővel elégedetek.

4. Sok olvasónk kérdezte, hogy mikor lesz az Ep kompatibilis a Sinclair Spectrummal, mert szeretnék használni a Spectrumhoz kapható programokat. Egyelőre csak annyit hallottunk, hogy: „majd”.

*

Egy korábbi olvasói levelünkre a Novotrade-tól az alábbi választ kaptuk

Tisztelt Kovács Győző elvtárs!

Hivatkozva 1987. június 9-én kelt levelére, melyben Fischer Kornél — Commodore problémára kért tanácsának — levelét mellékeltlen megküldték, írásban a lelvélrónak — cím hiányában — nem tudom válaszolni.

A kért kapcsolási rajzok Szervizünkben megtalálhatóak (Budapest, 1053 Magyar u. 1.)

A lelvélró — tanácsoljuk — közvetlenül keresse meg a Szerviz vezetőjét (Hegeudis Mihályt), akinek saját hatáskörébe tartozik a szervizdokumentáció kiadása.

Üdvözlettel:

P. H.

Záng Zoltán számítástechn. üzletágvezető	Friedrichné Újvári Zsuzsanna számítástechn. üzletágvezetőh.
--	--

Köszönöm olvasóink változatlanul aktív és szíves érdeklődését

KOVÁCS GYŐZŐ

Tájékoztatóul közöljük, hogy — korábbi hirdadásunktól eltérően — a C Plus 4 (1987. szeptember 14-én) a Skála Sprint áruházban kapható. Ára 15 000 forint, a hozzá tartozó magnó 3500 forint.

Csikós Zsolt:
Zene C-64/128 kedvelőknek
(Budapest, 1987.
LSI ATSZ,
136 oldal. Ára: 125,— Ft.)

Sokak szerint a számítógéppel nem lehet igazán zenélni, nem lehet a hangszerek tökéletesen utánozni. A zene számokkal való leírása embertelenné teszi a hangzást — mondják —, és ebben van is némi igazság. A számítógéppel való zenélés nem pótolhatja az élő zenét, de célja nem is ez, hanem új, szokatlan hanghatások keresése, a zenei lehetőségek szélesítése.

A szerző könyvének összeállításakor a BASIC nyelv ismeretét tételezte fel, remélve, hogy a programban előforduló trükkök a jártasabb olvasók számára is hasznosak lesznek.

A kötet első fejezete az olvasót a zeneelméleti alapismeretekbe és érdekességekbe vezeti be, majd ismerteti a SID 6581 chipet. Számos példát mutat be dallamok programozására, ismerteti a szintetizátort. Külön fejezet foglalkozik a zene és a grafika kapcsolatával.

ajánlások a hardverrel és szoftverrel kapcsolatban.

Az iskolai oktatásban még nem állnak kellő számban rendelkezésre szoftverek, és a meglévők sem az egységesen végiggondolt tantervek részei, hanem a meglévő tantervek egy-egy fejezetéhez írt programok. A középiskolai tanárok közül napjainkra közel 4000 fő részesült számítástechnikai alap- és továbbképzésben. Egyre nő a programokat író tanárok száma is.

Magyarországon 1983 szeptemberében indult el az iskolaszámítógép-program, amelynek eredményei, örömei és gondjai napjainkra már érzékelhetők.

A könyv azt a gondolatmenetet mutatja be, mely szerint a számítógépek oktatásbeli alkalmazása nem tantárgy, hanem elsősorban szemlélet, számítógépes műveltség. Kezdetben a tanárok egy része idegenkedett a számítógépes oktatástól, mert attól tartottak, hogy a gépek feleslegessé teszik munkájukat. Mára már világgossá vált, hogy a számítógépek nem pótolják vagy helyettesítik a pedagógusok tevékenységét, hanem hatékonyan egészítik ki azt.

Pajor Gábor:
Az IBM PC-ről kezdő felhasználók-nak
1. A hardver
(Budapest, 1987.
LSI ATSZ,
63 oldal. Ára: 66,— Ft.)

Ma a világon a személyi számítógépekkel dolgozók zöme felhasználó, aki munkája hatékonyságának növelésére használ számítógépet.

Ez a könyv az alkalmazókhöz szól. Célja, hogy a gépek használóit átsegítse a kezdeti lépések nehézségein. Olyan kérdésekre igyekszik választ adni, amelyek a legegyszerűbbek közé tartoznak, ezért sokszor nem is kerülnek szóba, és a géppel végül is egyedül maradt kezdőknek kell a — néha igen drága — tanulópenzt megfizetnie, vagy egyszerűen csak az idejét áldozza feleslegesen.

A könyv témái között olyanok szerepelnek, amelyek segíteni szeretnék azokat, akik még nem döntöttek, hogy milyen rendszert vásároljanak, mire legyenek tekintettel saját leendő hálózatuk kialakításában. A könyv szerzője azért választotta témáit az IBM PC XT és AT típusokat, mert ez az a két típus, amely a jelenlegi körülmények között beszerezhető és azt a teljesítményt nyújtja, amely ma a professzionális személyi számítógépektől elvárható.

A kötet a számítógép mechanikái és elektronikai részeivel foglalkozik, vagyis mindazzal, ami a gépben fizikailag jelen van. Részletesen ismerteti az IBM PC-k alapkonfigurációját, a háttértárolókat, a

monitorokat, a billentyűzetet, a nyomtatókat és plottereket, a számítógép és perifériáinak összekapcsolását.



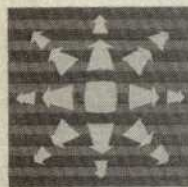
Faludi András:
A dBASE és a PROP-BASE
adatbázis-kezelő
rendszerek kézikönyve
(Budapest, 1987.
Számítástechnika-Alkalmazási
Vállalat,
322 oldal. Ára: 197,— Ft.)

A dBASE II és a PROP-BASE adatbázis-kezelő rendszer mikroszámítógépekre készült. Ezek az adatbázis-kezelők lehetővé teszik az adatok kényelmes kezelését — adatbázisok létrehozása, azok egyszerű módosíthatósága, nagyfokú egységesíthetőségük és fejlett belső automatikájuk révén.

Hazánkban a dBASE II 2.3. változata és a PROP-BASE rendszer a legelterjedtebb, így ezek teljes ismertetését tartalmazza a könyv. Mindkét adatbázis-kezelő használata középiskolai végzettség esetén minimális számítástechnikai ismeretekkel is elsajátítható. Természetesen ismerni kell hozzá az adott számítógép kezelését és tulajdonságait. A kezdők aránylag gyorsan megtanulhatnak értékes és használható programokat készíteni a könyvből közölt ismeretek alapján. Azok, akik már programoztak más nyelveken, könnyedén elsajátíthatják az adatbázis-kezelő rendszer használatát.

A könyv három fő részből áll. Az első rész ismerteti a rendszerek szerkezetét és nyelvi jellemzőit. A második rész az adatbázisok létrehozásától a feldolgozott adatok megjelenítéséig a rendszer alkalmazásának igényei szerint mutatja be a parancsok adta lehetőségeket egy-egy adott felhasználási környezetben. A harmadik rész a két adatbázis-kezelő rendszer parancsait tartalmazza alfabetikus sorrendben, részletes leírásokkal. Az első két részben megismert parancsok összes formáját megtalálhatja itt az olvasó. A függelék tartalmazza a parancsok, műveletek, beépített függvények pár szavas magyarázatát adó listát és a hibaüzenetek felsorolását.

Szűcs Pál:
Személyi számítógépek
az oktatásban



Szűcs Pál:
Személyi számítógépek
az oktatásban (Budapest, 1986.
OMIKK,
242 oldal. Ára: 98,— Ft.)

A 80-as évek elején a számítógépek megjelentek az oktatásban is. A számítógépek oktatásbeli alkalmazásának legfontosabb vonása az, hogy a tanítási-tanulási folyamatban új lehetőségek, módszerek megjelenését teszik lehetővé, miközben az egyes tanulók igényeihez alkalmazkodnak. Az elterjedésük és gyakorlati felhasználásuk ütemét jelentősen gátolja az a tény, hogy világszerte, sőt még az egyes országokon belül is, különböző, egymással nem kompatibilis berendezéseket használnak. Jelenleg nincsenek érvényben megfelelő szabványok és



Fotoelektronik — Novotrade — GT számítógép- szervizhálózat

Az ország egész területére kiterjedő és egyre bővülő szervizhálózatunk vállalja személyi és professzionális számítógépek, valamint perifériáik garanciális és térítéses javítását, karbantartását.

Kedvező feltételekkel kötünk átalánydíjas JAVÍTÁSI ÉS KARBANTARTÁSI szerződést!

Commodore típusú
(C16, C64, Plus/4, C128,
C610, C720, PC—10/20),
valamint

Atari, Sinclair Spectrum,
Primo, HT és más számítógépek,
Epson, Seikosha printerek
és egyéb perifériák
szervizelését, javítását
bízva szervizmérnökeinkre,
szakembereinkre!

Szervizeink:

Budapest V., Magyar u. 12—14.		
Telex: 22-7621	173-551	
Budapest VIII., Szilgyó u. 9.	343-153	
Pécs, Kolozsvár u. 20.	72/11-812	
Miskolc, Fazekas u. 4.	46/17-011	
Szeged, Székelysor 13.	62/13-377	
Békéscsaba, Bartók Béla u. 37.	66/27-195	
Szombathely, Szalonok u. 31.	94/14-519	
Debrecen, Holló L. u. 14.		

**Tekintse meg
az ELEKTROSOFT
Kisszövetkezet
bemutatóját
a SW '88-on!**

ÉPÍTŐIPAR

- költségvetés-készítés
- elemzés
- tervezés

HŰSIPAR

- ügyviteli csomag
- élőállat-gazdálkodás

ÁLTALÁNOS VÁLLALATI INFORMÁCIÓS RENDSZEREK

- főkönyv (kartonkezeléssel)
- folyószámla (kartonkezeléssel)
- anyagkönyvelés
- készletgazdálkodás

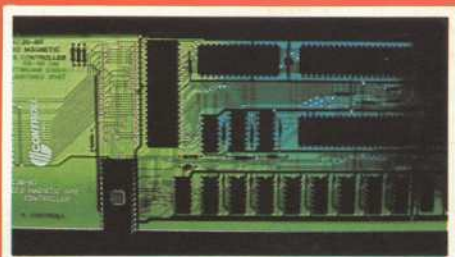
ALAPSZOFTVER

- szubrutincsomag C-hez
- konkurens IS fájlkezelő
NOVELL és MP/M alá
C—ISAM interfésszel

**A SLÁGER: 8 MUNKAHELYES,
CSOPORTOS ADATRÖGZÍTŐ**

Valamennyi termékünk portábilis.
A SW '88-on megtekintheti, hogyan
működik ugyanaz a 8 bites
technikán és NOVELL
hálózaton. Referenciahelyeink
képviselői is
az Önök rendelkezésére
állnak.

ÖSSZEÁLL A KÉP...



A SOFTWARE '88 kiállításon november 10. és 13. között

Újdonságok és jól bevált
rendszerek a hálózatos for-
mában működő szoftverek
között.

Bemutatjuk a CONTROLL
hardverein a Duna Inter-
Continental Szállodában.



Ha talál megfelelő progra-
mot, gépet is azonnal tudunk
szállítani kiegészítő egysé-
gekkel és különleges perifé-
riaillesztőkkel.

ÍGY VALÓBAN
ÖSSZEÁLL A KÉP!

CONTROLL
ELEKTRONIKAI ÉS SZÁMITÁSTECHNIKAI
KISZÖVEGYZET

Budapest, Szász K. u. 2. 1027
Tel.: 158-430; 158-428