



1986

október
Ára 30 Ft

Szellemgrafika
Helyi hálózatok
Játékprogramok
Barátunk a számítógép

Bevétel

orgtechnika
hungaria budapest

A MTE SZERVEZÉSI
ÉS VEZETÉSI
TUDOMÁNYOS
TÁRSASÁG
TAGJA SZERVEZÉSÉBEN

VI. SZERVEZÉSTECHNIKAI ESZKÖZÖK
ÉS AZOK ALKALMAZÁSA
SZAKKIÁLLÍTÁS ÉS
KONFERENCIA

* Budapest Sportcsarnok *
1986. október 3-8.



19
STAND

NOVOTRADA



1986.
OKTÓBER 3—8-ig

BUDAPEST SPORTCSARNOK NÉPSTADION ÚT 2.



**MIKROSZÁMÍTÓGÉP
MAGAZIN**

A NEUMANN JÁNOS SZÁMÍTÓGÉP- TUDOMÁNYI TÁRSASÁG ÉS A KISZ KÖZPONTI BIZOTTSÁG LAPJA

A kiadvány
a Tudományszervezési
és Informatikai
Intézettel
együttműködve készül

A szerkesztőbizottság
vezetője:
Kovács Győző

E számunkat
szerkesztették:
Bakos Tamás
(programozástechnika)

Broczkó Péter
(hírek)
Kovács Győző
(levelezés)

Lindner László
(sakkprogramozás)
Petróczy Judit
(könyvek)
Simonyi Endre
(klub)

Varga András
(iskola-számítógép)

Pálhalmi Vali
PR menedzser

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

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 hírlapkézbesítő
hivataloknál
és a Posta Hírlapelőfizetési
és Lapellátási Irodáján
(1900 Budapest V.,
József nádor tér 1.)
vagy átutalással a 215—96 162
pénzforgalmi jelzőszámra.
Egy szám ára 30,— Ft
Előfizetési díj:
fél évre 180,— Ft
egy évre 360,— Ft
Külföldön terjeszti
a Kultúra,
1389 Budapest, pf. 149.
és a Magyar Média
1932 Budapest, pf.
279. 86—253



Szikra Lapnyomda
Budapest (86-3951)
Felelős vezető:
CsönDES Zoltán vezérigazgató

INDEX: 25 629
ISSN 0236-6088

Címképünk:

ORGTECHNIK '86



Tartalom

Kettesben . . .	2
Assemblerek, cross-assemblerek	11
A minőségügy — közügy	20
A számítástechnika nem ámitástechnika	21
Olvastunk . . .	24
Helyi hálózatok	26
Ki ad magyarázatot?	33
Barátunk, a számítógép	38
Adok-veszek-cserélek	44

ISKOLA — SZÁMÍTÓGÉP

Az ismeretlen C16	3
Hangkeltés HT-gépen FORTH-ban	4

DIÁKROVAT

Programönkritika is . . .	5
Szövegszerkesztő és elemző	7
Szellemgrafika I.	8
Villogó grafika	9
Zeneprogram	10

PROGRAMOZÁSTECHNIKA

BASIC RAM felosztás	13
BASIC és gépi kód	15
Z80 programozási gyakorlatok 3.	16

μPROGRAMOK

PTA—4000. Egyváltozós függvények ábrázolása és megoldása	18
Tártartalom a képernyőn	19

JÁTÉKPROGRAMOK

A játékprogramozás technikája	22
Black Jack	23

μKLUB

PTA—4000. ROM leírás I.	29
A ZX81 bővítése	30
Botkormány	31
Prima ötlet	33

AZ OLVASÓ ÍRJA

SAKKPROGRAMOZÁS

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

KÖNYVEK

HÍREK, ÉRDEKESSEGEK	48
---------------------	----

*„Tanítani az ifjakat komoly szóval körülkeríteni a telket, és mert a világ bús tengere forr, felajzani a szilaj szenvedelmet, szólni a kor göztorlatairól, felverni az alvó nemzeti lelket.”
(Lator László: A hallgató Berzsenyi)*

A figyelmes olvasó bizonyára észrevette, hogy a μ M-nak már nemcsak egy tulajdonosa van, a Neumann János Számítógéptudományi Társaság, hanem kettő, ugyanis 1986. július 1-jétől, ahogy a lap címlapján is megjelent, a KISZ KB a lap társtulajdonosa.

A lapot az Ifjúsági Lapkiadó Vállalat gondozza, a szerkesztőségben a KISZ KB munkatársai is dolgozni fognak. Megalakul a lap szerkesztőbizottsága is; a bizottság elnökét és két tagját az NJSZT, három tagját a KISZ KB és egy tagját a Tudományszervezési és Informatikai Intézet delegálja. A szerkesztőbizottság feladata lesz a lap irányvonalának a meghatározása, tanácsadás, a szerkesztőség támogatása a felmerült problémák megoldásában.

Az előzetes megbeszéléseken egyetértettünk abban, hogy a lap alapkonceptiója, struktúrája nem változik, de szeretnénk a lapot még tartalmasabbá tenni, esetleg még a megjelenését is korszerűsíteni. Az elmúlt hónapokban jó kapcsolat alakult ki az Ölettel, amelyet tovább szeretnénk erősíteni, hiszen egy magazin jellegű havilap és egy hetilap együttműködése rendkívül sok előnyt jelenthet mindkét sajtótermék számára.

A KISZ-nek és az NJSZT-nek számos olyan terve van, amelyet — a lap szerkesztésében való együttműködés mellett — közösen szeretnénk megvalósítani. Ezek közül a legfontosabbak például a klub mozgalom kiterjesztése, a számítógépes amatőrizm

támogatása és fejlesztése, a távoktatási rendszer megvalósítása, amelyeket — összefogással — rövidebb idő alatt és valószínűleg gazdaságosabban lehet megvalósítani, mint külön-külön, arról nem is beszélve, hogy ha külön úton járunk, akarva-akaratlanul még nehézségeket is támaszthatnánk egymásnak.

Szeretnénk a — kissé elaludt — μ Könyvtár sorozatot is folytatni, már előkészületben van a következő, a Műszaki Könyvkiadó gondozásában megjelenő kötet, H. H. Goldstine informatikaitörténeti munkája, amelynek címe: „A számítógép Pascaltól von Neumannig”. Nagyon szoros együttműködésünk van a Magyar Televízióval (Mi és a számítógép, a TV BASIC és az újabb oktatófilmek), de elkezdődik a munka a Magyar Rádióval is. A legközelebbi tervünk egy közös kéthetenkinti magazinműsor, amelyet ugyancsak most készítettünk elő. A Tudományszervezési és Informatikai Intézzel és így a magyar oktatásügyvel a szoros kapcsolatunkat nem csak a TII munkatársának szerkesztőbizottsági tagsága jelzi, de a távoktatási rendszert is úgy szeretnénk megvalósítani, hogy az a magyar köz- és felsőoktatás része legyen.

A Neumann Társaság természetes állandó szövetségesei voltak a főhatóságok, szeretnénk, ha a KISZ KB-vel való együttműködésben is számíthatnánk a támogatásukra, és így magasabb szintre emelhetnénk kapcsolatunkat az OMFV-vel, az IpM-mel, a KSH-vel és az MM-mel.

Meggyőződésünk, hogy igazán nagy és komoly eredményt — ma amikor az anyagi körülmények egyre rosszabbodnak — csak összefogással, a meglévő szellemi energiáknak és anyagi erőknak a koncentrálsával lehet elérni.

Ehhez valamennyiünknek a legjobbkat kívánom.

KOVÁCS GYÖZŐ

A hazai szervezőtechnika nagy találkozója a kétévente megrendezésre kerülő ORGTECHNIK kiállítás, valamint a vele párhuzamosan folyó tanácskozás. Itt egy máshoz mérhetik magukat a szervezéssel és szervezőtechnikai segédeszközök gyártásával foglalkozó hazai nagy- és kisvállalkozások, de arra is lehetőség nyílik, hogy megismerhessék: vajon hol állnak e területen a külföldi cégek. Az idei kiállítást, amely a hatodik a hasonló rendezvények sorában, október 3–8. között a Budapest Sportszínház arénájában és galériájában tartják, s a kiállításához kapcsolt kongresszusra pedig október 6–8. között kerül sor. Az idei rendezvényről beszélgettünk dr. Pomper Bélaival, aki a Szervezési és Vezetési Tudományos Társaság Szervezési Szakosztálya, valamint a Számítógéppalkalmazási Munkabizottság részéről fogja össze a szervezés feladatait.

— Milyen újdonsággal, érdekességgel találkozhatunk az idei rendezvényen?

— A kiállítás célja elsődlegesen a szervezési, számítástechnikai módszerek bemutatása. A szervezési módszer azonban nem olyan látványos dolog, amelyeket érdeklődést keltően ki lehetne állítani. Így a külső szemlélő óhatatlanul egy eszközkészletnek, a szervezőtechnikai segédeszközök kiállításának lesz a látogatója. A technikának ez esetben a szervezést kell szolgálnia, s amit az előzetes jelentésekből látható, a kiállítók messzenemően igyekeznek így összeválogatni a bemutatójukat. Természetes, hogy a számítástechnika egyre inkább bevonul a mindennapi életbe. Így van ez a szervezőtudományban is, ahol a korábban fejben vagy papíron végzett adatelemzési, rendezési, statisztikai rutinmunkát elvégzi a számítógép. De az ipari szervezés részét képezik például az ipari robotok is...

— Ezek szerint nem csak számítógépeket, adatfeldolgozó berendezéseket láthat a publikum?

— A kiállításon részt vesz 5 nagy szociálista egyesülés a baráti országokból. Hazánk 110 nagyvállalatával és sok kisvállalkozással képviselteti magát. Önálló standdal jelentkezik az Alkotó Ifjúság Egyesülés, a Videoton, a Terta, SZKI, SZÜV, Novotrade, Csepel, SZAMALK, a Magyar Posta, valamint az 5G kisszövetkezet, hogy a várhatóan legérdekesebb hazai bemutatókat említsem, a teljesség igénye nélkül szemezgetve az előzetes listából.

— Tökés országokból is várnak kiállítókat?

— Még nem zárultak le a jelentkezések végleges, így korai lenne e tárgyban pontos adatokat mondani. Mintegy 20 cég jelezte részvételét. Már biztosnak látszik a REX-Rotary másológép, a Robinco fototechnikai, a DUIT cég mikrofilmetek berendezéseinek kiállítása. S hogy a számi-

Az ismeretlen C16

KERNAL rutinok 2.

A beviteli/kiviteli KERNAL rutinok egy összefüggő csoportjának leírásával folytatva a C16 ismertetését. Ezek közül a BSOUT az előző részben is szerepelt, most részletezem az ott leírtakat.

Az alábbiak között öt olyan rutin van, mely átvitelbit beállításával jelzi, hogy a végrehajtás hibátlan volt-e. Ha nem volt hiba, a C-bitet törli, ellenkező esetben a C-bitet 1-re állítja, és az A regiszterbe teszi a hiba kódját. Ajánlatos a hívó programrészbe a következő szerkezetű ellenőrző részt beépíteni:

JSR rutin
BCC TOVABB
JMP (\$0300)
TOVABB...

A rutinok

SETLFS \$FFBA 65 466

A fájl-paramétereket helyezi el az erre kijelölt rendszerváltozóknak. Ezek általában megegyeznek a BASIC OPEN utasításának első három paraméterével. Hívása előtt a paramétereket a regiszterekbe kell tölteni, az alábbiak szerint:

A — a logikai fájl-szám,

X — a fizikai eszközsorszám, ez megegyezik a BASIC-ben is használatossal,

Y — másodlagos cím (megnyitási mód, ill. csatornaszám).

SETNAM \$FFBD 65 469

A fájlnévre vonatkozó információkat helyezi el a kijelölt rendszerváltozóba. Hívása előtt az A regiszterbe a fájlnev karaktereinek a számát (a név hosszát) kell tölteni, név nélküli fájl esetén 0-t. Az X/Y regiszterpár a fájlnev kezdőcímét kell hogy tartalmazza úgy, hogy a cím nagyobb helyértékű bájta kerüljön Y-ba.

OPEN \$FFC0 65 472

Megnyitja azt a fájlt, melynek paramétereit előzőleg a SETLFS és a SETNAM ruti-

nokkal az erre kijelölt rendszerváltozóba küldték. Hiba esetén a bevezetőben leírtak szerint működik.

CLOSE \$FFC3 65 475

Lezárja azt a fájlt, melynek logikai száma azt híváskor az A regiszter tartalmazza.

CHKIN \$FFC6 65 478

Beviteli csatornát készít elő olvasásra. Hívása előtt az X regiszterbe annak a korábban OPEN-nel megnyitott fájlnak a logikai számát kell elhelyezni, melyből a következő olvasások történnének a BASIN rutinnal. A hibakezelés itt is a bevezetőben leírtak szerint történik.

CHKOUT \$FFC9 65 481

Kiviteli csatornát készít elő írásra. Az adatáramlás irányától eltekintve megegyezik a CHKIN rutinnal.

CLRCHN \$FFCC 65 484

Lezárja a CHKIN és CHKOUT rutinokkal megnyitott csatornákat, és visszaállítja az alaphelyzetet, azaz a továbbiakban az aktuális beviteli eszköz a billentyűzet, a kiviteli a képernyő lesz.

BASIN \$FFCF 65 487

Beolvassza egy karaktert az aktuális beviteli csatornáról az A regiszterbe. Alaphelyzetben (CHKIN használata nélkül, ill. CLRCHN után) a billentyűzetről olvas, a képernyőszerkesztő segítségével.

BSOUT \$FFD2 65 490

Kiküld egy karaktert az A regiszterből az aktuális kiviteli csatornára. Alaphelyzetben (CHKOUT használata nélkül, ill. CLRCHN után) a képernyő kurzorpozíciójába ír, és — néhány kivételtől eltekintve — a kurzort egy hellyel jobbra lépteti.

LOAD \$FFD5 65 493

Programfájl betöltő/ellenőrző rutin. Hívása előtt a fájl adatait a SETLFS és SET-

NAM rutinok segítségével be kell állítani. Ha hívásakor az A regiszter 0-t tartalmaz betölti a programot a háttértárolóról, ha 1-et, akkor összehasonlítja a tárbán lévő programot a háttértárolón lévővel. Ha a SETLFS hívásakor a megnyitási mód kódja 0 volt, a betöltési kezdőcímet az X/Y regiszterpárba kell a LOAD hívása előtt beírni, 1-es megnyitási mód esetén a betöltési címet a fájlból veszi.

A cím regiszterbe írásakor az Y-ba kell a nagyobb helyértékű bájtot tölteni, X-be az alacsony helyértékűt. Hibátlan végrehajtás után a regiszterpár a végcím+1 értéket fogja tartalmazni.

A hibakezelés itt is a bevezetőben leírtak értelmében történik.

SAVE \$FFD8 65 496

A programfájlt a háttértárolóra viszi. Használata előtt a fájl jellemzőit a SETNAM és SETLFS rutinokkal be kell állítani. Ezután a program kezdőcímét egy nulla-lapos bájtpáron kell elhelyezni, az A regiszterbe ennek a bájtpárnak a címét kell tölteni. Az X/Y regiszterpárba az első olyan bájti címet kell bevinni, amely már nem tartozik a programhoz (végcím+1). A címet úgy kell a regiszterekbe tenni, hogy az Y regiszterben legyen a cím magasabb helyértékű bájta.

A kezdőcím elhelyezésére még később visszatérek, most annyit ajánlok, hogy C16 esetén \$D8...\$D9 (216...217), a többi géptípus esetén \$FB...\$FC (251...252) című helyek alkalmasak erre a célra. Ne feledkezzünk meg arról, hogy itt is az alacsonyabb helyértékű bájtot kell előre írni!

A hibakezelés itt is a bevezetőben leírtak szerint történik.

CLALL \$FFE7 65 511

A CLRCHN hívásával lezárja a megnyitott csatornákat, majd lezárja a megnyitott fájlakat, és visszaállítja az alaphelyzetet.

BARNA LÁSZLÓ

tástechnika se maradjon ki: itt van például az INTERMEC, amelyik vonalkódtechnikai berendezéseket és a COMBIRING, amely mikroszámítógépeket mutat be. S várhatóan sok egyéb érdekességgel is találkozhat a látogató, aki október 3-8. között, naponta 10-18 óráig felkeresi a Budapest Sportcsarnokban lévő bemutatóstandokat. Itt mindenkit, laikust, szakembert vagy éppen diákokat szívesen látnak a kiállítókat. De mindenkit arra kérek: ne csak az elektronikát nézze! Az csak eszköz egy probléma megoldására.

— Meghívottak részére a kiállítás vége felé tudományos konferenciát is rendeznek, idén már másodikban. Mi ennek a programja?

— Kettős céllal rendezzük: először is azért, hogy aki kiállít, az elmagyarázhassa előadásában, milyen módszereket alkalmaz a gyakorlatban. Aki viszont nem állít ki, de alkalmazza a szervezőtechnikát, az is ismeresse tapasztalatait. Mind a kongresszus, mind a kiállítás alkalmat nyújt arra, hogy szolgáltató és felhasználó megtalálhassa egymást. Hazai feladat van elegendő,

s vállalkozók is akadnak ennek megoldására. A feladatunk az, s ezért szervezzük ezeket a rendezvényeket a Szervezési Vezetési Tudományos Társasággal, a Neumann János Számítógéptudományi Társasággal és az Országos Vezetőképző Intézettel közösen, hogy nőjön a szervezési iránti igényesség, a partnerek egymásra találhassanak, mint egy szakvárosban, s nem utolsósorban a szakma egy helyen láthassa: ilyen a magyar szervezési kultúra. Nézzétek! Itt tartunk, s azt is láthatjuk, merre kellene tovább fejlődünk.

KIS

Hangkeltés HT gépen FORTH-ban

Ha a FORTH rendszerben nincs beépített assembler szótárunk, akkor is írhatunk egyszerűbb gépi kódú primitíveket. Néha erre kifejezetten szükség van. Vegyük egy példát. A FORTH kiválóan alkalmas különféle hardvereszközök vezérlésére. Ha azonban a BASIC-nek megfelelő INP és OUT utasítások nincsenek meg, akkor éppen a legfontosabb láncszem hiányzik. A fenti szókat kellene használnunk a HT iskolaszámítógépen a hang létrehozásához is. Bizonyára többen fordulnának a FORTH nyelvhez, ha könnyen elérhető lenne néhány ilyen praktikus dolog.

Először is létre kell hoznunk azokat a primitíveket, amelyek a portokat kezelik. A gépi utasítások felépítését assembler hiányában magunknak kell megszerveznünk. Segítségképpen az OUTC és INPC primitíveket közöljük. A kódokat legegyszerűbben a C, és a, FORTH szókkal rakhatjuk le. Tudnunk kell, hogy a CREATE szó egy primitív fejrész hoz létre, és a SMUDGE szó érvényesíti azt. Próbaképpen hozzuk létre a DUP FORTH szót újra. Ez elég rövid ahhoz, hogy megérintse kipróbálni. A szó egyébként a rendszerben így van megvalósítva (1. lista).

A sor lezárásakor a DUP IS NOT UNIQUE üzenet jelenik meg, mert a DUP szó már megvan a szótárban. Ez azonban most csak azt jelenti, hogy az új DUP van érvényben. A primitívek létrehozásakor néhány dologra ügyelnünk kell. A FORTH rendszer a működések egymáshoz láncolt primitíveket hajt végre. Néhány különleges esettől eltekintve ezek mindegyike egy meghatározott rutinra ugrik, amely gondoskodik a következő primitív megkereséséről. Ez a rutin az implementációtól függően helyen kezdődik, fizikailag általában a primitívek előtt helyezik el. Az általunk használt rendszerben a 4CBAH, 4CBBH vagy 4CBCH címen kezdődik, attól függően, hogy akarunk-e a veremre eredményt elhelyezni. Ez a rutin feltételezi, hogy a Z80 processzor BC regiszterpárját azonos állapotban hagyjuk. Amennyiben mégis használjuk, a tartalmát meg kell őriznünk és vissza kell töltenünk.

Ha a DUP szó újradefiniálása sikerült, bátran nekifoghatunk egy komolyabb feladatnak. Hosszabb primitívek létrehozásához jobb módszert ajánlunk. A 2. listában bemutatjuk az assembler listát, a 3. listában a FORTH megvalósítást.

A LERAK szót azért hoztuk létre, hogy hosszabb primitíveket kényelmesebben lehessen beírni. A szó egy a verem tetején ha-

```

1 TASK 1
  DUP ISD
  1. E1 POP HL
  2. E5 PUSH HL
  3. CBBAC JF ACBBH
  UVSAMEZ FORTH-BAN
  HEX CREATE DUP SMUDGE EI C5 E5 C2 E4 ACBB + DECIMAL
VEGERNAUTVA
4. VEREM 1 - (HL)
5. VEREM 1
3. 3. ON

```

1. lista

```

OUTC
3. E3 POP HL 2 ADAT
2. 7D LD A,L
3. E1 POP HL 3 PERIFERIA
4. E5 PUSH BC 2C PERIFERIA
5. E5 PUSH HL
6. C5 POP BC 1 PERIFERIA
7. E5 PUSH HL
8. C5 POP BC 1 PERIFERIA
9. C3 POP BC 1 PERIFERIA
10. C3 POP BC 1 PERIFERIA
11. C3 POP BC 1 PERIFERIA
12. C3 POP BC 1 PERIFERIA
13. C3 POP BC 1 PERIFERIA
14. C3 POP BC 1 PERIFERIA
15. C3 POP BC 1 PERIFERIA
16. C3 POP BC 1 PERIFERIA
17. C3 POP BC 1 PERIFERIA
18. C3 POP BC 1 PERIFERIA
19. C3 POP BC 1 PERIFERIA
20. C3 POP BC 1 PERIFERIA
21. C3 POP BC 1 PERIFERIA
22. C3 POP BC 1 PERIFERIA
23. C3 POP BC 1 PERIFERIA
24. C3 POP BC 1 PERIFERIA
25. C3 POP BC 1 PERIFERIA
26. C3 POP BC 1 PERIFERIA
27. C3 POP BC 1 PERIFERIA
28. C3 POP BC 1 PERIFERIA
29. C3 POP BC 1 PERIFERIA
30. C3 POP BC 1 PERIFERIA
31. C3 POP BC 1 PERIFERIA
32. C3 POP BC 1 PERIFERIA
33. C3 POP BC 1 PERIFERIA
34. C3 POP BC 1 PERIFERIA
35. C3 POP BC 1 PERIFERIA
36. C3 POP BC 1 PERIFERIA
37. C3 POP BC 1 PERIFERIA
38. C3 POP BC 1 PERIFERIA
39. C3 POP BC 1 PERIFERIA
40. C3 POP BC 1 PERIFERIA
41. C3 POP BC 1 PERIFERIA
42. C3 POP BC 1 PERIFERIA
43. C3 POP BC 1 PERIFERIA
44. C3 POP BC 1 PERIFERIA
45. C3 POP BC 1 PERIFERIA
46. C3 POP BC 1 PERIFERIA
47. C3 POP BC 1 PERIFERIA
48. C3 POP BC 1 PERIFERIA
49. C3 POP BC 1 PERIFERIA
50. C3 POP BC 1 PERIFERIA
51. C3 POP BC 1 PERIFERIA
52. C3 POP BC 1 PERIFERIA
53. C3 POP BC 1 PERIFERIA
54. C3 POP BC 1 PERIFERIA
55. C3 POP BC 1 PERIFERIA
56. C3 POP BC 1 PERIFERIA
57. C3 POP BC 1 PERIFERIA
58. C3 POP BC 1 PERIFERIA
59. C3 POP BC 1 PERIFERIA
60. C3 POP BC 1 PERIFERIA
61. C3 POP BC 1 PERIFERIA
62. C3 POP BC 1 PERIFERIA
63. C3 POP BC 1 PERIFERIA
64. C3 POP BC 1 PERIFERIA
65. C3 POP BC 1 PERIFERIA
66. C3 POP BC 1 PERIFERIA
67. C3 POP BC 1 PERIFERIA
68. C3 POP BC 1 PERIFERIA
69. C3 POP BC 1 PERIFERIA
70. C3 POP BC 1 PERIFERIA
71. C3 POP BC 1 PERIFERIA
72. C3 POP BC 1 PERIFERIA
73. C3 POP BC 1 PERIFERIA
74. C3 POP BC 1 PERIFERIA
75. C3 POP BC 1 PERIFERIA
76. C3 POP BC 1 PERIFERIA
77. C3 POP BC 1 PERIFERIA
78. C3 POP BC 1 PERIFERIA
79. C3 POP BC 1 PERIFERIA
80. C3 POP BC 1 PERIFERIA
81. C3 POP BC 1 PERIFERIA
82. C3 POP BC 1 PERIFERIA
83. C3 POP BC 1 PERIFERIA
84. C3 POP BC 1 PERIFERIA
85. C3 POP BC 1 PERIFERIA
86. C3 POP BC 1 PERIFERIA
87. C3 POP BC 1 PERIFERIA
88. C3 POP BC 1 PERIFERIA
89. C3 POP BC 1 PERIFERIA
90. C3 POP BC 1 PERIFERIA
91. C3 POP BC 1 PERIFERIA
92. C3 POP BC 1 PERIFERIA
93. C3 POP BC 1 PERIFERIA
94. C3 POP BC 1 PERIFERIA
95. C3 POP BC 1 PERIFERIA
96. C3 POP BC 1 PERIFERIA
97. C3 POP BC 1 PERIFERIA
98. C3 POP BC 1 PERIFERIA
99. C3 POP BC 1 PERIFERIA
100. C3 POP BC 1 PERIFERIA

```

2. lista

```

1 TASK 1
  DUP ISD
  1. E1 POP HL
  2. E5 PUSH HL
  3. CBBAC JF ACBBH
  UVSAMEZ FORTH-BAN
  HEX CREATE DUP SMUDGE EI C5 E5 C2 E4 ACBB + DECIMAL
VEGERNAUTVA
4. VEREM 1 - (HL)
5. VEREM 1
3. 3. ON

```

3. lista

```

1 TASK 1
  DUP ISD
  1. E1 POP HL
  2. E5 PUSH HL
  3. CBBAC JF ACBBH
  UVSAMEZ FORTH-BAN
  HEX CREATE DUP SMUDGE EI C5 E5 C2 E4 ACBB + DECIMAL
VEGERNAUTVA
4. VEREM 1 - (HL)
5. VEREM 1
3. 3. ON

```

4. lista

```

1 TASK 1
  DUP ISD
  1. E1 POP HL
  2. E5 PUSH HL
  3. CBBAC JF ACBBH
  UVSAMEZ FORTH-BAN
  HEX CREATE DUP SMUDGE EI C5 E5 C2 E4 ACBB + DECIMAL
VEGERNAUTVA
4. VEREM 1 - (HL)
5. VEREM 1
3. 3. ON

```

5. lista

```

10 DATA 152,2;152,2;152,4;135,2;135,2;135,4;120,2;113,2;101,2;152,2;161,2;152,2;
135,2;203,2
120 DATA 152,2;152,2;152,4;135,2;135,2;135,4;120,2;113,2;101,2;152,2;135,2;161,2;
152,4
30 OUT31;7;OUT30;254;OUT31;8;OUT30;9
40 FOR L=1 TO 27
50 READ P;0
60 OUT31;0;OUT30;P
70 FOR T=1 TO 0*30+NEXT
80 OUT31;0;OUT30;0
90 NEXT L

```

gyott szám alapján a veremről sorba az aktuális HERE címre rakja az adatokat. Előnye, hogy a gépi utasításokat a rendes sorrendben lehet megadni. Ezek ugyan a verem fordított sorrendben jelennek meg a LERAK szó azonban a verem aljáról indulva bányássza ki őket. Vigyáznia kell a használatával, mert az SP! szó újra inicializálja a vermet.

Az OUTC és INPC szókban a C végződés arra utal, hogy a primitívben a port címet a C regiszter adja meg.

Ha a létrehozott szókat nemcsak hangkeltésre akarjuk használni, akkor az INIC, KIMEN és BEHOZ szók alkalmasak a perifériával való kommunikációra.

A CVEK egy definiáló szó. Példánkban a TENG szót hoztuk létre vele. A szó definiálásakor a szó neve után rendre lerakja a szó előtt megadott egyéjtős számokat, a megadás sorrendjében. Itt a sorozat letelején azonban azt találjuk majd, hogy hány eleme van a sorozatnak. A definiáló szó végrehajtásakor egy-egy számot vár a verem, és az ennyedik bajtot fogja ráhelyezni. Közben ellenőrzést is végez, hogy létezik-e egyáltalán annyiadit szám. CVEK segítségével tehát olyan szókat hozhatunk létre, amelyek tárolják a portra kivinni kívánt információt.

A 4. listán egy dallam programozását mutatjuk meg. A H1, H2, H3 vektorváltozóba tettük a hangkódokat, az S1, S2, S3 vektorváltozóba pedig a hosszúságokat. A HANG szók adják a dallam egyes részét. Mellékeljük a dallam BASIC változatát is (5. lista), hogy legyen módja az olvasónak összehasonlítani. Néhány érdekességre felhívjuk a figyelmet.

BASIC-ben a hangzás hosszúságát egy 30-szoros ciklus adja; itt ugyanezt egy 7 hosszúságú belső és 80-szoros külső ciklus adja. Külön kellett definiálni egy S szót a hangváltozók közé, mert az azonos magasságú hangok összefolytak.

NYIRATI LÁSZLÓ



PROGRAMÖNKRITIKA IS...



A *µ*Magazin 1986. júniusi számában megjelent Ékezetes ábécé C64-re című cikkünkben a BASIC kezdőcím eltolását végző parancssor hibás. Helyesen: POKE44,16:POKE43,65:POKE4096+64,0:NEW

Itt a második parancs 43-as címe helyett 45 jelent meg. A cikk többi része helyes, kivéve az 1. program 450-es sorának szövegében lévő helyesírási hibát és a javasolt elhelyezés SHIFT oszlopában szereplő, teljesen felesleges ő betűt.

A cikk megjelenése után elég sűrűn kerestek telefonon, egyrészt a hiba (43 helyett 45), másrészt általában a BASIC kezdőcímet eltolása, illetve a használható nyomtató típusa miatt. Vegyük sorra a kérdéseket! Az első esetben nem mentegetőzöm, én írtam el gépeléskor! Ezért kötelességemnek tartom egy hibátlan, általánosan használható BASIC kezdő- és végcímeltolást végző

1. program

```

0 REM ***** INDITO *****
1
2 X=X+1:IFX=1THENGOSUB40
3 ONXGOTO5,6,10
4 PRINT"MIHA" STOP
5 LOAD"KEP",8,1
6 LOAD"KEP BE",8,1
7
10 PRINT"LOAD"CHR$(34)"BETU"CHR$(34);
11 PRINT";,1"
12 PRINT"LOAD"CHR$(34)"MENU";
13 PRINTCHR$(34);,8"
14 PRINT"LOAD"RUN"
15
20 FORI=0TO3:READR:POKE631+I,R:NEXT
21 POKE198,4
22 DATA19,13,13,13
23
30 POKE43,1:POKE44,16:POKE4096,0
31 POKE55,0:POKE56,120:NEW
32
40 POKE53280,5:POKE53281,5
41 PRINT"*****PROGRAM*****";
42 PRINT"*****B,1 PERC *****";
43 PRINT"*****";
44 RETURN
    
```

parancssor leírását programban is felhasználható módon, kiegészítve még néhány fontos technikai megoldással.

Az ékezetes ábécét használva eddig VC 1525, MPS 801 és EPSON GX típusú nyomtatókon nyomtattunk, ezeken működött. Kaptam olyan telefont is, hogy MPS 802-n az ékezetes betűk helyett a megfelelő grafikus karaktereket nyomtatja ki a program. Ehhez nem tudok hozzászólni, eddig nem nyomtattam 802-esen.

A BASIC terület kezdete a 43, 44 címen lévő érték LO—HI sorrendben. Alaphelyzetben LO=1 HI=8. Azaz a BASIC kezdőcím a

KE = PEEK(43) + 256 * PEEK(44)
alapján számítható ki. A BASIC által hasz-

2. program

```

0 REM ***** MENU *****
1
2 POKE53280,5:POKE53281,5:PRINT"MI"
3 POKE53272,(PEEK(53272)AND240)OR2
4 REM SZINEK, KARAKTERKESZLET ALLITAS
5 RESTORE
6
10 PRINT"*****MENU*****";
11 PRINT"*****";
12 PRINT"*****" REM C = + P
13 PRINT"*****PROGRAM VEGE*****";
14 PRINT"*****NAGV1 PROGRAM*****";
15 PRINT"*****NAGV2 PROGRAM*****";
16 PRINT"*****NAGV3 PROGRAM*****";
17 PRINT"*****NAGV4 PROGRAM*****";
18
20 POKE198,0:AS=""
21 GET#1:IFAS=""THENZ1
22 X=ASC(AS)-48
23 IFX<0ORX>4THENZ0
24
30 ONGOTO50,68,78,88
31
32 PRINT"*****";
33 PRINT"ENNYI VOLT" SVS64738
34
40 PRINT"*****" POKE198,3
41 PRINT"*****"
42 PRINT"*****B,1 PERC*****";
43
44 FORI=0TO3:READR:POKE631+I,R:NEXT
45 DATA19,13,13
46 LO=PEEK(45):HI=PEEK(46):IFLO<2THENH48
47 POKE43,LO-2:POKE44,HI:RETURN
48 POKE43,254+LO:POKE44,HI-1:RETURN
49
50 GOSUB40:PRINT"LOAD"CHR$(34)"NAGV1";
51 PRINTCHR$(34);,8" PRINT"LOAD"RUN"
52 CLR STOP
53
60 GOSUB40:PRINT"LOAD"CHR$(34)"NAGV2";
61 PRINTCHR$(34);,8" PRINT"LOAD"RUN"
62 CLR STOP
63
70 GOSUB40:PRINT"LOAD"CHR$(34)"NAGV3";
71 PRINTCHR$(34);,8" PRINT"LOAD"RUN"
72 CLR STOP
73
80 GOSUB40:PRINT"LOAD"CHR$(34)"NAGV4";
81 PRINTCHR$(34);,8" PRINT"LOAD"RUN"
82 CLR STOP
83
94 END
    
```

nált legmagasabb címet az 55, 56-os címen találjuk szintén LO—HI sorrendben. Alaphelyzetben LO=0, HI=160. A BASIC végcím tehát a
VE = PEEK(55) + 256 * PEEK(56)
összefüggés alapján megállapítható. A BASIC kezdő-, illetve végcímeltolás több okból is szükségesse válhat (új karakterkészlet, gépi rutin, szellemek helye, bitterkések kép tárolása stb.). A BASIC kezdete előtti bájtt tartalma kötelezően nulla! A kezdő- és végcímek megváltozását az interpreter csak akkor veszi tudomásul, ha kiadjuk a NEW vagy CLR utasítást! A feladat általában az, hogy a BASIC kezdő- vagy végcímét akármely átállítani valamilyen CIM-re. A betöltendő LO—HI értékek a CIM = LO + 256 * HI összefüggés alapján számoltatók, ezért a kezdő terület a
POKE43,CIM-256+INT(CIM/256)
POKE44,INT(CIM/256)
POKE CIM-1,0:NEW
a végcím a
POKE55,CIM-256+INT(CIM/256)
POKE56,INT(CIM/256):NEW

3. program

```

0 REM ***** NAGVX *****
1
2 POKE53280,5:POKE53281,5:PRINT"MI"
3
4 REM EZEN KIVUL EGYES INITIALIZALIAS
5
10 PRINT"*****MENU*****";
11 PRINT"*****";
12 PRINT"*****" REM C = + P
13 PRINT"*****VISSZA A FO MENURE*****";
14 PRINT"*****FUNKCIO 1*****";
15 PRINT"*****FUNKCIO 2*****";
16 PRINT"*****";
17 PRINT"*****";
18 PRINT"*****FUNKCIO 9*****";
19
20 POKE198,0:AS=""
21 GET#1:IFAS=""THENZ0
22 X=ASC(AS)-48
23 IFX<0ORX>9THENZ0
24
30 ONXGOTO51,52,53,54,55,56,57,58,59
31
32 POKE53280,5:POKE53281,5:PRINT"MI";
33 RESTORE
34 PRINT"POKE43,1-POKE44,16 CLR";
35 PRINT"LOAD"RUN:POKE198,3
36 FORI=0TO3:READR:POKE631+I,R:NEXT
37 DATA19,13,13
38 REM A 37-ES AZ ELSO DATA SOR ILLI
39 NEW STOP
40
41
51 REM TENNIVALO FUNKCIO 1 VALASZTASKOR
52 REM TENNIVALO FUNKCIO 2 VALASZTASKOR
53 REM TENNIVALO FUNKCIO 3 VALASZTASKOR
54 REM TENNIVALO FUNKCIO 4 VALASZTASKOR
55 REM TENNIVALO FUNKCIO 5 VALASZTASKOR
56 REM TENNIVALO FUNKCIO 6 VALASZTASKOR
57 REM TENNIVALO FUNKCIO 7 VALASZTASKOR
58 REM TENNIVALO FUNKCIO 8 VALASZTASKOR
59 REM TENNIVALO FUNKCIO 9 VALASZTASKOR
60 END
    
```

parancsokkal beállítható. A NEW helyett CLR is használható.

Egy-egy ilyen állítás végrehajtása a program minden egyes felhasználása előtt fardságos, ugyanakkor az is elképzelhető, sőt valószínű, hogy a program a működéséhez az esetleges eltolások mellett igényli más programok jelenlétét is. Ezek parancsokkal történő mindenkor betöltése megnegyezi mind a programozó, mind a felhasználó dolgát.

A fenti problémák automatikus megoldására, a felhasználó helyzetének megkönnyítésére valók a betöltő- (LOADER, BOOT) programok. Megoldásuk többféle lehet; a leggyakrabban előforduló, általában is használható típusokat ismertetem. A cél kettős: egyrészt tájékoztassuk a felhasználót az eseményekről, hogy nyugton maradjon; másrészt végezzük el a szükséges betöltéseket, átállításokat. Ez utóbbiakhoz a felhasználónak semmi köze, csak zavarná. Előtte így tüntetjük el a zajló eseményeket, hogy a neki szánt világos, megnyugtató, a képernyő alján elhelyezett kiírás után egyeztetjük a tinta és a papír színet, így a betöltésekhez, átállításokhoz szükséges kiírások a képernyő tetején a felhasználó előtt láthatatlanok maradnak.

A feladat megoldásához tudnunk kell még, hogy az utóljára lenyomott billentyűk ASCII kódjait tároló billentyűzetpuffer a 631–640 címen van. Azt, hogy pillanatnyilag hány értékes (max. 10) lenyomott billentyű van a pufferben, a 198-as cím tárolja. Ismert még, hogy a LOAD utasítás végrehajtása után a program futása annak első sorától folytatódik, ezért mindenképpen gondoskodnunk kell a betöltőprogram helyes működéséről. Erre szolgál az ON X GOTO utasítás, melynek szemantikáját az esetleges programhibák kiküszöbölésére is felhasználhatjuk. Tudjuk ezenkívül, hogy a LOAD "név", 8,1 utasítással betöltött programok, fájlok a mentési helyükre kerülnek vissza, míg a LOAD "név", 8 utasítás mindig a BASIC terület aktuális elejétől tölti be a programot.

Az előzők mellett találkozhatunk még azzal a problémával is, hogy egyszerre két vagy több BASIC programot szeretnénk elhelyezni a rendelkezésünkre álló BASIC területen, illetve olyan programrendszer irtnak, melynek programjai nem férnek el egyszerre a tárbán. Ekkor feltétlenül szükséges egy olyan rövid menüprogram megírása is, mely biztosítja az egyes fő BASIC programok tetszőleges sorrendben történő behívását, indítását. Célnak az, hogy a felhasználó feladata az általunk írt programrendszer futtatásakor mindössze annyi legyen, hogy behívja és elindítja a betöltőprogramot. A többi automatikusan menjen anélkül, hogy a felhasználó arról tudomást szerezne, beavatkozna. A technikai részletek pedig végképp nem tartoznak rá.

Tegyük fel, hogy irtnak egy olyan programrendszer, mely a NAGY1, NAGY2, NAGY3, NAGY4 nevű BASIC programok mellett tartalmaz egy BETU nevű ékezetes ábécé programot (helye 2 k–4 k között), egy KEP nevű, KOALA PAINT programmal készített képet (helye 30 k–40 k között), valamint ennek előhívását és a szükséges oda-vissza kapcsolásokat végző KEP BE nevű gépi kódú programot (helye a kazettapuffer). KEP és KEP BE valahol a

NAGYX programokban kerülnek felhasználásra. NAGYX programjaink olyan hosszúak, hogy kettő együtt már nem fér el a tárbán, viszont mindegyik mellett van még hely egy rövid BASIC program számára. A NAGYX programok négy különböző feladatot oldanak meg, a felhasználókat ezeket egy-egy futtatás alkalmával tetszőleges sorrendben, tetszőleges sokszor hívhatják.

Feladatunk: írjuk meg az INDITO nevű betöltőprogramot, mely a kezdő- és végcím átállításra mellett betölti a helyükre a BETU, KEP, KEP BE programokat, valamint behívja és elindítja az ugyancsak most megírandó MENU programot, mely a felhasználó választása alapján behívja és elindítja a NAGYX programok valamelyikét. A felhasználó NAGYX programok mindegyikéből visszatérhet a MENU programra. Ezt a visszatérést végző részt is bemutatom, a programrészletet közlöm.

Először vegyük szemügyre az INDITO program szerkezetét! Első feladata, hogy tájékoztassa a felhasználót az eseményekről. Ezután helyükre töltheti a KEP, KEP BE programokat. BETU betöltésével viszont már baj van, hisz éppen INDITO helyére kerül! Betöltése előtt feltétlenül át kell állítani a BASIC terület kezdetét, aminek végrehajtása a NEW miatt megöli INDITO-t! KEP is csak azért tölthető be a végcím átállítás előtt, mert INDITO rövid és nem használ szöveg típusú változókat. Ha a kezdőcímet átállítottuk, akkor átállítjuk a végcímét is.

Hogyan töltjük hát be BETU és MENU programokat és indítjuk el az utóbbit? Úgy, hogy az átállítások előtt a képernyő tetejét kezdődően a megfelelő sorokba PRINT-tel kiírjuk a betöltéseket és az indítást végző parancsokat, majd a billentyűpuffert feltöltjük a megfelelő vezérlő karakterekkel (HOME, RETURN), ugyanis ezek a programfutás végén a pufferből automatikusan a képernyőre kerülnek, biztosítva a parancsok végrehajtását. A felhasználó előtt mindez rejtve marad, ha a papír és a tinta színe megegyezik. Próbá alatt azonban jobb, ha látjuk, mi történik (lásd 1. program). Még annyit: CHR\$(34) az idézőjel, 19 a HOME, 13 a RETURN kódja.

MENU lesz az a program, amely a mindenkor behívott NAGYX mellett még elfér a tárbán. Lényegében — a felhasználó szempontjából — egy menüből áll, ahol a program vége választás mellett a mindenkor NAGYX behívása és indítása választható. Emellett mindössze annyit tud még, hogy INDITO-hoz hasonlóan a BASIC kezdőcím további átállításával eltünteti, de nem semmisíti meg önmagát, mielőtt behívna és indítaná NAGYX-et, valamint átállítja a karakterkészletet az ékezetes betűkre. Közben természetesen értesíti a felhasználót a programbetöltés tényéről. Az eljárás logikája ugyanaz, mint INDITO-é az átkapcsolások végrehajtásakor, illetve BETU és MENU betöltésekor, indításakor.

A különbség mindössze annyi, hogy ott (INDITO írásakor) előre tudtuk az új kezdőcím értékét, míg most csak akkor tudnánk, ha MENU már kész lenne. A probléma megoldható úgy is, hogy elkészítem MENU-t, a végcím helyét kipontozom, majd a kész program után PRINTPEEK-kel lekérdezem a címet és a pontok helyére beírom. De az így elkészített program a leg-

kisebb változtatás után már rosszul működik. Sokkal biztosabb, ha a feladatot általánosan oldjuk meg. Így az esetleges későbbi változtatások mellett is működik, használható. Tudjuk már, hogy a BASIC kezdőcím előtti bajt értéke kötelezően nulla. Ez a felvétel most automatikusan teljesül, hiszen az utolsó BASIC sor vége úgyis 0, amit a program végét jelző két darab 0 követ.

A BASIC program végét a 45, 46-os címenek tárolja a gép LO—HI sorrendben. MENU tulajdonképpen NAGYX betöltése és futtatása előtt lekérdezi e címekről saját végét, majd a BASIC kezdetét ennél két bajtál előbbre állítja be, törölve a program végét jelző két nullát. Az utolsó sor végét jelző nulla megmarad. Csak ezután tölti be és indítja a választott NAGYX-et. Tulajdonképpen ezt a technikát alkalmazzuk programok összefűzésekor is, azzal a különbséggel, hogy ott az utolsó program betöltése után visszaállítjuk a BASIC első program előtti kezdetét, és az így egybefűzött programokat egy programként mentjük lemezre. Visszatérve MENU-re, a felsoroltak mellett még azt is tudja, hogy ha a lemezen nincs meg véletlenül a választott NAGYX, akkor a lemezegyező piros lámpája villog ugyan, de a program nem áll le, a képernyő újra a menü lesz látható. A lámpa villogása egy, a programba épített, inicializáló utasítással kivédhető. Részletebben lásd a 2. programot.

Mi a leirt technika előnye? Ha a felhasználó egy-egy futtatás alkalmával sűrűn hívja a különböző NAGYX-eket, akkor nem kell várnia MENU újabb betöltésére, az átkapcsolás egy pillanat műve.

Igy már érthető, hogy NAGYX-ek menüjében miért legyen visszatérési lehetőség a MENU programra. A visszatérési, programvege-lehetőség általában az aktuális menü 0 választási pontja, hiszen így nehezen téveszthető, lásd a billentyűzetet. NAGYX programoknak csak a menüjét és a MENU programra való visszatérést végző részt írtam meg (3. program).

Mindhárom programban zöld a keret és a papír színe, fehér a felhasználónak szóló üzenet és a menü szövege, zöld viszont a tinta színe az átállító, betöltő, futtató parancsok kiírásakor. Ismét hangsúlyozom, hogy a próbák alatt ajánlatos a tinta színét végig fehéren tartani! Futás közben a képernyő figyelmes tanulmányozása választ adhat a még felmerülő kérdésekre. INDITO csak akkor futtatható, ha valóban létezik BETU, KEP, KEP BE.

ÉNEKES FERENC

Kedves Olvasóink!

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

Szövegszerkesztő és elemző



A program 48 k-s géphez készült, ugyan- is egymástól függetlenül 40 képernyőt kezelhetünk (karakteres felbontás), de fut 16 k-s gépen is, amennyiben csak egy képernyőre van szükségünk s többet nem akarunk kezelni.

A program indítás után a nevét írja ki, gombnyomás után hozzáfoghatunk a szerkesztendő szöveg beírásához. Használhatóak a kis- és nagybetűk, számok, valamint a SYMBOL SHIFT-tel elérhető jelek, például írásjelek. A karaktereket keret alakú, villogó kurzor követi.

Ha a sor végére érünk, a következő betű automatikusan új sorba kerül. Az ENTER gomb lenyomásával is kezdhetünk új sort. Ha mellétünk, a DELETE gomb (CAPS SHIFT + 0) leütésével törölhetünk egyet vissza. Ezenkívül a kurzort bárhova mozgathatjuk a képernyőn a nyilgombok segítségével (CAPS SHIFT + 5, + 6, + 7, + 8). Rámehetünk vele egy karakterre, s az általam kívánt betűre cserélhetjük az adott billentyű leütésével. A CAPS LOCK segítségével (CAPS SHIFT + 2) a kurzor a bal felső sarokba ugrik (Commodore-on: CURSOR HOME). Ha egy hosszabb sort gépeltünk be, de az elején kihagytunk egy szót, akkor a kérdéses helyre megyünk, s a SYMBOL SHIFT, CAPS SHIFT együttes lenyomásával a kurzortól jobbra elhelyezkedő karaktereket jobbra tolhatjuk, helyet szorítva a kimaradt szónak. Így tehát probléma nélkül szűrhatunk be szavakat. Ha viszont sok SPACE-t tettünk egy szó elé, vagy a sor túlságosan jobbra csúszott, akkor a GRAPHICS gombbal (CAPS SHIFT + 9) balra húzhatjuk a kurzortól jobbra elhelyezkedő részeket.

Egyébként a programban minden betű leütését hangjelzés követi. Van egy különleges funkció is, amelyet a TURE VIDEO gombbal (CAPS SHIFT + 3) kapcsolhatunk be, és az INV VIDEO-val (CAPS SHIFT + 4) kapcsolhatunk ki. Ha ezt a funkciót használjuk, akkor a keret nem fekete, hanem kék lesz, s egy gomb lenyomásáig vált vissza feketére. Ha kiírta a karaktert, ismét kék lesz, s várja a következő betűt. Mindez a másodperc tört része alatt történik, de így csak a billentyűzetre kell figyelni, a keret változásaiból érzékelhetjük a

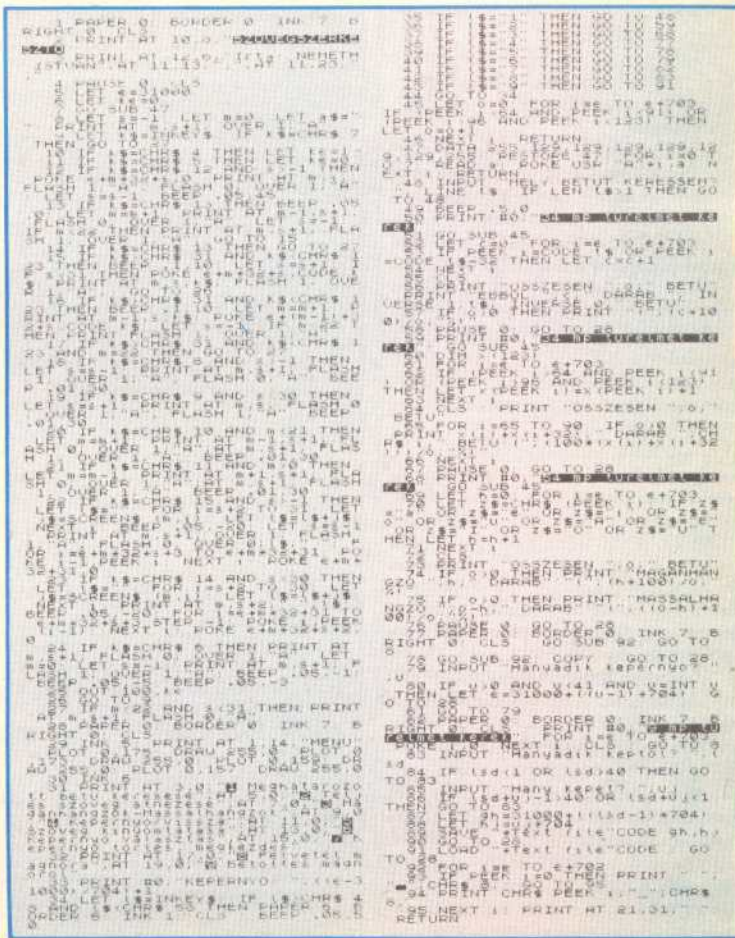
megfelelő tempót. A program a beírt szöveget a 31 000. memóriacímre tárolja karakterenként. Szerkesztés közben is közvetlenül ide írja be a friss adatokat.

Ha a szerkesztést befejeztük, az EDIT gombbal (CAPS SHIFT + 1) szállhatunk ki, s tekinthetjük meg a MENÜ-t. Itt az utasítások arra a képre vonatkoznak, amelynek sorszáma alul is megjelenik. Az 1-es gomb lenyomására a gép kikeresi a kívánt betűt, kiírja, mennyi van belőle összesen, s ez az egészek hány százaléka. A 2-es gombbal a teljes szöveg elemzését kérhetjük. Várakozás után kiíródik minden betű darabszáma és százalékos aránya. A 3-as gombbal a magánhangzók és mássalhangzók arányát tudhatjuk meg. A programmal tehát nyelvtudási elemzéseket is végezhetünk. A 4-es gombbal visszatérhetünk az

aktuális képhez, folytathatjuk a szerkesztést. Az 5-ös gombbal printerre kinyomtatható a szöveg. A 6-os gomb lenyomásával megválasztjuk, hányadik képernyőn akarunk dolgozni, 1–40-ig. A 7-es gombbal egy képernyő tartalmát törölhetjük. A 8-as gombbal felvehetjük tetszőleges számú képernyő tartalmát kazettára. Először az első felveendő képet kell megadni, majd azt, hogy összesen hányat. Például, ha az első négy képet kívánjuk rögzíteni, akkor a beadandó számok: 1;4. A 9-es gombbal betölthetjük a felvett képeket.

A program beírásánál ügyeljünk arra, hogy a 12–24-es soroknál az idézőjelben lévő A betűk grafikusak.

NÉMETH ISTVÁN
ELTE Ságvári Endre Gyakorló Iskola



SZELLEMGRAFIKA I.

A szellem (sprite) grafika, és a vele kapcsolatos műveletek a C64 speciális lehetőségei közé tartoznak. Programkéntesen egyszerre maximum 8 szellem jeleníthető meg a képernyőn, annak bármilyen üzemmódban. Ezt a számot csak a megszakítások menetének biztos ismeretében, annak felhasználásával léphetjük át, kihasználva az optikai csalódást.

Egy szellem BASIC-ből történő felhasználásához az alábbi műveletek szükségesek:

- alakjának meghatározása, az azt előállító 63 bájttól létrehozása, betöltése;
- a helymutató beállítása;
- a normál vagy többszínű üzemmód kiválasztása;
- a szín meghatározása, prioritások beállítása;
- nagytítás beállítása;
- a kezdőhelyzet beállítása, bekapcsolás;
- az előbbiektől folyamatos változtatása, esetleg a megszakítások között, ütközésvizsgálat.

Vegyük sorra az egyes műveleteket! A szellem alakjának megtervezésére több módszer ismert, a „kockás” papírtól a SPRITE EDITOR programokig. Egy szellem alakját 3 × 21 bájttal határozza meg, ahol minden bekapcsolt pontnak 1-es érték felel meg, azaz egy szellem 24 × 21 képpontból áll. Ezt a 63 bájtot kell sorfolytonosan bevenni a memória megfelelő helyére. Mivel a szellemeket a VIC-II chip kezeli, ezért azok adatait az általa használt 16 k-s szelet valamely 64-gyel osztható címétől kezdődően helyezzzük el.

Az egyszerre használható 8 szellem — sorszámaik 0–7 — alakját meghatározó adatok kezdetére egy-egy mutató mutat, melyek értékei rendre a 2040–2047 címeken találhatók. A VIC-II a szellem adatainak kezdőcímét e mutatók és a szeletsorszám (0–3) alapján a cím = 16384 + szeletsorszám + 64 × mutató összefüggés alapján határozza meg. Természetesen nem használható minden 64 bájtos lap! A 11, 13, 14, 15-ös sorszámu helyek minden további nélkül használhatók. A 32–63 sorszámuak a BASIC kezdőcím 4 k-ra, a 128–255 helyek a 16 k-ra történő eltolás mellett érhetőek el.

A szellem — a bekapcsolt pontok — színe a 16 szín bármelyike lehet. A kikapcsolt pontokban azt látjuk, ami a szellem mögött egyébként a képernyőn van. A szellemek színregiszterei az 53287–53294 címeken találhatók. A többszínű üzemmód kiválasztása az 53276 címre írt érték n. bitjének magasra állítását jelenti. Ekkor az n. sorszámu

szellem többszínű üzemmódban jelenik meg. A többszínű üzemmód azonos a karaktergrafikai megfelelőjével, a szellem sorokénti 24 bitje helyett 12 bitpár jelenik meg, melyek színei:

- 00 — átlátszó,
- 01 — többszín-regiszter 1,
- 10 — színregiszter,
- 11 — többszín-regiszter 2.

A többszín-regiszter 1 az 53285, a többszín-regiszter 2 az 53286 címeken található.

Két szellem találkozáskor a kisebb sorszámu a nagyobb sorszámu előtt jelenik meg, azaz a kisebb sorszámu eltakarja a nagyobbat. Külön intézkedni csak a szellem és a képernyő kitöltött részeinek egymás-

hoz viszonyított helyzetére vonatkozóan kell. Ha azt akarjuk, hogy az n. szellem eltakarja a háttérrel, akkor az 53275-ös cím n. bitjét állítsuk ALACSONYRA! Ez a kezdő érték is.

A szellemek megjeleníthetők normál méretben, vagy X, Y, illetve mindkét irányban kétszeresre nagyítva. A nagyítási pont a szellem bal felső sarka. A vízszintes, X irányú nagyítás az 53277-es, a függőleges, Y irányú nagyítás az 53271-es címre írt értékkel szabályozható. Ha a címre írt érték n. bitje magas, akkor az n. szellem az adott irányban kétszeresre nagyítva jelenik meg. A felbontás nem lesz finomabb.

A szellem helyének megadásakor mindig a bal felső sarkának koordinátáit adjuk

```

180 PER *****
191 PER 4 *****
192 PER *****
193 PER *****
194 PER *****
195 PER *****
196 PER *****
197 PER *****
198 PER *****
199 PER *****
200 PER *****
201 PER *****
202 PER *****
203 PER *****
204 PER *****
205 PER *****
206 PER *****
207 PER *****
208 PER *****
209 PER *****
210 PER *****
211 PER *****
212 PER *****
213 PER *****
214 PER *****
215 PER *****
216 PER *****
217 PER *****
218 PER *****
219 PER *****
220 PER *****
221 PER *****
222 PER *****
223 PER *****
224 PER *****
225 PER *****
226 PER *****
227 PER *****
228 PER *****
229 PER *****
230 PER *****
231 PER *****
232 PER *****
233 PER *****
234 PER *****
235 PER *****
236 PER *****
237 PER *****
238 PER *****
239 PER *****
240 PER *****
241 PER *****
242 PER *****
243 PER *****
244 PER *****
245 PER *****
246 PER *****
247 PER *****
248 PER *****
249 PER *****
250 PER *****
251 PER *****
252 PER *****
253 PER *****
254 PER *****
255 PER *****
256 PER *****
257 PER *****
258 PER *****
259 PER *****
260 PER *****
261 PER *****
262 PER *****
263 PER *****
264 PER *****
265 PER *****
266 PER *****
267 PER *****
268 PER *****
269 PER *****
270 PER *****
271 PER *****
272 PER *****
273 PER *****
274 PER *****
275 PER *****
276 PER *****
277 PER *****
278 PER *****
279 PER *****
280 PER *****
281 PER *****
282 PER *****
283 PER *****
284 PER *****
285 PER *****
286 PER *****
287 PER *****
288 PER *****
289 PER *****
290 PER *****
291 PER *****
292 PER *****
293 PER *****
294 PER *****
295 PER *****
296 PER *****
297 PER *****
298 PER *****
299 PER *****
300 PER *****
301 PER *****
302 PER *****
303 PER *****
304 PER *****
305 PER *****
306 PER *****
307 PER *****
308 PER *****
309 PER *****
310 PER *****
311 PER *****
312 PER *****
313 PER *****
314 PER *****
315 PER *****
316 PER *****
317 PER *****
318 PER *****
319 PER *****
320 PER *****
321 PER *****
322 PER *****
323 PER *****
324 PER *****
325 PER *****
326 PER *****
327 PER *****
328 PER *****
329 PER *****
330 PER *****
331 PER *****
332 PER *****
333 PER *****
334 PER *****
335 PER *****
336 PER *****
337 PER *****
338 PER *****
339 PER *****
340 PER *****
341 PER *****
342 PER *****
343 PER *****
344 PER *****
345 PER *****
346 PER *****
347 PER *****
348 PER *****
349 PER *****
350 PER *****
351 PER *****
352 PER *****
353 PER *****
354 PER *****
355 PER *****
356 PER *****
357 PER *****
358 PER *****
359 PER *****
360 PER *****
361 PER *****
362 PER *****
363 PER *****
364 PER *****
365 PER *****
366 PER *****
367 PER *****
368 PER *****
369 PER *****
370 PER *****
371 PER *****
372 PER *****
373 PER *****
374 PER *****
375 PER *****
376 PER *****
377 PER *****
378 PER *****
379 PER *****
380 PER *****
381 PER *****
382 PER *****
383 PER *****
384 PER *****
385 PER *****
386 PER *****
387 PER *****
388 PER *****
389 PER *****
390 PER *****
391 PER *****
392 PER *****
393 PER *****
394 PER *****
395 PER *****
396 PER *****
397 PER *****
398 PER *****
399 PER *****
400 PER *****
401 PER *****
402 PER *****
403 PER *****
404 PER *****
405 PER *****
406 PER *****
407 PER *****
408 PER *****
409 PER *****
410 PER *****
411 PER *****
412 PER *****
413 PER *****
414 PER *****
415 PER *****
416 PER *****
417 PER *****
418 PER *****
419 PER *****
420 PER *****
421 PER *****
422 PER *****
423 PER *****
424 PER *****
425 PER *****
426 PER *****
427 PER *****
428 PER *****
429 PER *****
430 PER *****
431 PER *****
432 PER *****
433 PER *****
434 PER *****
435 PER *****
436 PER *****
437 PER *****
438 PER *****
439 PER *****
440 PER *****
441 PER *****
442 PER *****
443 PER *****
444 PER *****
445 PER *****
446 PER *****
447 PER *****
448 PER *****
449 PER *****
450 PER *****
451 PER *****
452 PER *****
453 PER *****
454 PER *****
455 PER *****
456 PER *****
457 PER *****
458 PER *****
459 PER *****
460 PER *****
461 PER *****
462 PER *****
463 PER *****
464 PER *****
465 PER *****
466 PER *****
467 PER *****
468 PER *****
469 PER *****
470 PER *****
471 PER *****
472 PER *****
473 PER *****
474 PER *****
475 PER *****
476 PER *****
477 PER *****
478 PER *****
479 PER *****
480 PER *****
481 PER *****
482 PER *****
483 PER *****
484 PER *****
485 PER *****
486 PER *****
487 PER *****
488 PER *****
489 PER *****
490 PER *****
491 PER *****
492 PER *****
493 PER *****
494 PER *****
495 PER *****
496 PER *****
497 PER *****
498 PER *****
499 PER *****
500 PER *****
501 PER *****
502 PER *****
503 PER *****
504 PER *****
505 PER *****
506 PER *****
507 PER *****
508 PER *****
509 PER *****
510 PER *****
511 PER *****
512 PER *****
513 PER *****
514 PER *****
515 PER *****
516 PER *****
517 PER *****
518 PER *****
519 PER *****
520 PER *****
521 PER *****
522 PER *****
523 PER *****
524 PER *****
525 PER *****
526 PER *****
527 PER *****
528 PER *****
529 PER *****
530 PER *****
531 PER *****
532 PER *****
533 PER *****
534 PER *****
535 PER *****
536 PER *****
537 PER *****
538 PER *****
539 PER *****
540 PER *****
541 PER *****
542 PER *****
543 PER *****
544 PER *****
545 PER *****
546 PER *****
547 PER *****
548 PER *****
549 PER *****
550 PER *****
551 PER *****
552 PER *****
553 PER *****
554 PER *****
555 PER *****
556 PER *****
557 PER *****
558 PER *****
559 PER *****
560 PER *****
561 PER *****
562 PER *****
563 PER *****
564 PER *****
565 PER *****
566 PER *****
567 PER *****
568 PER *****
569 PER *****
570 PER *****
571 PER *****
572 PER *****
573 PER *****
574 PER *****
575 PER *****
576 PER *****
577 PER *****
578 PER *****
579 PER *****
580 PER *****
581 PER *****
582 PER *****
583 PER *****
584 PER *****
585 PER *****
586 PER *****
587 PER *****
588 PER *****
589 PER *****
590 PER *****
591 PER *****
592 PER *****
593 PER *****
594 PER *****
595 PER *****
596 PER *****
597 PER *****
598 PER *****
599 PER *****
600 PER *****
601 PER *****
602 PER *****
603 PER *****
604 PER *****
605 PER *****
606 PER *****
607 PER *****
608 PER *****
609 PER *****
610 PER *****
611 PER *****
612 PER *****
613 PER *****
614 PER *****
615 PER *****
616 PER *****
617 PER *****
618 PER *****
619 PER *****
620 PER *****
621 PER *****
622 PER *****
623 PER *****
624 PER *****
625 PER *****
626 PER *****
627 PER *****
628 PER *****
629 PER *****
630 PER *****
631 PER *****
632 PER *****
633 PER *****
634 PER *****
635 PER *****
636 PER *****
637 PER *****
638 PER *****
639 PER *****
640 PER *****
641 PER *****
642 PER *****
643 PER *****
644 PER *****
645 PER *****
646 PER *****
647 PER *****
648 PER *****
649 PER *****
650 PER *****
651 PER *****
652 PER *****
653 PER *****
654 PER *****
655 PER *****
656 PER *****
657 PER *****
658 PER *****
659 PER *****
660 PER *****
661 PER *****
662 PER *****
663 PER *****
664 PER *****
665 PER *****
666 PER *****
667 PER *****
668 PER *****
669 PER *****
670 PER *****
671 PER *****
672 PER *****
673 PER *****
674 PER *****
675 PER *****
676 PER *****
677 PER *****
678 PER *****
679 PER *****
680 PER *****
681 PER *****
682 PER *****
683 PER *****
684 PER *****
685 PER *****
686 PER *****
687 PER *****
688 PER *****
689 PER *****
690 PER *****
691 PER *****
692 PER *****
693 PER *****
694 PER *****
695 PER *****
696 PER *****
697 PER *****
698 PER *****
699 PER *****
700 PER *****
701 PER *****
702 PER *****
703 PER *****
704 PER *****
705 PER *****
706 PER *****
707 PER *****
708 PER *****
709 PER *****
710 PER *****
711 PER *****
712 PER *****
713 PER *****
714 PER *****
715 PER *****
716 PER *****
717 PER *****
718 PER *****
719 PER *****
720 PER *****
721 PER *****
722 PER *****
723 PER *****
724 PER *****
725 PER *****
726 PER *****
727 PER *****
728 PER *****
729 PER *****
730 PER *****
731 PER *****
732 PER *****
733 PER *****
734 PER *****
735 PER *****
736 PER *****
737 PER *****
738 PER *****
739 PER *****
740 PER *****
741 PER *****
742 PER *****
743 PER *****
744 PER *****
745 PER *****
746 PER *****
747 PER *****
748 PER *****
749 PER *****
750 PER *****
751 PER *****
752 PER *****
753 PER *****
754 PER *****
755 PER *****
756 PER *****
757 PER *****
758 PER *****
759 PER *****
760 PER *****
761 PER *****
762 PER *****
763 PER *****
764 PER *****
765 PER *****
766 PER *****
767 PER *****
768 PER *****
769 PER *****
770 PER *****
771 PER *****
772 PER *****
773 PER *****
774 PER *****
775 PER *****
776 PER *****
777 PER *****
778 PER *****
779 PER *****
780 PER *****
781 PER *****
782 PER *****
783 PER *****
784 PER *****
785 PER *****
786 PER *****
787 PER *****
788 PER *****
789 PER *****
790 PER *****
791 PER *****
792 PER *****
793 PER *****
794 PER *****
795 PER *****
796 PER *****
797 PER *****
798 PER *****
799 PER *****
800 PER *****
801 PER *****
802 PER *****
803 PER *****
804 PER *****
805 PER *****
806 PER *****
807 PER *****
808 PER *****
809 PER *****
810 PER *****
811 PER *****
812 PER *****
813 PER *****
814 PER *****
815 PER *****
816 PER *****
817 PER *****
818 PER *****
819 PER *****
820 PER *****
821 PER *****
822 PER *****
823 PER *****
824 PER *****
825 PER *****
826 PER *****
827 PER *****
828 PER *****
829 PER *****
830 PER *****
831 PER *****
832 PER *****
833 PER *****
834 PER *****
835 PER *****
836 PER *****
837 PER *****
838 PER *****
839 PER *****
840 PER *****
841 PER *****
842 PER *****
843 PER *****
844 PER *****
845 PER *****
846 PER *****
847 PER *****
848 PER *****
849 PER *****
850 PER *****
851 PER *****
852 PER *****
853 PER *****
854 PER *****
855 PER *****
856 PER *****
857 PER *****
858 PER *****
859 PER *****
860 PER *****
861 PER *****
862 PER *****
863 PER *****
864 PER *****
865 PER *****
866 PER *****
867 PER *****
868 PER *****
869 PER *****
870 PER *****
871 PER *****
872 PER *****
873 PER *****
874 PER *****
875 PER *****
876 PER *****
877 PER *****
878 PER *****
879 PER *****
880 PER *****
881 PER *****
882 PER *****
883 PER *****
884 PER *****
885 PER *****
886 PER *****
887 PER *****
888 PER *****
889 PER *****
890 PER *****
891 PER *****
892 PER *****
893 PER *****
894 PER *****
895 PER *****
896 PER *****
897 PER *****
898 PER *****
899 PER *****
900 PER *****
901 PER *****
902 PER *****
903 PER *****
904 PER *****
905 PER *****
906 PER *****
907 PER *****
908 PER *****
909 PER *****
910 PER *****
911 PER *****
912 PER *****
913 PER *****
914 PER *****
915 PER *****
916 PER *****
917 PER *****
918 PER *****
919 PER *****
920 PER *****
921 PER *****
922 PER *****
923 PER *****
924 PER *****
925 PER *****
926 PER *****
927 PER *****
928 PER *****
929 PER *****
930 PER *****
931 PER *****
932 PER *****
933 PER *****
934 PER *****
935 PER *****
936 PER *****
937 PER *****
938 PER *****
939 PER *****
940 PER *****
941 PER *****
942 PER *****
943 PER *****
944 PER *****
945 PER *****
946 PER *****
947 PER *****
948 PER *****
949 PER *****
950 PER *****
951 PER *****
952 PER *****
953 PER *****
954 PER *****
955 PER *****
956 PER *****
957 PER *****
958 PER *****
959 PER *****
960 PER *****
961 PER *****
962 PER *****
963 PER *****
964 PER *****
965 PER *****
966 PER *****
967 PER *****
968 PER *****
969 PER *****
970 PER *****
971 PER *****
972 PER *****
973 PER *****
974 PER *****
975 PER *****
976 PER *****
977 PER *****
978 PER *****
979 PER *****
980 PER *****
981 PER *****
982 PER *****
983 PER *****
984 PER *****
985 PER *****
986 PER *****
987 PER *****
988 PER *****
989 PER *****
990 PER *****
991 PER *****
992 PER *****
993 PER *****
994 PER *****
995 PER *****
996 PER *****
997 PER *****
998 PER *****
999 PER *****
1000 PER *****

```



Villogó grafika

A program a grafikus karakterek inverzét állítja elő, ezáltal villogtatja a képernyőt. A programot megadjuk BASIC-ben DATA sorokba építve és assemblerben is. A BASIC programnak van egy egyszerű része, amellyel kipróbálhatjuk a szubrutint.

BARANYI ZSOLT

A BASIC program

```
5 REM BETÖLTÉS
10 CLS
20 FOR C = 28672 TO 28708
30 READ A : POKE C,A : NEXT C
40 REM DEMO
50 X = RND(127) : Y = RND(47)
60 SET(X,Y)
70 IF INKEY = "E" THEN 80 ELSE 50
80 POKE 16526,0:POKE 16527,12
90 FOR T = 0 TO 50 : REM VILLANÁSOK SZÁMA
100 Z =USR(0)
110 FOR C = 0 TO 120:NEXT C
120 REM SZÜNET
130 NEXT T
140 REM A GÉPI KÓDÚ RÉSZ
150 DATA 33,0,60,6,0,14,4,126,254,32
160 DATA 40,16,254,128,56,4,254,192
170 DATA 56,12,119,35,16,239,13,32,236
180 DATA 201,62,191,24,244,47,198,64,24
190 DATA 239
```

A program ASSEMBLER-ben

```
LD HL,3C00H; A KÉPERNYŐ
ELSŐ PONTJA
LD B,0
LD C,4
CIKL:LD A,(HL)
CP 32; SPACE?
JR Z,FEHER; HA IGEN, UGORJ
FEHERRE
CP 128; KARAKTER?
JR C,KI; HA IGEN, UGORJ KI-
RE
CP 192; GRAF. JEL?
JR C,INVERZ; HA IGEN,
UGORJ INVERZBE
KI:LD (HL),A
INC HL
DJNZ CIKL
DEC C
JR NZ,CIKL
RET
FEHER:LD A,191
JR KI
INVERZ:CPL
ADD A,64; A 6. BIT NULLÁZÁ-
SÁ
JR KI
```

meg. Elhelyezésük az úgynevezett szellem-
térben történik, mely egy 512×256-os tó-
ruszfelület, lényegesen nagyobb, mint a
320×200-as képernyő. A koordináták érté-
kei a szokásos módon, 0-val kezdődnek.
A tóruszfelület azt jelenti, hogy a szellem-
tér alja-teteje, bal és jobb oldala összeér.
A szellemtér képernyőhöz viszonyított
helyzete a következő: a szellemtér (24;50)
koordinátájú pontja a képernyő bal felső
sarka a (0;0) koordinátákkal. Mivel a szel-
lem helyének X koordinátája 0—511 közé
eshet, tárolására nem elegendő egy bájt,
egy további, 256 értékű bit is szükséges; ezt
a bitet nevezzük MSB-ne'. Ha az n. szel-
lem MSB bitje magas, akkor az X koordi-
nátát meghatározó bájt értékéhez 256 adó-
dik hozzá. A szellemek MSB bitjét megha-
tározó bájt címe 53264. Ennek n. bitje az n.
szellem MSB-jét kapcsolja be-ki. Az 53248,
53249, ..., 53262, 53263 címek rendre az n.
szellem helyzetének X, Y koordinátáit ha-
tározzák meg.

Az n. szellem bekapcsolása, megjeleníté-
se az 53269 címen levő érték n. bitjének
magasra állításával érhető el. Hogy a szel-
lem a képernyőn látszik-e vagy sem, azt ter-
mészetesen a pillanattani helyzetét megadó
X, Y koordinátapár dönti el.

Ahhoz, hogy a megadott álló szellem
mozogjon, legalább a helyzetét meghatá-
rozó X, Y koordinátapár változtatása szüksé-
ges. Egy figura mozgásához ezenkívül
legtöbbször a mozgást jellemző fázisokat
változtatjuk úgy, hogy azokat egy szel-
lem jelenti meg, és ezeket kapcsoljuk ki-
be. A VIC—II biztosítja azt a lehetőséget
is, hogy megvizsgáljuk két szellem, illetve
egy szellem és a háttér ütközésének megtör-
téntét is. Ütközésen mindig a bekapcsolt
pontok ütközése érthető. Két szellem ter-
mészetesen akkor is ütközhet, ha nincsenek
a látható képernyőn. Az 53278 című regisz-
ter két szellem, az 53279 a szellem-háttér-
ütközéseket regisztrálja. Ütközéskor az első-
nél mindig kettő, az utóbbinál egy bit érté-
ke lesz magas. Olvasásukkal (PEEK) az üt-
kőzés megtörténtét ellenőrizhetjük. Vigyá-
zat, olvasáskor mindkét regiszter törlődik,
ezért értéküket a további vizsgálathoz köte-
lező elmenteni! Többszínű üzemmódban a
01 bitkombinációjú pontok ütközés szem-
pontjából áttetszőnek számítanak. Az üt-
közések vizsgálata a program elágaztatását
szolgálja.

Végül az X. cím N-edik bitjének magasra
állítását a POKE X,PEEK(X) OR 2^N, ala-
csonyra állítását a POKE X,PEEK(X)
AND (255-2^N) utasítás végzi el. A szelle-
mek BASIC-ből történő felhasználása elég
körülményes, sokszor a gyorsaság sem
megfelelő. Lehetőségeik valódi kihasználá-
sát csak gépi kódban érhetjük el.

A program címe Menő Manó. Begépelé-
se előtt toljuk el a BASIC terület kezdetét 4
k-ra az ismert módon. A program egy szel-
lemet jelenít meg, majd mozgat hat fázis-
ban többszínű karaktergrafikai támogatás-
sal. Felépítése:

103 Kiindulási értékek beállítása
helyben részletezve.

105—107 A szellem mozgatása: két rác-
spont jobbra, időzítés, fázisváltás.

109—110 A szellem kétszer balra, és ⓐ
karakter átirása.

112—113 A szellem helyben mozog fázis-
váltással.

115—117 Fázisváltás, szellem és karakter-
kép mozgatása balra.

119—121 A karakterkép fázisváltásának
kivárása, kezdet beállítása, visz-
sza az elejére.

126 F0: 0. szellem, F1: 2 k eleje, F2:
VIC—II eleje,
F3: szellembetöltési cím a 37-es
blokkba.

127 A szellem adatainak nullázása

128 ⓐ alsó öt bájta lesz.

129 ⓐ felső három bájtnak négy
fázisa.

130 Az első fázis betöltése.

131 Ha a szellem betöltve, akkor
nem tölt újra.

133—136 DATA sorok betöltése, ellenőrző
összeg.

138 0. szellem a 37-es blokkban, szel-
lem többszínű üzemmódban.

139—140 0. szellem színe fekete, többszín-
regiszterek világos piros és fehér,
Y koordináta.

141—142 X koordináta, MSB.

144—146 Egy sor már új ⓐ .

147—149 Karakter többszín színei, több-
színű üzemmódban be, 0 szeletra át-
kapcsolás.

152—153 ⓐ új adatai.

155—202 A szellem hat fázisának adatai,
fázisonként nyolc sor.

207—211 A szellem jobbra mozgatása,
MSB kapcsolása.

213—217 A szellem balra mozgatása, MSB
kapcsolása.

219—221 Karakterkép fázisváltása.

223—224 A szellem fázisváltása.

A szellem egy közismert játék főszereplője.

FÖLDI ENDRE—ÉNEKES FERENC



Zeneprogram

Ezzel a programmal dallamokat csalszunk ki számítógépünkben. A dallam öt szakaszból áll. Az adatok bekérések az első szám a hosszúságot, a második pedig a magasságot jelenti. Májris játszhatunk.

MUTSCHLER PÉTER

```

0 REM ZENEGENERALAS PRIMORA
5 REM MUTSCHLER PETER 1986
10 CLS
20 PRINT CHR$(2)
30 PRINT " ZENEGENERALAS"
40 PRINT CHR$(1)
50 PRINT:PRINT
60 PRINT "EZZEL A PROGRAMMAL DALLAMOKAT CSALHATSZ KI GEPEDBOL."
65 PRINT "HA LEHET 1500-NAL NAGYOBB SZAMOT NE IRJ BE!"
70 PRINT:PRINT"ELSO SZAM A HOSSZUSAG, A MASODIK A MAGASSAG"
72 REM *****
74 REM * ADATOK BEKERESE *
76 REM *****
80 PRINT "A DALLAM ELSO SZAKASZA"
90 INPUT D,B
100 PRINT:PRINT "A MASODIK SZAKASZ"
110 INPUT C,D
120 PRINT:PRINT "A DALLAM HARMADIK SZAKASZA"
130 INPUT E,F
140 PRINT:PRINT "A DALLAM NEGYEDIK SZAKASZA"
150 INPUT H,J
170 PRINT "BEFEJEZO SZAKASZ"
180 INPUT K,L
190 REM *****
192 REM * A ZENE MEGSZOLALTATASA *
194 REM *****
200 FOR A=0 TO 20
210 BEEP INT(A/0.1)*B
220 NEXT A
230 BEEP C,D
240 BEEP E,F
250 FOR G=0 TO 22
260 BEEP H,J
270 NEXT G
280 BEEP K,L
290 PRINT:PRINT
300 PRINT "AKARSZ MEG EGY MENETET? (I/N)"
310 INPUT M#
320 IF M#="I" THEN 10
330 IF M#="N" THEN 340
340 PRINT "HAT HA NEM HAT NEM "
350 PRINT CHR$(2):PRINT "SZIA":PRINT CHR$(1)
360 END
    
```

**Formatervezett,
megbízható működésű
botkormánygyártmányaink között
biztosan megtalálja
az igényeinek megfelelőit!**
Az alaptípusú készülék
termelői ára csupán 369 Ft.

Kérje katalógusunkat,
amit megcímzett válaszboríték
ellenében
díjtalanul megküldünk!

INTER GM.

Kaposvár, Rákóczi tér 1. 7400

ASM-250 típusú szünetmentes áramforrás

Hálózatkimaradás esetén megszakítás nélkül min. 30 perc időtartamig biztosítja a tápfeszültséget.
Névleges teljesítmény: 250 VA.

Érdeklődni lehet: ERFI -
Erőáramú Gyártmány-
és Felszerelési
Vállalat, Vállalkozási
Iroda 1027 Bp. Medve
u. 28-29. Telefoni
369-740, 354-140.
Telexi: 22 59 82 érli h.



Az előző részben összeállított kapcsolatra segítségével leírt rendszer működésére két példát mutatunk be: az első egy direktíva feldolgozásához, a második pedig a makróhívás felismeréséhez és a kifejtés indításához szolgál illusztrációként.

A példákban megadjuk a feldolgozandó forrásnyelvi sort, a rendszer állapotát és a feldolgozáshoz szükséges táblák és verememóriák tartalmát a feldolgozási ciklus kezdetekor, majd megvizsgáljuk a feldolgozással kapcsolatosan végrehajtandó feladatokat, s operátorról operátorra haladva végigkövetjük a feldolgozási ciklus folyamatát. Az elágazást megvalósító operátoroknál kijelöljük, hogy a T (igaz) vagy az F (hamis) ágon haladunk-e tovább.

1. példa

Legyen a rendszer az A1 állapotban:

```

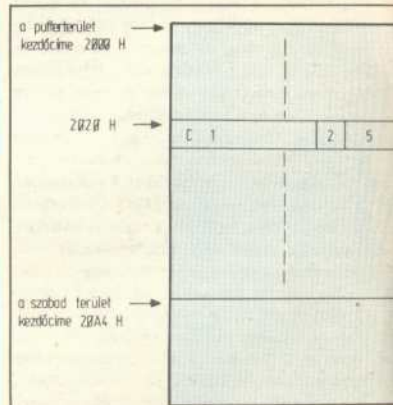
PASS=1
DEF=0
IF1=x (közömbös)
IF2=0
MIND=0
LOC=100 H (H-val hexadecimális
formában megadott
számot jelölünk)
    
```

A feldolgozás alatt lévő, SPB2-n tárolt forrásprogram sor legyen:

CIM1 DS C1+3

A végrehajtandó feladatok:

- a címkezőben szereplő „CIM1” relatív szimbólum elhelyezése a modulcím-táblában;
- a műveleti kód mezőben szereplő direktíva kiértékelése (LOC növelése), ehhez azonban előbb
- meg kell határozni az operandusmező értékét: (a szükségképpen predefini-



1. ábra

2. ábra

CIM1	1	100 H
------	---	-------

Assemblerek, cross-assemblerek

8. Példák a rendszer működésére

szimbólum) C1 értékét hárommal kell növelni;

— a sor megőrzése a második menet számára.

Tegyük fel továbbá, hogy a modulcímke-tábla puffertérületének tartalma az 1. ábra szerinti, és a „C1M1” szimbólumot még nem definiáltuk. A táblából kitűnik, hogy a „C1” szimbólum korábban EQU direktívával (2-es címkeflag) 5-ös értéket kapott, azaz programunkban létezik az aktuális sor előtt valahol egy például C1 EQU 5 alakú forrásor.

Az egymás után aktivizált operátorok a forrásprogramoknak az SPB2 területre való átvitele után a kapcsolatábrát követve az alábbiak.

Az operátor típusa	Az operátor funkciója
5	— Megtörténik a mezőkre bontás: megjegyzésmező: üres, címkemező: C1M1, műveletikód-mező: DS, operandusmező: C1+3.
2	— Elindul közben a következő forrásprogram sor olvasása, s ezzel párhuzamosan folytatódik az aktuális sor feldolgozása.
6	— A T ág aktivizálódik, mert a megjegyzésmező üres.
6	— $IF2 = \emptyset$, tehát a T ág érvényes.
6	— $DEF = \emptyset$, a T ágon haladunk tovább.
7	— Mivel $MIND = \emptyset$, az operátor aktivizálása után közvetlenül be is fejezi a működését.
6	— A címkemező 1. karaktere nem üres, tehát: F.
6	— $MIND = \emptyset$, tehát: T.
8	— Korábban nem definiáltuk a címkét, tehát: F.
9	— Az operátor elkészíti a címkebejegyzést (2. ábra: a címke relatív értéke 100 H), s elhelyezi a címke táblában a 20A4 H címre, a szabad terület kezdőcímének mutatóját az

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

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

12	— A műveletikód-mezőben nem makronév szerepel: F.
14	— A műveletikód-mező tartalma szerint keres a teljes műveletikódtáblában, megtalálja a DS direktívának megfelelő bejegyzést, ami tartalmazza, hogy — direktíváról van szó; — kötelező operandusa van a direktívának; — és megadja a 19-es operátoron belül a feldolgozó szubrutin kezdőcímét;
15	— Direktíva szerepel a műveletikód-mezőben: T.

16

— Az operátor először is megállapítja, hogy az operandusmező nem üres és benne egy összetett kifejezés szerepel; szétválasztja az egyszerű kifejezéseket: C1 és 3, és a kijelölt műveletet: +. Felismeri az első kifejezés szimbólum jellegét, és kutatni kezd a modulcímke-táblában, ahol a 2020 H címen megtalálja a „C1”-hez tartozó bejegyzést és megállapítja, hogy a szimbólum egy abszolút kifejezés és értéke 5. Elvégzi az összeadási műveletet, s a következő operátornak a 8-as operandusértéket és egy indikátort annak jelzését adja át, hogy az operandus abszolút típusú.

19

— Az operátor elvégzi a DS direktívának megfelelő műveletet: — beállítja a tárfoglalás nagyságát jelző változót 8-as értékre.

20 — SPB2 tartalma (a forrásor) lemezre kerül.

11 — A címke tábla még nem felt, tehát lemezre való kiírásra nincs szükség.

6 — Még nem érkezett be az END direktíva, IEND = \emptyset ; T.

21 — Lezárja a sorfeldolgozási ciklust; LOC értékét a tárfoglalás nagyságát megadó változó értékével növeli, az új érték 108 H.

Ezután indulhat újra a következő sorfeldolgozási ciklus.

2. példa

Legyen a rendszer az MK1 makrókifejtési állapotban, s a feldolgozás alatt álló forrásprogram sorban szerepeljen újabb (beskatulyázott) makróhívás. A rendszerváltozók kezdeti értéke:

PASS = 1

DEF=0
 IF1=x (közömbös)
 IF2=0
 MIND=1
 LOC=15A H.

Az első szinten kifejtett makróban a feldolgozásban soron következő sor legyen CIM2 MULTP OPI,OP2,X3 alakú. Tegyük fel, hogy a makródefiníciós könyvtárunkban már létezik egy MULTP nevű makró, amelynek formája a következő:

MULTP MACRO X1,X2,X3

utasítások

MEND

azaz három formális paraméterünk van: legyen ezek közül X1 és X2 két operandus címe, az eredményt pedig az X3 címre kell elhelyezni. A hívősorban már láthatóan az első két formális paramétert az OPI és OP2 aktuális paraméterek váltották fel (legyenek ezek modulcímek), a harmadik paraméter viszont még nincs behelyettesítve. Ez csak úgy lehetséges, ha X3 egyben a külső, a MULTP makróhívást tartalmazó makrónak is formális paramétere.

A végrehajtandó feladatok:
 — az X3 helyén a formális-aktuális paraméterscseré elvégzése az első szint paramétertábla-bejegyzésének alapján;
 — a címkemezőben szereplő „CIM2” relatív szimbólum elhelyezése az első szintű makrócímetáblában (ez tehát még a külső makró címkéje lesz);
 — a műveletikód-mezőben szereplő makrónév felismerése, a definíciós táblából a makró elhelyezkedési kezdőcímeinek meghatározása;

— az első szinthez tartozó makrócímetábla ideiglenes lezárása és kivitele a verem megfelelő eleme által meghatározott kezdőcímű lemezterületre;

— a makrókifejtési verem mélyítése: a megszakítás helyének és az aktuális paramétereknek az elhelyezése a megszakítási hely és a makróparaméterek veremtárában, LMCT növelése és az új érték beírása a makrócímetábla verembe;

— a nyert elhelyezkedési kezdőcímmel mint bemenő változóval a 3. operátor indítása és az SPB1-en az első beolvasott makró sor megjelenésének kivárása;
 — a sor megőrzése a második menet számára.

A makrókifejtési veremtárak állapotát a vizsgált sor feldolgozása előtt a 3. ábra adja meg. Az ábra szerinti felvételtől kitűnik, hogy:

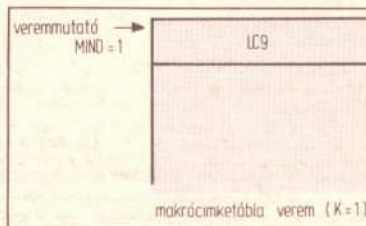
— a megszakítási hely és a makróparaméterek veremében a megszakítási cím helyén FFFF H áll, jelezve, hogy az elsődleges bemeneti periféria nem mágneslemez

(különbön érvényes lemezcinéma került volna erre a helyre);

— az első szinten kifejtett makróban négy formális paramétere van, köztük „X3”, ami egyben a második szinten hívott makróban is formális paramétere;
 — az aktuális paraméterek között szimbolikus nevek mellett számkonstans is szerepel.

A rendszer működését újból a kapcsolatra ugyanattól a pontjától kezdődően figyeljük. Az indított operátorok (az előző példánál kevésbé részletezve):

Az operátor típusa	Az operátor funkciója
5	Mezőkre bontja a feldolgozandó sort.
6	Nem megjegyzés sor:



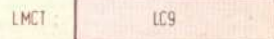
- T.
- 6 — IF2=0: T.
- 6 — DEF=0: T.
- 7 — Elvégzi a paraméterscserét: a hívősorban egyetlen olyan szimbólumot talál, amely egyezik egy, a formális paraméterek között szereplő szimbólummal („X3”), ezt a neki megfelelő (harmadik) aktuális paraméterre cseréli ki („DEST”), mégpedig mind az operandusmezőben, mind az SPB2 pufferterületen.

A sor végleges alakja tehát:
 CIM2 MULTP
 OPI,OP2,DEST lesz.
 Van címke: F.
 MIND=1: F.
 8,9 — A címkemező tartalmának feldolgozása megegyezik az 1. példában leírtakkal, azaz a különbséggel, hogy a bejegyzést nem a modulcímetáblában, hanem a kifejtés alatt álló makróhoz rendelt címetáblában helyezi el.

— Az operátor a műveletikód-mező tartalmával (MULTP)



megszakítási hely és makróparaméterek verem



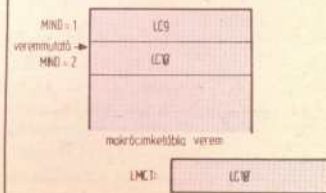
3. ábra

megegyező szimbólumot keres a makródefiníciós táblában. Feltételezésünk szerint talál ilyen szimbólumot, s a táblabejegyzést hozzáférhe-

4. ábra



megszakítási hely és makróparaméterek verem



tövé teszi a T ágon következő operátor számára.

Az operátor indítja a 2. szintű makrókifejtést, azaz

- az LC9-es lemez címen kezdődő területre kiviszi a makrócímetábla ideiglenesen lezárt rekordját;
- a soron következő lemezterület kezdetére állítja az LMCT változót: $LC10 = LC9 + 1$ ($K = 1$);
- mélyíti a megszakítási hely és makróparaméterek veremtarát (beírja az első szintű kifejtés megszakítási helyét és a makróhívás aktuális paramétereit);
- mélyíti a makrócímetábla vermet az LC10 cím beírásával;
- a makródefiníciós táblából kiolvassa a MULTP makró definíciójának elhelyezkedési kezdőcímét és a 3. operátor meghívásával beolvassa SPB1-re annak első sorát;
- MIND értékét 2-re növeli.

A két veremmemóriának az operátori működés befejeződése után előálló állapotát a 4. ábra mutatja be. A hívó sor lemezre kerül, lezáródik a sorfeldolgozási ciklus, s indulhat a következő sorfeldolgozása, amelynek tartalma éppen MULTP MACRO X1,X2,X3 lesz. A második szinthez tartozó formális paraméterek csak ekkor kerülnek be a veremtarba.

COMMODORE—64

BASIC RAM felosztás

A C64 memóriájának 64 k RAM-ja van. Ez azonban nem jelenti azt, hogy a felhasználónak a BASIC program tárolására ennyi szabad terület áll rendelkezésére. A 0—2048-ig terjedő területet az operációs rendszer a rendszer változóinak tárolására, munkaterületeknek és a képernyőtárnak foglalja.

A 4096 feletti területen a RAM-on kívül ugyanazokon a címenek RAM-ok is vannak, amik a gép operációs rendszerét és az alap karakterkészletet tárolják. Ezt a RAM területet a BASIC nem használja. Ide helyezheti el a felhasználó saját karakterkészletét, saját gépi kódban írt programjait és saját szerkesztésű adatait.

A BASIC munkaterület tehát alaphelyzetben, a gép bekapcsolása után a 2049—4096 közötti terület. Itt fognak elhelyezkedni a BASIC programok, a program által használt változók és sztringek. A megmaradó terület a szabad BASIC terület.

A teljes munkaterület felosztását a gép operációs rendszere végzi, de lehetőség van a felosztás megváltoztatására. A felosztást a munkaterület mutatóinak olvasásával vizsgálhatjuk, azok átirásával pedig megváltoztathatjuk. A mutatók a következő tárcímeken helyezkednek el (2. ábra):

- 43-44 SOB (Start Of Basic)
- BASIC terület eleje, alaph: 2049
- 45-46 SOV (Start Of Variables)
- változók kezdete, alaph: 2051
- 47-48 SOA (Start Of Arrays)
- tömbök kezdő címe, alaph: 2051
- 49-50 EOA (End Of Arrays)
- tömbök vége, alaph: 2051
- 51-52 BOS (Bottom Of Strings)
- sztringterület, alja, alaph: 40960
- 55-56 TOM (Top Of Memory)
- BASIC terület vége, alaph: 40960

A „Programmer's Guide” részletesen ismerteti a memória mutatóinak használatát, és megemlíti azok áthelyezésének lehetőségét.

Az áthelyezett „SOV” (Start Of Variables = változók kezdete mutató) esetén a program már nem módosítható, mivel a program vége és a „SOV” nem esik egybe. Az ilyen programmal végzendő manipulációhoz ismerni kell a program végcímét. Ezt a problémát oldja meg az itt közölt mintaprogram.

A program írásakor vagy betöltésekor a program elhelyezése a SOB által meghatározott címtől kezdődik. A program hosszától függően a SOV mindig följebb tolódik úgy, hogy a program által elfoglalt terület végét mindig követi. Ennek megfelelően változik a többi mutató is.

Mint látjuk, minden mutató két bajtot foglal el. Az első a cím alacsonyabb, a második a magasabb helyértékű bajtját tartalmazza. Például a SOV értékét így kapjuk meg:

```
PRINT PEEK(45)+PEEK(46)*256
```

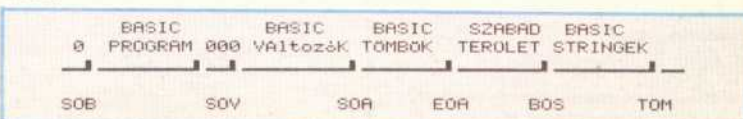
Valamelyik mutatóba új értéket a következő módszerrel írhatunk be (UJ=új érték):

```
HI=INT(UJ/256):LO=UJ-HI*256
POKE 45,LO:POKE 46,HI
```

A mutatók átállítását azonban nagy körültekintéssel kell végezni, hiszen ezek jelzik, hogy hol kezdődik és végződik a programunk, hol kezdődnek és végződnek a különböző típusú adataink (változók, tömbök, sztringek). A mutatók átállításával hozzáférhetlenné tehetjük azokat a gép számára. A BASIC NEW parancs sem törli valójában a meglévő programot, hanem csak a mutatókat állítja vissza a SOB értékére. A BASIC CLR parancs a SOB-ot és a SOV-ot változatlanul hagyva, csak a többi mutatót állítja; a program működőképes marad, de az eddig előállított változók eltűnnek.

Mi célja lehet e mutatók megváltoztatásának? Például, hogy a BASIC programtól független területeket foglaljunk le a tárban

1. ábra



Sorozatunkban a (cross-assembler programok működésének egy lehetséges algoritmusát tárgyalunk rendszermodelljezzel eszközökkel. Reméljük, az olvasó megjegyzésként a alkalmazott módszer hasznosságát.

egyéb célokra, sprite-ok, gépi kódú programok vagy adatok tárolására.

Sprite-ok

A sprite-okat a tárnak abban a 16 k-s blokkjában kell elhelyezni, amelyikben a képernyőtár is van. A képernyőtár alaphelyzetben az 1024—2023 közötti címen van, így a sprite-oknak is az első 16 k-s blokkban kell lenniük. Elvileg így a 2048—16383-ig terjedő területből választhatnánk, de a valóságban csak a 2048—4095 és a 8192—16383 közötti terület használható. Mivel a BASIC program is ezen a területen dolgozik, így vagy a program fogja elrontani a sprite-okat, vagy azok rontják el a programot. Ezt megelőzhetjük, ha a tárnak egy részét lefoglaljuk.

Gépi kódú programok

Ha munkánkhoz gépi kódú programot is használunk, azt elhelyezhetjük a 40960 fölötti területen. Szükség lehet arra is, hogy a gépi kódú rész a BASIC programmal egy egészet alkosson, és azzal együtt lehessen betölteni. Ebben az esetben szintén le kell foglalni egy területet.

A BASIC területen belül egy meghatározott terület lefoglalására három lehetőség van: területfoglalás a BASIC program előtt, után, vagy a BASIC munkaterület végén.

Területfoglalás a BASIC program előtt

A SOB feljebb helyezésével tetszés szerinti terület foglalható le a BASIC program elhelyezési területe előtt (2. ábra).

Például a SOB-ot 8192-re állíthatjuk a következő módon: POKE 43,0:POKE 44,32:POKE 8191,0:NEW

A POKE 8191,0 az új SOB előtti bajtot állítja nullára. Jegyezzük meg, hogy a BASIC kezdőcím előtti bajtnak mindig nullának kell lennie, ellenkező esetben a gép SYNTAX ERROR-t jelez. A NEW parancs a többi mutatót igazítja a megváltozott SOB-hoz.

A módszer hátránya, hogy az utasítást a program betöltése előtt kell kiadni. Ha az áthelyezett SOB után kezdjük el betölteni a programot, akkor az a tárban a 8192-es címtől kezdve helyezkedik el, szabadon hagyva a 2048—8190 közötti területet.

Területfoglalás a BASIC program végén

A program betöltések vagy beírásokor a program a SOB-tól kezdődően fog a tár-



2. ábra

ban elhelyezkedni. A program végét a SOV jelzi. Innen kezdve helyezkednek el a változók. Ha a SOV-ot feljebb helyezzük meg mielőtt bármilyen változót használnánk, a program vége és a változók kezdete előtt tetszés szerinti területet foglalhatunk le. Amit az így lefoglalt területre helyezzünk, az a program futása alatt sértetlen marad. De ennél sokkal nagyobb a jelentősége annak, hogy a BASIC SAVE a mentést a SOB-tól kezdve a SOV-ig végzi, így a BASIC programmal együtt a lefoglalt terület tartalmát is kimentí. Ugyanígy a LOAD parancs az egészet vissza fogja tölteni a tárba (3. ábra).

Hátránya ennek a megoldásnak, hogy ha a BASIC programot és a lefoglalt területre töltött gépi kódú programot egyszer így összetársítottuk, akkor a BASIC programban már semmilyen változtatást nem lehet végezni. Ennek oka a következő. A BASIC program végét a SOV már nem helyesen jelzi. A BASIC program módosításakor a SOV értéke annyival fog változni, ahány bajtot a programhoz hozzátettünk vagy tö-

következik, ami a következő BASIC utasítás tárbeli kezdőcímét mutatja. Ha nincs több utasítás, akkor ez a két bajt is nulla. Ezt az egymás utáni három nullás bajtot keresi meg a program.

A programot saját programunk elé kell beírni. A saját program ez után, a 10-es sor számtól kezdődik. A saját program indítása a RUN parancsral, a BASIC végét kereső program indítása pedig RUN 2 parancsral történhet.

Ha a SOV-ot megváltoztatjuk és programunkat is módosítjuk, a program végét a RUN 2 parancsral bármikor ismét megkaphatjuk, miután az végigköveti a BASIC sorok kezdetére mutató címekeket, amíg a BASIC végét jelző 00 értékű mutatót meg nem találja.

A P-2 jelű program ugyanezt a feladatot látja el gépi kódú program segítségével.

A 100-as sorban történik a gépi kódú program betöltése.

A 110-es sor ellenőrzi, hogy helyes adatokat töltöttünk-e be. A 120-as sor törli a BASIC programot.

A program futtatása után vihetjük be saját programunkat, átírhatjuk a SOV-ot és módosíthatjuk programunkat. A BASIC program végét a SYS 49169 parancsral írhatjuk ki.



3. ábra

röltünk. Ha ismerjük a SOV előző és a megváltozott értékét is, akkor abból kiszámíthatjuk, hogy szabad területünk helye hogyan változott. Többször is változtatgatások után előfordul, hogy nem tudjuk követni azokat. Ha ilyenkor megkeressük a BASIC program végét, akkor ismét megkaphatjuk a szabad terület helyét: a program vége és a SOV közötti területet.

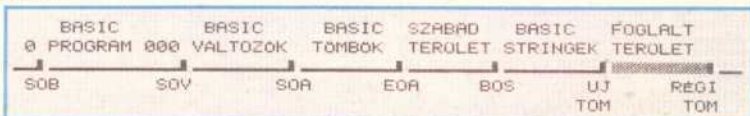
A P-1 jelű program a BASIC program által elfoglalt terület utolsó bajtját keresi meg és írja ki a képernyőre. A keresés a következő módszerrel történik. A BASIC utasítássorokat a tárban egy-egy 0-ás bajt választja el. Ezek után egy kétbajtos mutató

Területfoglalás a BASIC munkaterület végén

A BASIC munkaterület a SOB-tól a TOM-ig terjed. A program futása közben keletkező sztringeket a gép a TOM-tól lefelé helyezi el. Az elfoglalt terület alját a BOS jelzi. Ha a program indítása előtt a TOM-ot lefelé eltoljuk, akkor a régi és az új TOM közötti terület szabadabb válik. A TOM mutató lekérdezését és átírását a mutatóknál ismertetett módon végezhetjük (4. ábra).

ZSOM BÉLA

4. ábra



BASIC és gépi kód

Legutóbb megnéztük egy rövid gépi kódú rutin elkészítésének folyamatát. Ez a kiszámított GOTO utasítás használatát lehetővé tevő alprogram a gép csak olvasható memóriájában található gépi kódú rutintokat aktivizálta JSR és JMP utasításokkal.

Most az úgynevezett töltőutasításokkal és néhány velük kapcsolatos tudnivalóval ismerkedhetünk meg.

A töltőutasítások

Nevük a „load” angol szó első és utolsó betűjéből és a használt regiszter nevéből áll. Hármán vannak: LDA, LDX, LDY. Használatuk az értékadó utasításokéhoz hasonlítható: az operandussal meghatározott memóriacím tartalma a műveleti kódban szereplő regiszterbe kerül. A regiszter korábbi tartalma elvész, a címzésben szereplő memóriabájt tartalma változatlan marad.

A töltőutasítások a mikroprocesszor állapotregiszterének N- és Z-bitjeit állítják be, a betöltött bájt értékének megfelelően.

Ezeknek a feltételbiteknek a beállítását a töltőutasítást követő, ún. feltételes ugróutasításban lehet kihasználni. Ezekről később lesz szó.

A feltételbitek értelmezése

A regiszterbe töltött bájt tartalmát tekintetjük egy olyan előjel nélküli számnak, melynek értéke a 0 ... 255 tartományba esik, de ugyanígy tekinthetjük egy -128 ... +127 értéktartományba eső előjeles számnak is. Ez utóbbi esetben, a szám 7. bitjét előjelbitként használva, az ún. kettes komplementens ábrázolással dolgozhatunk. Ebben az esetben segít az N-bit; a regiszterbe töltött szám előjelét megőrzi egy esetleges elágazó utasítás számára, mely a regiszterbe töltött szám pozitív vagy negatív voltától függően szabja meg a további programfutás menetét.

Röviden: az N-bit a töltőutasítások végrehajtása során az éppen használatos (A, X, ill. Y) regiszter 7-es sorszámú bitjének

értékét veszi fel, vagyis akkor lesz 1, ha a regiszterbe kerülő szám negatív.

Egyszerűbb a Z-bit működésének megértése. Akkor veszi fel az 1-es értéket, ha a regiszterbe kerülő bájt minden biteje 0, vagyis a regiszterbe kerülő szám nulla (zéró).

A Z-bit állapotát is feltételes ugróutasításban lehet kihasználni.

Tárolóutasítások

A töltőutasítások párpai, a regiszterek tartalmát viszik (másolják) az operandus által meghatározott tárterületre. Nevük az angol „store” szó első két betűjéből és a használt regiszter nevéből tevődik össze. Szintén hármán vannak: STA, STX és STY. Működésük a töltőutasításokéhoz hasonlóan a magas szintű nyelvek értékadó utasításához hasonlítható. Egy nagyon fontos tulajdonságuk különbözteti meg őket a töltőutasításoktól: a tárolóutasítások a feltételbiteket változatlanul hagyják. Érdemes megjegyezni!

BARNA LÁSZLÓ

```

1 GOTO 7
2 PRINT : PRINT " A BASIC PROGRAM UTOLS
  0 BYTE-JA." : PRINT
3 MUTAT = 43
4 MUTAT = PEEK(MUTAT) + PEEK(MUTAT +
  1) * 256
5 IF MUTAT > 0 THEN PRINT "[CU]"; M
  UTAT + 2; GOTO 4
6 END
7

```

A P-1 program

A P-2 program

```

10 PRINT "[CLR][CD][CD][CD]"
20 PRINT " *****"
22 PRINT " * * * * * "
24 PRINT " * [RVS] END OF BASIC
  [RVO] * "
26 PRINT " * <BAS.PROGR. VEGE> * "
28 PRINT " * HIVAS:SYS 49169 * "
30 PRINT " * * * * * "
32 PRINT " *****"
40 :
50 SOB = PEEK(43) + PEEK(44) * 256
60 PRINT "[CD][CD][CR][RVS][RVO]TART
  [RVS][RVO]F [RVS]B[RVO]ASIC:"
  PRINT "(BASIC PROGRAM KEZDOCIME)":
  PRINT SOB
100 FOR I = 49152 TO 49288: READ A:
  POKE I, A: S = S + A: NEXT

```

```

110 IF S < > 18828 THEN PRINT "HIBA
  A DATA SORBAN"
120 POKE SOB,0: POKE SOB + 1,0: SYS 49
  169: NEW
1000 DATA 32, 69, 78, 68, 32
1002 DATA 79, 70, 32, 66, 65
1004 DATA 83, 73, 67, 32, 65
1006 DATA 84, 58, 169, 13, 32
1008 DATA 210, 255, 162, 0, 189
1010 DATA 0, 192, 32, 210, 255
1012 DATA 232, 224, 17, 208, 245
1014 DATA 169, 13, 32, 210, 255
1016 DATA 169, 43, 133, 252, 169
1018 DATA 0, 133, 253, 141, 192
1020 DATA 2, 160, 0, 177, 252
1022 DATA 200, 17, 252, 240, 59
1024 DATA 177, 252, 133, 255, 136
1026 DATA 177, 252, 133, 254, 169
1028 DATA 13, 32, 210, 255, 169
1030 DATA 145, 32, 210, 255, 164
1032 DATA 254, 165, 255, 32, 145
1034 DATA 179, 32, 221, 189, 162
1036 DATA 0, 189, 0, 1, 8
1038 DATA 32, 210, 255, 232, 40
1040 DATA 208, 245, 173, 192, 2
1042 DATA 240, 1, 96, 165, 254
1044 DATA 133, 252, 165, 255, 133
1046 DATA 253, 76, 51, 192, 169
1048 DATA 1, 141, 192, 2, 101
1050 DATA 254, 133, 254, 165, 255
1052 DATA 105, 0, 133, 255, 76
1054 DATA 69, 192

```

Z80 programozási

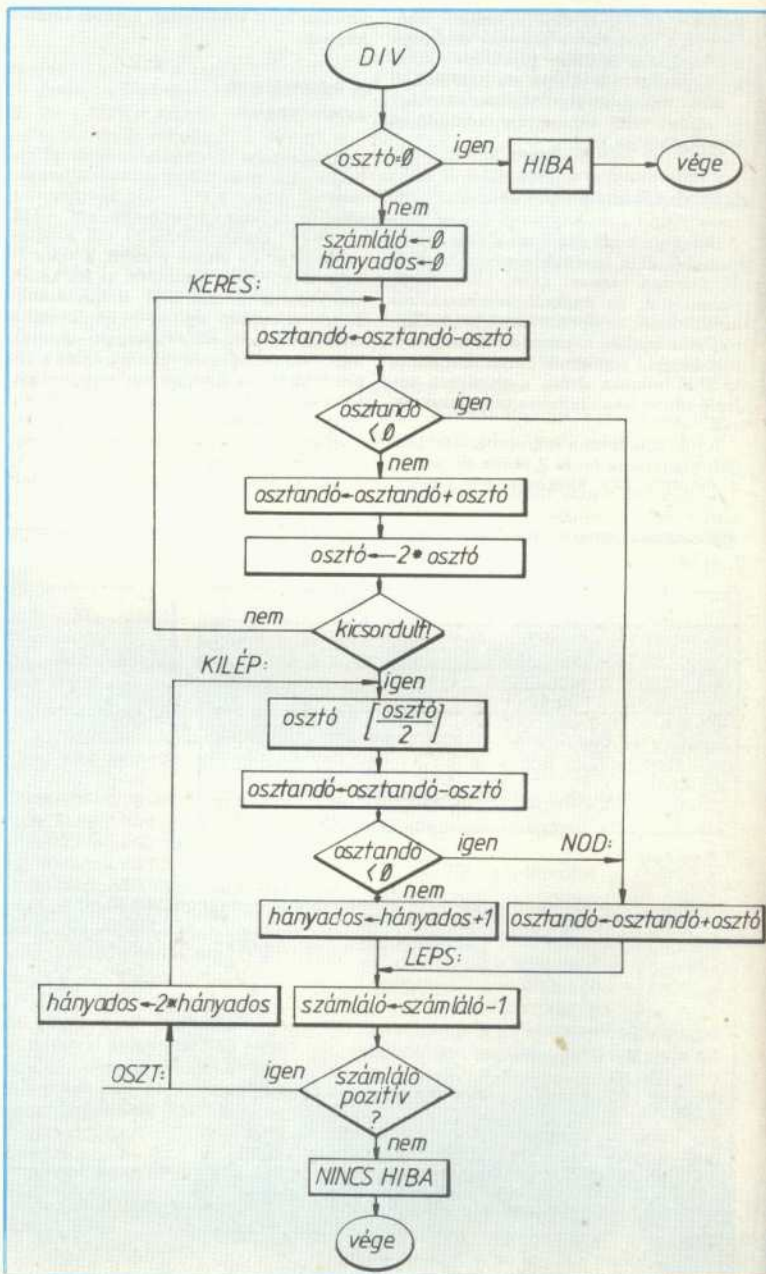
Az előző részben megnéztük, hogy hogyan tudunk két bájtól ábrázolt számokat decimálisan kiírni. Most nézzük meg ennek a fordítottját!

1. Írjuk szubrutint, amely egy, a HL által mutatott címtől karakter alakban adott decimális, előjel nélküli egész szám bináris értékét beírja DE-be! Ha hibás a szám, carry bit jelezze a hibát!

A karakter alakban adott decimális egész azt jelenti, hogy a tárban valahol a számjegyek ASCII kódjai szerepelnek egymás után. Például 124 a következőképpen: ...49,50,52,... Valahonnan tudnunk kell, hogy hol kezdődik és hol végződik a szám. Az első karakterre mutasson HL regiszterpár, és tartson addig a szám, ameddig számjegyek vannak, azaz az első nem "0" és "9" közé eső bájtig.

Egy tetszőleges számrendszerben felírt szám, amelynek jegyei: a, b, c, d (legyen most csak négyjegyű!), értéke $aA^3 + bA^2 + cA + d$, ahol A a számrendszer alapszáma. Ez felírható $((aA + b)A + c)A + d$ alakban is. Ez az alak, az ún. Horner-séma, egy nagyon egyszerű kiszámítási módot sugall: szorozzuk a-t A-val, adjuk hozzá b-t, szorozzuk A-val, adjuk hozzá c-t, szorozzuk A-val, adjuk hozzá d-t. Ha pedig a szám végére íránk még egy számjegyet, akkor az előzőekből kiszámítható az így kapott szám értéke, csak szorozni kell A-val, és hozzáadni az újabb számjegyet. Az algoritmus ezek után a következő.

Mindaddig, amíg találunk számjegyet, az eddigi értéket szorozzuk tizzel, és hozzáadjuk az újabb számjegy értékét. Hogy ne legyen gondunk az „eddigi” értékkel, a szám elején feltételezünk egy fiktív nullát, amit azután szorozhatunk tizzel és hozzáadhatjuk a megfelelő számot (lásd az 1. programot). A program elején DE-be nullát töltünk, feltételezve a fiktív nullát a szám előtt. Az akkumulátor vesszős párjában számláljuk a számjegyeket, ezért ezt a szubrutin elején lenullázzuk. Ha ugyanis az első nem számjegyekarakter előtt nem találunk egyetlen számjegyekaraktert sem, akkor vizsgálatéskor a carry bittel jelezniünk kell a hibát. Ezután előveszük a következő karaktert. Az akkumulátorból levonva "0"-t megkapjuk az ASCII kódhoz tartozó számot. Ha a kivonás után carry=1, akkor nem számjegyekarakter volt az elővett karakter. Akkor sem, ha a kivonás után az akkumulátor nagyobb mint 9, azaz ha az eredeti érték nagyobb volt, mint "9". Ezután szoroz-





nagyobb, mint az osztandó. Ezután lenulázzuk az akkumulátort és a BC regiszterpárt. A BC regiszterpárban fogjuk képezni a hányadost, az akkumulátorban pedig azt számoljuk, hogy hányszor szoroztuk kettővel az osztót, míg az éppen nagyobbá vált, mint az osztandó.

Mivel így eggyel többször szorozunk kettővel, mint ahogy arra szükségünk lenne, kivéve azt az esetet, amikor az osztó nagyobb, mint az osztandó, ezért a BC balra tofása után rögtön az osztó kettővel való osztásával kezdjük a levonogatásokat.

BC-ben a hányadost szeretnénk kapni, ezért amikor balra toljuk, azaz kettővel szorozzuk, a művelet után nem lehet carry=1, hiszen az túlcsofordulást jelentene; így az osztó kettővel való osztásakor a 15-ös bitbe nulla kerül, és nem okoz gondot, hogy forgató- és nem léptetőutasítást használunk. Ez után a művelet után ugyancsak nem lehet carry=1, mert DE, azaz az osztó első bitjébe nullát töltöttünk, és nem forgatjuk annyiszor jobbra, hogy értékes bit kicsorodjon. A BC-ben először nulla van, így nem baj, hogy először azt szorozzuk kettővel.

Ha az osztó nagyobb, mint az osztandó, akkor nem történik műveletvégzés, a carry törlése után befejeződik a szubrutin. Amikor az osztót szorozzuk kettővel, az kétféleképpen válhat nagyobbá az osztandónál. Vagy úgy, hogy DE nagyobb lesz, mint HL, és a kivonás után carry bit jelzi ezt, vagy úgy, hogy nagyobbá válik, mint ami 16 biten elfér, ezt a kettővel való szorzás után jelzi a carry jelzőbit, és ekkor is biztos, hogy nagyobb az osztandónál, hiszen nagyobb minden 16 bites számmal. Ekkor a futás a KILEP címkével jelölt sorban folytatódik, és itt a 15-ös bitbe visszakerül a helyes érték. Ezért nem írhatunk ebbe a sorba léptetőutasítást.

Három különböző esetre nézzük végig a program futását! Egy olyanra, amikor az osztó nagyobb, mint az osztandó, egy olyanra, amikor az osztó nagyobb, mint 7FFFH, azaz a 15-ös bitje nem nulla, és kövszünk végig egy olyan esetet is, amelyik ezek egyike sem.

Gondoljuk végig a regiszterek megválasztását! Próbáljunk meg előjeles számok osztására programot írni! Próbáljunk meg olyan szorzó szubrutint írni, amely az eredményt 32 biten adja meg!

Egyváltozós függvények ábrázolása és megoldása

A program egyváltozós függvényt, illetve első deriváltját ábrázolja a megadott intervallumban. Megkeresi ebben az intervallumban a függvény lokális szélsőértékeit, és mint-asor tartalmazza, a y=f(x) alakban.

Bemenetek

MODE 1 csak a függvényt, MODE 2 deriváltját is ábrázolja

k kezdőpont

v végpont (az ábra intervalluma)

F függvényazonosító, nyomtatásra kerül

y az ábrázolt y tengely hossza (egységben)

C1 a függvény színe

C2 a derivált színe ábrázolásnál

ITV gyökkeresés léptetési intervalluma

IT szélsőérték-keresés léptetési intervalluma

A program felépítése

5-10 kezdőértékek
10-40 értékészlet AC-ban, min. és max. „P” és „O”-ban
40-50 bemenetek és nyomtatás
50-60 függvény és deriváltjának ábrázolása

69-80 x tengely
81-86 x tengely beosztások
87-100 y tengely
101-107 y tengely beosztások
110-170 gyökkeresés
180-230 szélsőérték-keresés
250 derivált függvény
300-310 x és y tengely beosztásának kezdőértéke és léptetése
400-420 értékészlet max. és min. kiválasztása

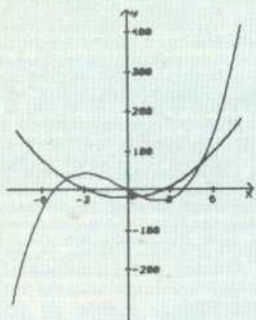
500 y=f(x) függvény
Egy harmadfokú (1. ábra) és két összetett trigonometrikus függvényen (2. ábra) mutatjuk be a program használatát. Az eredeti ábrák önmagukban érthetőek voltak, de a négy színű ábrázolás lehetőségével nem rendelkezünk, ezért tudni kell, hogy minden ábrán f(x) a lejjebb levő.

```

5: CLEAR : TEXT : P=9999: Q=-P: NA="+"
  .#####: INPUT "MODE (1,2) =" : R
10: Q=51: DIM A(Q): INPUT "K=": K: "U=":
  U=(U-K)/Q: X=K-U: FOR I=0 TO Q: X=
  X+1: GOSUB 500
20: A(I)=Y: GOSUB 400
25: IF I=0 OR R<1 GOTO 40
30: GOSUB 250: GOSUB 400
40: NEXT I: INPUT "F=": A0: USING NA:
  LPRINT "Y=": A0: LPRINT "MIN=": P:
  " MAX=": Q: USING
45: T=200: INPUT "Y=": E: "C 1=": C:
  COLOR C
46: A0="(" : ---: IF R=2 LPRINT A0:
  INPUT "C 2=": D: COLOR D: LPRINT A0:
  GRAPH : SORGM : LINE (18,33)-(18,
  20): TEXT
47: LF 2: GRAPH : GLCURSOR (5, 8): SORGM
  : U=0: P=N: PVE/U-E
50: COLOR 0: L=3: FOR I=L TO Q: X=I+T/Q
  : Y=A(I): IF L=1 GOSUB 250
52: X=I+T/Q: Y=A(I)
55: IF I=L GLCURSOR (X, Y)
60: LINE -(X, Y): NEXT I: IF R=2 AND L=0
  LET L=1: GOTO 50
63: Y=Q+E: CO-P: IF Y) BLET Y=5
70: IF Y<ELET Y=-E-5
80: T=210: COLOR 0: LINE (L-5, Y)-(T, Y):
  RLINE (L-5, 5)-(L-5, -5):
  GLCURSOR (T-0, Y-B): CSIZE 1:
  LPRINT "X"
81: T=200: C=U-K: D=K: GOSUB 300
82: B=(J-K)/WT/(U-K): IF J=0 GOTO 06
83: LINE (B, Y+2)-(B, Y-3): L=0-3*LEN
  STR$ J: IF L<5 LET L=5
84: GLCURSOR (L, Y+12): LPRINT STR$ J
85: J=J+1: IF J<0 GOTO 02
87: X=K+T/(Q-U): IF X<5 LET X=5
90: IF X) LET X=T+5
100: LINE (X, E-15)-(X, 15): RLINE (L-5,
  -5)-(L-5, 5)-(L-5, -5): LPRINT "Y"
101: C=J: D=P: GOSUB 300: L=(LEN STR$ J
  )/6-X*5/11
102: B=(C-P)/VE/U-E: IF J=0 GOTO 107
103: IF B) B GOTO 103
104: LINE (X-2, B)-(X+3, B): IF L=0
  GLCURSOR (X+5, B-3): GOTO 106
105: GLCURSOR (X-(LEN STR$ J)/6-2, B
  -3)
106: LPRINT STR$ J
107: J=J+1: GOTO 102
109: GLCURSOR (0, -E): TEXT : CSIZE 2: LF
  3
110: USING NA: INPUT "ITU.=": I: I=I+K-U
  120: P=N+M: C=N: IF C) GOTO 100
130: X=C: GOSUB 500: U=Y: X=X+.001: GOSUB
  500: D=C: C=D-U/(Y-U)*.001: IF C)
  AND C) GOTO 100
140: IF ABS (D-C) > 1E-06 GOTO 130
170: LPRINT "GYOK=": C: GOTO 120
180: INPUT "IT=": I: X=K: A0=" " : B=" "
  " min="
190: B=4: GOSUB 500: U=Y: X=X+B: GOSUB 50
  0: B=2:(U)Y: X=X-B
200: N=X+B: GOSUB 500: IF Z<(U)Y) OR U=
  YLET B=-B/3: IF ABS B) 1E-06 GOTO 23
  8
210: IF X=0) JLF 4: END
220: U=Y: GOTO 200
230: LPRINT 00(2+1): (U+Y)/2: " X=": X
  : GOTO 130
250: Y=(A(I)-A(J-1))/4: RETURN
300: A=I*B+INT (LDE CC(0)): I=INT (A
  CC/50.5)/A: J=INT (D+1.93*I
  310: S=J+1/2: IF D<S.05 AND INT (ABS)
  A)LET J=S
320: RETURN
400: IF Y<P LET P=Y
410: IF Y) DLET D=Y
420: RETURN
500: Y=X*Y*X+X*X-20*Y: RETURN

```

Y=X**X*X+X**X-20**X
 MIN=-2.880000E 02
 MAX=+4.159999E 02
 f(x) ---
 f'(x) ---

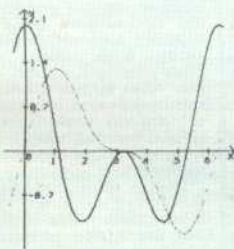


GYOK=-5.000000E 00
 GYOK=+0.000000E 00
 GYOK=+4.000000E 00
 max=+4.203149E 01
 X=-2.936724E 00
 min=-2.855001E 01
 X=+2.270096E 00

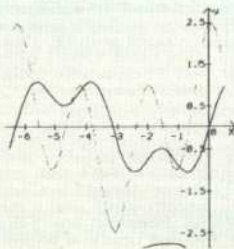
1. ábra
 2. ábra

Y=SINX+SIN(2X)/2
 MIN=-1.297961E 00
 MAX=+1.995780E 00

f(x) ---



Y=SINX+SIN(3X)/2
 MIN=-2.488505E 00
 MAX=+2.487780E 00
 f(x) ---



Tartalom a képernyőn

A bemutatott program a mikroszámítógép memóriájában tárolt információkat jeleníti meg. C64 gépre készült, de más géptípusra is könnyen átirható. A képernyőn egyszerre $16 \times 8 = 128$ bájttal hexadecimális és karakteres alakját tanulmányozhatjuk.

A gyorsabb futás érdekében a hexadecimális megjelenő szubrutin a program elején szerepel. Ugyanezért a konstansokat változókból tároljuk, és a változóknak a program elején a felhasználási gyakoriságuk sorrendjében értékadásokkal helyet foglalunk. A valódi változók itt csak egy üres értéket kapnak.

A program valódi változói

Y: egy 8 bájtos memóriaszelet kezdőcíme
 X: az átalakítandó és megjelenítendő érték
 K: sorok száma a képernyőn
 I: ciklusváltozó
 Z: egy hexadecimális jegy értéke az átalakításkor

Lényeges konstansok

4096, 256, 16: a hexadecimális átalakításhoz szükséges osztók

0,1, ..., F: hexadecimális karakterek, az A5 tömbben tárolódnak

127: maszk egy bájttal előjelbitjének elnyomására

32: az ennél kisebb bájtok karaktereké helyett a tizedespont jelenik meg, melynek ASC kódja 46

8: ahány bájtról egy képernyősorban információ jelenik meg

65528: ennél nagyobb kezdőcímmel 8 bájtos sorozatról már nem jeleníthető meg információ

TS: 37 darab létezés a kereten belüli törlesztés

Vezérlő és grafikus karakterek

CHRS(3):	STOP
CHRS(13):	RETURN
CHRS(17):	CRSR
	DOWN
CHRS(18):	RVS ON
CHRS(19):	HOME
CHRS(29):	CRSR
	RIGHT
CHRS(147):	CLR
CHRS(149):	barna
CHRS(194) és CHRS(221):	függőleges vonal karakterek

A program felépítése

200—230: X megjelenítése 4 jegyű hexadecimális formában
 220—230: X megjelenítése 2 jegyű hexadecimális formában
 300—490: helyfoglalás a valódi változóknak és a konstansok áttöltése változókbá (konverzió!!)
 510 : képernyőszin-beállítás
 520—540: kezdőcímbekérés
 550—610: keret rajzolása
 630 : STOP/RESTORE tiltás
 640—700: 8 darab bájttal megjelenítés
 710 : a következő képernyősorra lé-

COMMODORE—64

pés, ha még nem telte be a képernyő

730—770: üzenet kiadása a képernyő alján
 780—800: RETURN-ra várakozás, STOP esetén befejezés

810—850: törlés, és folytatás tiszta képernyőn

Speciális utasítások

POKE 198,0 : billentyűzet-puffer „üritése”

POKE 808,225: STOP/RESTORE tiltás
 POKE 808,237: STOP/RESTORE engedélyezés

POKE 781,X: POKE 782,N—I: SYS 59905 az X-edik sor elejétől N darab pozíció törlése a képernyőn, de a kurzor eredeti helye nem változik (N < 128)

DR. SZÖRÉNYI MIKLÓS

```

100 GOTO 300
110
200 Z=INT(X/Z) : X=X-Z*C : PRINT A$(Z)
210 Z=INT(X/B) : X=X-Z*B : PRINT A$(Z)
220 Z=INT(X/A) : X=X-Z*A : PRINT A$(Z)
230 PRINT A$(Z)A$(Z)*" " : RETURN
240
300 X=0 : Z=0 : I=0 : A=16 : E=127
310 F=32 : S=0 : G=46 : Y=0 : K=0
320 B=256 : H=0 : C=4096 : D=65528
330 P=128 : Q=1 : R=7 : T=0 : R#=#81
340 W=782 : S=508 : M=108 : N#=#781
350 U=225 : V=237 : A#=#19 : L#=#119
360 A#=#CHR$(10) : B#=#CHR$(25)
370 C#=#CHR$(19) : D#=#CHR$(17)
380 E#=#CHR$(147) : F#=#CHR$(149)
390 T#=#" "
400 T#=#" "
410 P#=#CHR$(3) : R#=#CHR$(13)
420 K#=#CHR$(221)+T#+#CHR$(194)
430 H#=#"FOLYTATÁS " : RETURN " "
440 N#=#" DISPLAY MEMORY "
450 DIM A$(15)
460 FOR I=0 TO 9
470 A$(I)=CHR$(48+I) : NEXT I
480 FOR I=10 TO 15
490 A$(I)=CHR$(55+I) : NEXT I
500
510 POKE 53280,13 : POKE 53281,13
520 PRINT E#F#;
530 INPUT "KEZDOCIK ( 0 - 65528 ) : " V
540 IF VCT OR VCI=INT(V) OR VDI GOTO 520
550 PRINT E#SPC(11)/R#;
560 PRINT " "
570 PRINT " "
580 FOR I=1 TO 16
590 PRINT K# : NEXT I
600 PRINT " "
610 PRINT " "
620
630 POKE R#;V : PRINT D#
640 IF VDI THEN POKE R#;V : END
650 S=Y : PRINT B# : GOSUB 200 : S=Y+R
660 FOR I=0 TO S : X=PEEK(I)
670 GOSUB 220 : NEXT I : PRINT A#;
680 FOR I=0 TO 15 : X=PEEK(I) AND E
690 IF XCF THEN S=S+X : NEXT I
700 PRINT CHR$(X) : NEXT I : PRINT
710 V=Y+H : K#=#+2 : IF KCR GOTO 640
720
730 PRINT
740 X#=#F : PRINT SPC(13) : GOSUB 200
750 PRINT " " : X#=#0 : GOSUB 200
760 PRINT PRINT
770 PRINT SPC(5)/H# : POKE KN;T
780 GET X# : IF X#="" GOTO 780
790 IF X#=#P THEN POKE R#;V : END
800 IF X#=#R GOTO 780
810 POKE R#;N : POKE YR;L : SYS 59905
820 PRINT B# : K#;T
830 FOR I=0 TO 17
840 PRINT B#F# : NEXT I
850 PRINT D# : GOTO 640

```


Az Iparban, a mindennapi életben használatos, értékesíthető megoldások megtalálása nem kizárólag az idő szakemberek privilégiuma! Ma már egyre inkább terjed az a felismerés, hogy esetleg még egy általános vagy éppenséggel egy középiskolás fiatal is képes egyes területeken packépes ötletek létrehozására, csak komolyan kell foglalkozni az általuk felvetett gondolattal. Ez a felvétele kissé különösnek hangzott hazánkban öt esztendővel ezelőtt, amikor a fiatalok és különösképpen a fiatal szakemberek ötleteinek felkarolására a KISZ KÖ az Állami Fejlesztési Bankkal közösen meghirdette az „Alkotó Ifjúság” pályázatot, majd ennek menedzselésére, a beérkezett pályaművek hasznosítására létrehozta az Alkotó Ifjúság Egyesületét, amely napjainkra már jócskán túlnötte egykori kereteit. Most már dinamikus fejlődés és változó szolgáltatás-menedzser-fejlesztő-innovatív formában tevékenykedik. Budapesti központjában négy szakirodában, az észak-magyarországi területi képviseletben integrált formában folyik ez a komplex munka, amelyet kiegészít a tíz megyei képviseleten folyó piaci feltáró és komplex fejlesztő tevékenység. Több száz újítást, szabadalom, üzleti sikereket és egypár kevésbé eredményes próbálkozást jellemző tevékenységüket, amelyet a számítástechnika korai időszakban történő megjelenése, majd napjainkig tartó egyre dinamikusabb felütása fémjeléz. E szerzőgázó tevékenységről beszélgettünk Kulcsár Istvánnal, az Alkotó Ifjúság Egyesülés budapesti számítástechnikai irodájának vezetőjével, valamint dr. Csomai Zoltánnal, aki az észak-magyarországi területi képviseletet vezeti.

µM: Egy olyan menedzser jellegű vállalat, mint az Alkotó Ifjúság Egyesülés hogyan válhatott a számítástechnika területén szolgáltató vállalat?

K. I.: Alig egy évvel az egyesülés beindulása után nyilvánvalóvá vált, hogy nem nélkülözheti a számítástechnikát. Sem saját belső munkájában, sem abban, hogy eladásra alkalmazói szoftvereket menedzseljen. Ehhez viszont kellett egy gép is. Akkoriban a PC-k nem voltak még elterjedve, így csakis nagyszámítógép jöhetett szóba. Az OMFB-nél megpályáztunk egy ilyen irányú támogatást, amire 20 millió forintot kaptunk. Ehhez viszont az a feltétel csatlakozott, hogy géppünkkel meg kell oldani az Országos Találmányi Hivatal számítástechnikai feladatait. Így sikerült egy PDP kompatibilis magyar TPA gépet beállítani, amely ma is üzemel. Eلسorban a találmányi bejelentésekkel kapcsolatos adminisztrációt intézi az OTH részére. Működik már a megadott magyar szabadalmak nyilvántartása, fejlesztés alatt áll a védjegyregiszter. Emellett az OTH nyilvántartásait és statisztikáit is e gép programrendszereivel készítik. A fennmaradó időben a gépet az egyesülés feladataira lehet felhasználni. Természetes, hogy az első időszakban forgalmazott szoftverek is e géptípusra íródtak. Mintegy 15 TPA rendszer-szoftver-kiegészítő forgalmazunk belföldre és a SZÁMALK—OSAK megállapodás keretében a Szovjetunióba is. Ekkor

A számítástechnika nem ámitástechnika

Interjú az Alkotó Ifjúság vezetőivel

kezdődött a komplex számítógépes szolgáltatás is: a TPA-t készítő KFKI-vel közösen alkalmazói és rendszer-szoftver-tanfolyamokat és dokumentációkat hoztunk létre.

Cs. Z.: Miskolcon a számítógépes munka később, a PC-k elterjedésével kezdődött, 1984-ben. Ennek oka egyszerű volt: nem volt nagygépünk. Ezután az egyesülés kínálata ugrásszerűen megnőtt. Jelenleg több mint 128-féle szoftver van forgalomban, természetesen a TPA-SzM 4 nagygépre valamint a Proper, MO8X, Apple, IBM PC és a Commodore sorozat gépeire. Még mindig a legszélesebb a kínálat Commodore szoftverekkel, a több mint 90 program között ügyviteli, matematikai, tervezési és mezőgazdasági feladatokat megoldó programcsomagot találhat a vásárló. Ezen túlmenően, az egyesülés keretén belül a hardverújdonosságokat is fejlesztjük és menedzseljük. A MOM megvásárolta a floppy- és winchesterlisztő egységünket, amelyet jövő évtől várhatóan belföldre és szocialista relációba nagy sorozatban fog készíteni.

µM: Sok hazai szoftver-előállító céget ért kimondatás vagy éppen kimondatlanul a vád: A „szerűség” terjesztő. Hozzájárulnak ahhoz, hogy a számítástechnika „ámitástechnika” legyen, azaz segítik a vállalatokat abban, hogy kicsiny kapacitású személyi számítógépekre, például C—64-re vagy Sinclair Spectrumra építhessék számítástechnikai rendszerüket. Már a kezdet kezdetén nyilvánvaló, hogy ezek a szinte játékszámba menő — bár olcsó — gépek nem alkalmasak komoly feladatok ellátására. Így a vállalat kipipálja a megoldott feladatot, de az megmarad, sőt a helyzet súlyosbodik...

Cs. Z.: A probléma valójában létezik, de... Ez viszont a vállalatokon múlik. Ugyanis, amikor felkérnek bennünket egy probléma megoldására, megvizsgáljuk a cég adottságait, s több változatban megoldási javaslatot teszünk. Általában egy olcsó — szükségmegoldásnak is felfogható —, egy közepes és egy a hosszú távú igényeket is figyelembe vevő változatot. Mi a javaslatainkban legalább a közepes kategóriájú PC-k beszerzését javasoljuk, pl. IBM vagy APPLE, esetleg IBM kompatibilis más gépen alapuló rendszert. Ilyenkor a válasz ismert: Sajnos nincs pénz... Így maradnak a C—64-ek vagy valami hasonló megoldás. Ilyenkor azonban nem szabad csodálkozni, hogy a vállalatnál rövidesen eljutnak a kérdésben felvetett problémához. S az egyesülés is csak azt szállíthatja, amire megrendelés és fizetőkészes kereslet van...

K. I.: Ennek ellenére 1986—1987-ben szeretnénk teljes egészében áttérni a gazdasági célú programoknál, de az ipari és a mezőgazdasági egyéb szoftvereknél is az IBM PC és azzal kompatibilis

gépek alkalmazására. Így már a javaslatban sem szerepel egy idő után Commodore 64 mint alternatíva. Bár feltétlen érdemes megjegyezni, hogy a kicsiny, mondhatni játékgépeken futtatott komoly felhasználói programok elkészítése csak igen fejlett programozói ismeretekkel, nagy rutinnal és gyakorlati eredménnyel. Ugyanis egy olyan program, amely simán és gyorsan lefut egy IBM PC XT-n, nem fér bele egy C—64-be. Ilyenkor rafinált módon szét kell törölni a programot, s olyan fagókat kell alkalmazni, hogy a folytonos háttértárháznál mellett a program futása elvisehető ideig tartson, ne legyen lassú. Ha egy ilyen „turbó” programot összehasonlítunk a megfelelő gépen futtatottal, annál alig lassúbb. Ez a programozói tudást dicsejri. Sőt alkalmat adhat a nagyobb gépek ilyesfajta „megpatkolására”. Ez feltétlen előrevizsi a programozási technika fejlődését.

Cs. Z.: A sokat emlegetett „szerűség” tehát a pilanatnyi kisebb beruházási igény következménye. Ugyanis jelenleg senkit sem érdekel, hogy egy vagy két esztendő múlva esetleg nagyságnedekkel nagyobb összeget kell majd beruházni, mint most egy drága, de hosszú távra előrelátó beruházás igénye lett volna. Emellett a szemlélettel is baj van. Legtöbb helyen elzárkóznak a számítástechnika alkalmazásától. Ha a közép-korban lennének, akkor biztosan az „ördög művének” kiáltanák ki sok helyen. Rébusznak veszik a gép használatát, s nem is akarják megtanulni az alkalmazást. A kitarítás azonban itt is sikerre vezet. Ilyen esetünk volt az egyik észak-magyarországi vállalattal. Amikor először felkerestük őket, finoman, de eltanácsoltak bennünket. Második alkalommal „véletlenül” egy Commodore 64-gyel és egypár játékgéppel jártunk arra. Mivel az egyik ottani vezető érdekelte, hogyan játszik egy bizonyos, általa kedvelt kártyajátékot a gép, hajlandó volt kipróbálni. Mondanom sem kell, megszerette s rövid időn belül egy szerény mértékű vállalati megrendelést tudhatunk a magunkénak.

µM: Főleg milyen korúak ötleteivel foglalkoznak?

Cs. Z.: Nálunk is megfigyelhető az a tendencia, amellyről olvashatunk a világsajtóban. Egyre fiatalabbak azok, akik a szoftverötleteikkel megkeresnek bennünket. Egyik legfiatalabb szerzőnk 12 éves(!), s sok segédprogramot készített el öreg programozókat megszerényítő ügességgel édesapja programötlethez. Éppen ezért nagyon nagy tartalomok vannak az országban. S ezért kérjük azokat, akik úgy érzik, hasznosítható szoftver- vagy akár hardverötletük, találmányuk van, keressenek fel bennünket. (Cím: Alkotó Ifjúság Egyesülés, Bp. V., Garibaldi u. 2. Számítás-

technikai irda. Tel.: 112-666 vagy Miskolc, Győri kapu 21. Tel.: 85-810.) Itt megvizsgáljuk ötleteiket, és ha értekesítő, menedzseljük. S csak az ötlet jósága a mérvadó, korlátár, képzettség, szakma nem számít!

K. T.: Az egyesülés jelenleg még nem rendelkezik külkereskedelmi joggal, de ha szükségesnek látszik, akkor kérvényezni fogjuk. Gazdasági rendszerünk annyira eltér a nyugatitól, hogy ilyen témájú termékeinket nem tudjuk ott értékesíteni. Ennek ellenére felmerült a gondolat a rendelkezésre álló hatalmas szellemi bázis jobb kihasználására, például bérmentés keretében. De a kilépés a világgpiaca köréből áll. Például az észak-magyarországi területi képviselőt szoros kapcsolatokat épített ki az NSZK-beli Kornhoffer céggel, amely a Multitec rendszert képviseli. Az így szerzett tudás is növeli a munka színvonalát.

µM: *S a gyakran változó gazdasági környezet, a dinamikus átalakuló pénzügyi szabályozórendszer nyenyiben befolyásolja terveket? Korábban prosperáló kisvállalkozások — a számítástechnikában van belőlük egynehány — ennek csödbe, változnak az adók, nőnek az elvonások.*

Cs. I.: Mi az északi régióban a nyitással a kisvállalkozások felé megpróbáljuk megakadályozni, hogy az ott kialakult szellemi tőke, gyártási tapasztalat veszendőbe menjen a gazdaság számára. Tudomásul vesszük azt, hogy a szabályozó-változások tendenciái a magán kisvállalkozások részleges állami — kiszövetkezeti — keretekbe történő terelésére irányulnak. A miskolci tanácsal az Alkottó Ifjúság Egyesülésé köztött egy olyan együttműködési megállapodást, hogy számos kihasználatlan ipari és egyéb épületet megkaphatunk, ahol elhelyezhetjük a velünk együttműködő kisvállalkozásokat akár gazdasági társaság, akár más formában történő együttműködés keretében. Így kialakulhatnak az éltechnológia ottani bázisai is. Akik túlélték a szabályozás változásainak viharait, azoknál feltétlenül egy olyan innovatív képesség kapcsolódott a vállalkozókészséggel és szakmai tudással, ami létfontosságú.

K. I.: Ami viszont az ösztöndíjakat illeti, ott egy kedvezőtlen változás bőszítette fel a szoftverek szerzőit. Egy újrendelet alapján a Szerzői Jogvédő Hivatalon keresztül kifizetett jogdíjak ezután 40% TBK- és 10% béradó terheli. Ezért, hogy rentábilis legyen a forgalmazásunk, csökkenteni kell a szerzők díjazását, módosítani a szerződíjást. Csak az a megfogandó ezzel a szabályozással, nem lőtt-e öngölt a rendelet megalkotója: a költségvetési bevétele növekményenél nem jóval nagyobb-e az a kár, amelyet a nem hasznosított szoftverek okoznak?

µM: *Vizonylag kevés szó esett az elektronikáról, a hardverről. Azok innoválásában mennyire képesek részt venni?*

Cs. Z.: Mint említettem, a városi tanácstól kapott épületekben szeretnénk megkezdeni egy nagyobb mérvű fejlesztéssel összekapcsoló gyártást. Elsősorban az alapgépek bizonyos kiegészítőit, például illesztőegységeket akarunk készíteni. De már az egyesülés most is mutathat fel e területen is sikereket. Ilyen, nagy érdeklődésre számot tartó produktumunk a COMES illesztőegység, amely a Commodore és más e kategóriába tartozó számítógép és az IBM, illetve ESZR rendszerek nagygepei közötti kommunikációt biztosítja. Vagy a másik, a négyterminális multiplexer,

amely egy bérlet vonal és egy modem használatával négy terminál számára szükséges adatátviteli feladatokat képes ellátni. Ez a mostani vonalinséges időkben szinte megfizethetetlen előny. Így alakult ki a jó kapcsolat is a Postával, hiszen ők nemcsak szoftverjeinket használják, hanem e termékünkkel bizonyos felhasználói nyomást is levelesznek a vállalkozó. Így nem csoda, hogy a multiplexert ők is ajánlják a felhasználóknak.

µM: *Nem érzik az egyre szaporodó számítástechnikai kis- és nagyvállalkozások okozta konkurenciát?*

K. I.: Megpróbálunk mindennel foglalkozni, amivel érdemes. Hazánkban szerencsére elég széles a versenypálya, egymás mellett van még elegendő hely a piacon. Bár vannak azonos tevékenységi formáink, mégsem vagyunk a szó hagyományos értelmében konkurencék. Egyesítjük az általában „tisztán” előforduló vállalati funkciókat: egyaránt vagyunk műszaki fejlesztővállalat építő, mint innovációs szakcég, lizingbeadó és szakértő-szervező szakvállalat. Ugyanakkor a kis egységekből álló felépítés, a fiatal szakembergárda nagyfokú rugalmasságra ad módot. S csak a produktumot néző szemléletünkkel a merítési bázisunk is nagyobb, mert hozzáink be mer jönni egy olyan fiatal szakmunkás, szakember, esetleg tanuló, aki nem tenné be a lábát például egy mammutvállalathoz, hogy kilincseljen ötletétel. A fiatalok vállalata vagyunk, s megpróbálunk ekképp cselekedni.

µM: *Miként kapcsolódnak be az oktatásba, a számítástechnikai ismeretterjesztésbe?*

K. I.: Mi szereltük fel a Szabolcs Szatmár Megyei KISZ-bizottság számítástechnikai klubját, valamint a politikai képzési központot is. Az természetes — hiszen van némi közünk a fiatalokhoz és a KISZ-hez! — hogyha segítséget kérnek tőlünk, megpróbálunk tenni valamit, például a hardver árérték segítjük a KISZ számítástechnikai klubját. Az NJSZT és a KISZ Heves megyei számítástechnikai napján éppúgy ott voltunk, mint az augusztusi miskolci helyi ipari kiállításon és vásáron észak-magyarországi területi képviselőink révén. Ilyenkor nemcsak a szakemberek, hanem az érdeklődőket is várjuk, mert szeretnénk megismertetni és megkedvelteni a számítástechnikát.

Cs. Z.: A múlt esztendő végén szűkebb pátriánkban, azaz itt Miskolcon 20 általános iskolának telepítettünk megrendelésükre Commodore 16-os gépet és a szükséges perifériákat, gondoskodva a szükséges betanítás és más feladatokról is. Ezenkívül keressük a lehetőséget, hogyan tudnánk részt venni tevékenyen tantárgyi és egyéb oktatóprogramok írásában. Eglyelőre itt még kétségek merülnek fel: az egyes tantárgyokhoz, például földrajzhoz, nyelvhez, biológiához, kémiához, milyen típusú programokat kell írni. Ugyanis oktatási anyagunk, iskolarendszerünk más országoktól eltérő lévén, nem lehet azok tapasztalatait adaptálni. Nekünk is végig kell járni a megismerés útját. Az iparban ez a tevékenységünk már régebben folyik, mert tanítjuk a gépek kezelését, s bár nem adhatunk bizonyítványokat, mégis sikerrel és nagy érdeklődéssel tartjuk tanfolyamainkat. Ez a továbblévő útja is e rendszerben: bekapcsolódnai a számítógép alkalmazását oktató tanfolyami rendszerbe, programba.

KIS JÁNOS

A játékprogramozás technikája

A játék irányítása

A 36. listában látható rész a helikopter sebességét szabályozza. Ha a botkormány tűzelőgombját lenyomjuk, a 65280 memóriacímeken a legelső bit értéke 0 lesz. Ezt észlve a program a sebesség (SP) egyébként 3 egység értékét 8-ra növeli („turbó”-érték). Gyors mozgás esetén az üzemanyag is gyorsan fogy (FU csökkenése 1 egység, egyébként 0,75), és a helikopter útján történő terület is gyorsabban változik.

A 37. listában a botkormány állapotát vizsgáló rész látható. A Dragon, az Apple, az IBM PC gépek ún. analóg botkormányt használnak. Ezeknél a bot irányával két egymásra merőlegesen elhelyezett potenciometer skáláján mozgatjuk a tolókat. Ez tehát két változó ellenállásérték ad a két, X, illetve Y irányú elmozdulással többkevesebb arányos ellenállásérték a számítógép számára szolgáltatott analóg jelpár. A Dragon analóg bemenetei 6 bit felbontásúak, vagyis 1:63 tartományt fognak át. A program ezt nem használja ki, csak (az ún. Atari típusú) kapcsolós botkormányoknál, amilyen a Commodore gépek is használnak) a 8 fő irányba tett elmozdulásokat vizsgálja. Így az X irány (J):

— balra J < 3,
— jobbra J > 60,
— nincs elmozdulás $3 \leq J \leq 60$.
Az Y irány (J1) hasonló. A botkormány a JOYSTK—BASIC utasítással olvasható be.

A 800-as sorban azt vizsgáljuk, hogy földet ért-e a helikopter. Amennyiben nem, úgy a lista hátralevő részét és a 38. listában szereplő részt átugorja. A lista maradékában azt vizsgáljuk, hogy a földet érés jó helyen történe-e meg (itt $94 < X < 106$ értékű)? Itt csak függőleges mozgás lehetséges.

A 38. listában a földet érés miatt újrarajzoljuk a repülőteret négyzetgöget.

A 39. listában a rajzteret korlátozzuk, és megakadályozzuk, hogy ezt a területet a helikopter elhagyja.

Az utolsó, még nem vizsgált ténykedésünk a repülőterén földet ért helikopter üzemanyaggal történő feltöltése, növekvő magasságra hang adása mellett. A hang jelzi a feltöltöttséget. Túltölti nem lehet, mert az üzemanyag mennyiségét a 920-as sorban korlátozzuk. Ez a 40. listában történik meg.

A szerző szerint a programozás „konstruktív tudomány”. Hibás tehát az az elképzelés, mely szerint az ismeretek átadása előbb-utóbb valamiféle „receptgyűjtemény” betanításává egyszerűsíthető. *A jó példák tanulmányozása azonban igen hasznos lehet.* A könyvben bemutatott programok egyúttal *programolvasási gyakorlatul* is szolgálnak. A szerző az olvasót bevonja a programok fokozatos kialakításába, különböző „pillanatfelvételeket” mutat be egy-egy program fejlődéséről, a lépésenkénti finomítás módszeréről. A bemutatott programok számítógépen valóban le is futnak.

Itt most figyelemfelkeltőül a könyv „Rendezések” című fejezetéből mutatunk be tanulságos részeket. A rendezés (sorting) a dolgok egy meghatározott halmazának valamilyen sorrend szerinti átrendezését jelenti. A rendezés feladata az, hogy megkönyítse az elemek későbbi keresését. A rendezés kiemelkedően fontos tevékenység — különösen az adatfeldolgozásban —, így különösen ideális téma az algoritmusok sokféleségének bemutatására: például ugyanazt a feladatot végző algoritmusok közül több bizonyos értelemben optimális és a legtöbb algoritmusnak van valamilyen előnye a többivel szemben. Például az egyik csak kis mértékben rendezetlen tömböknél, a másik a teljesen rendezetleneknél ideális.

A rendezőmódszerekét általában a következő két kategóriába szokás sorolni:

- tömbök rendezése és
- (soros) fájlok rendezése.

E két kategóriát gyakran nevezik *belső*, illetve *külső* rendezésnek is, utalva arra, hogy a tömbök rendezése a számítógép gyors, véletlen (random) elérésű, „belső” tárában zajlik, amíg a fájlokat lassúbb, de tagasabban „külső” tárra szokás elhelyezni, melyek mechanikusan mozgatható berendezések (lemezek és szalagok). A kétfajta módszer különböző, az alábbiakban csak a tömbök rendezéséből adunk ízelítőt.

A könyvben bemutatott rendező algoritmusok olyanok, amelyek a tételek átrendezését saját helyükön hajtják végre: tehát a rendezéshez nem vesznek igénybe újabb tárterületet lefoglaló segéd tömböt vagy tömböket. Az algoritmusok elterének egymástól például a rendezéshez szükséges összehasonlítások és cserék számában. Minél kevesebb összehasonlítással és cserével jut el az algoritmus a teljes rendezésig, annál jobbnak minősíthető. Először egy ún. közvetlen módszert mutatunk be röviden, jelzészérien. Ez a módszer nem a legjobb, de rövid, egyszerű és alkalmas a rendező elvek jellemzőinek megvilágítására. Az itt bemutatott rendezés közvetlen beszúrással történik. A rendezendő tételeket elvben két részre osztjuk:

- az a_1, \dots, a_n fogadó sorozatra és
- az a_1, \dots, a_n fogadó sorozatra.

Minden lépésben $i=2$ -től egészevel n -ig, kiemeljük a leadó sorozat első elemét és átesszük a fogadó sorozatra, *beszúrva* őt a megfelelő helyre. Először a tömböt a második tételnel választom ketté, először a második elemet emelem ki és *szúrom* be a most még csak egyetlen elemet (az első) tartalmazó fogadó „sorozatba”.

Az első lépés után a_1 és a_2 sorrendje már helyes lesz. Ezután lépek egyet jobbra és a harmadik elemnél választom ketté a tömböt fogadó és leadó részre. Kivielem a harmadik elemet és be-

szúrom a már rendezett a_1 és sorozatba. És így tovább:

for $i:=2$ to n do

begin

$x:=a[i]$; [„kivielem” a soron következő tételt]

ide jön az a rész, amelyik x -et beszúrja az a_1, \dots, a_{i-1} fogadó sorozat megfelelő helyére

end.

A bekeretezett részt, a *beszúrás* úgy valósítjuk meg, hogy a kiemelt tételt „visszafelé”, a fogadó sorozat végétől az eleje felé haladva sorban összehasonlítjuk a fogadó sorozat egyes tagjaival egészen addig, amíg x helyét meg nem találjuk. Az alábbi programban vegyük észre, hogy két különböző feltétel is van, melyre ez a visszafelé haladó „rostálós” eljárás befejeződik:

1. Találunk egy a tételt, amely x -nél kisebb.
2. Elértünk a fogadó sorozat bal szélére (mert a kiemelt tétel helye a fogadó sorozat elején van).

A második esetet a program az ún. „stráza” (örszem, sentinel)-technikával kezeli. Az alábbi program éppen a kiemelt $x:=a[i]$ elemet állítja „örszemként” a sorozat elejére ($a[0]:=x$). Emiatt a rendezéshez igénybe vett tömb egy tétellel hosszabb, mint a rendezendő sorozat (éppen az a $[0]$ -val).

Tehát a rendezés közvetlen beszúrással:

```
procedure közvetlen beszúrás;
type tétel = record kulcs: integer;
              [egyéb komponensek]
            end;
index = 0 .. n; [Fügelem, a 0 a stráza helye!]
var ij: index;
    x: tétel;
    a: array [index] of tétel;
```

begin for $i:=2$ to n do [a fogadó és leadó sorozat határa]

```
begin  $x:=a[i]$ ; [a kiemelés]
       $a[0]:=x$ ; [a „stráza” felállítása]
       $j:=i-1$ ;
      while  $x.kulcs < a[j].kulcs$  do
      begin  $a[j+1]:=a[j]$ ;
             $j:=j-1$ ; [haladunk visszafelé a fogadó sorozatban]
      end;
       $a[j+1]:=x$ ; [itt a beszúrás]
```

end

end

A könyv elemzi a közvetlen beszúró algoritmust: megállapítja, hogy

- az i -edik „rostálás” során a kulcs-összehasonlítások C száma legalább 1 és legfeljebb $i-1$, tehát némi egyszerűsítéssel átlagosan $1/2$.
- A mozgatók (tehát a tételek tárolásának) száma:

$M = C + 2$ (a strázát is beszámítva).

- Az összehasonlítások és a mozgatók teljes száma így:

$C_{n-1} = n - 1, \quad M_{n-1} = 2(n - 1),$

$$C_n = \frac{1}{4}(n^2 + n - 2), \quad M_n = \frac{1}{4}(n^2 + 9n - 10),$$

$$C_{n-1} = \frac{1}{2}(n^2 + n) - 1, \quad M_{n-1} = \frac{1}{2}(n^2 + 3n - 4).$$

A legkisebb értékek akkor adódnak, ha a tételek eredetileg sorrendben voltak; a legrosszaság eset (a „worst case”) az, ha a tételek eredetileg fordított sorrendben álltak. Ebben az értelemben a fenti algoritmus *viselkedése természetesen mondható*. Látható, hogy a megadott algoritmus egyúttal *stabil* rendezőeljárás is: az azonos kulcsú tételek eredeti sorrendje nem változik.

A közvetlen beszúró algoritmus — úgy tűnik — egyszerűen javítható, ha észrevesszük, hogy az a_1, \dots, a_n fogadó sorozat, ahova az új tételt beszúrjuk, már rendezett. Kézenfekvő a „bináris” keresés alkalmazása, melyben a sorozat felezését alkalmazzuk, amíg a beszúrás helyéhez el nem érkeztünk:

```
procedure bináris beszúrás;
type [mint az előző algoritmusnál]
var ij,lr,m:index;
    x:tétel;
    a:array [index] of tétel;
begin
  for  $i:=2$  to  $n$  do
  begin  $x:=a[i]$ ;  $l:=1$ ;  $r:=i-1$ ;
        while  $l \leq r$  do
        begin  $m:=(l+r) \div 2$ ;
              if  $x.kulcs < a[m].kulcs$ 
              then
                 $r:=m-1$ 
              else
                 $l:=m+1$ 
        end;
         $a[j]:=a[l-1]$  downto  $l$  do  $a[j+1]:=a[j]$ ;
         $a[j]:=x$ ;
        end
  end
```

A *bináris beszúrás elemzése* (amit itt elhagytunk) azt mutatja, hogy az elért javítás csak az összehasonlítások számára vonatkozik, a szükséges mozgatók számában nem, így nem érhető el ugrásszerű javulás. Nem is beszélve arról, hogy egy rendezett tömb újarendezése ezzel több időt venne igénybe, mint közvetlen beszúrással. Ez a példa jól mutatja, hogy egy „kézenfekvő javításnak” gyakran kisebb jelentőségű a következményei, mint amit az ember első pillantásra remél, sőt néhány esetben (elő is fordult) a „javítás” valójában rontást eredményez.

A közvetlen beszúró rendezés továbbfejlesztését D. L. Shell vetette fel 1959-ben. A *Shell-rendezés* úgy működik, hogy először egymástól távolabb álló elemeket hasonlít össze és cserél ki, miután így végigpásztázta a tömböt, a távolságot csökkentik a két elem között és megismétli az összehasonlításokat és (szükség szerint) a cseréket. Az utolsó menetben szomszédos elemeket hasonlít össze és cserél ki. A csökkenő távolságokat („a főgyő növekményeket”) különböző módon meghatározhatjuk. Az alábbi program olyan, hogy a t darab növekményt („távolságot”) a h tömbben helyezhetjük el. Ahhoz, hogy az algoritmus működjék, ki kell előtéríteni az alábbi követelményeket:

$$h|t = 1 \text{ és } h_{i+1} < h_i$$

Természetesen minden egyes menetnek saját strázát kell állítania. Ezért a tömböt nemcsak

okkal ezolított egy másik (D. Miller könyvről az előző számban adunk tudósítást), ma is nagyon aktuális, jó számítástechnikai „olvasókönyvet”, mely nemrég magyarul is megjelent. (N. Wirth: Algoritmusok + Adatstruktúrák = Programok; Műszaki Könyvkiadó 1982, fordította: Lehel Jenő okl. matematikus, lektorálta: Bőszörményi László, okl. villamosmérnök). A magyar fordítás az üzletekben rég elfogyott, a könyv hiánycikk a könyvtárakban is, az angol nyelvű kiadás (N. Wirth: Algorithms + Data Structures = Programs; Prentice-Hall, Inc. 1976.) is csak nehezen hozzáférhető. A könyv, mely a zürichi Műszaki Egyetemen (az ETH-n) tartott előadások összefoglalása, a PASCAL-nyelvet használja a programok bemutatására. A könyv egyike a számítástechnika alapműveinek. Jó lenne gyorsan ismét kiadni. Ezzel különösen az egyetemi hallgatóságnak tenéknék nagy szolgálatot.

```

begin a [j+k] := a [j]; j:=j-k
end;
a [j+k] := x
end
end

```

Akik a μ Magazin múltkori számában olvastakon felbuzdulva szerencsésen meg tudták szerezni D. Miller könyvét a C-64 adat-fájliáról, azok a „Home inventory System” nevű nagyobb programban megtalálhatták a fenti program egyik módosított változatának BASIC-ben írt megfelelőjét, mely így néz ki:

```

25000 rem** -- Shell-Metzner-rendező --
szubrutin --**a rendezendő füzérek
(tételek) az a$ elnevezésű változóban
vannak. A tételek száma:
n
25010 m = n
25020 m = int(m/2); rem felezem a sorozatot
25030 if m = 0 then 25150: rem VÉGE
25040 j = 1 : k = n - m
25050 i = j
25060 z = i + m
25070 if a$(i) < a$(z) then 25120
25080 t$ = a$(i); a$(i) = a$(z); a$(z) = t$
25090 i = i - m
25100 if i < 1 then 25120
25110 goto 25060
25120 j = j + 1
25130 if j > k then 25020
25140 goto 25050
25150 return

```

A működés magyarázata helyett álljon itt az algoritmus folyamatábrája. Ezzel együtt jó kis szelmei torna lesz a Shell-Metzner-rendező működésének megértése. Az algoritmus a „távolságot” így állítja be PASCAL-ra fordítva:

```

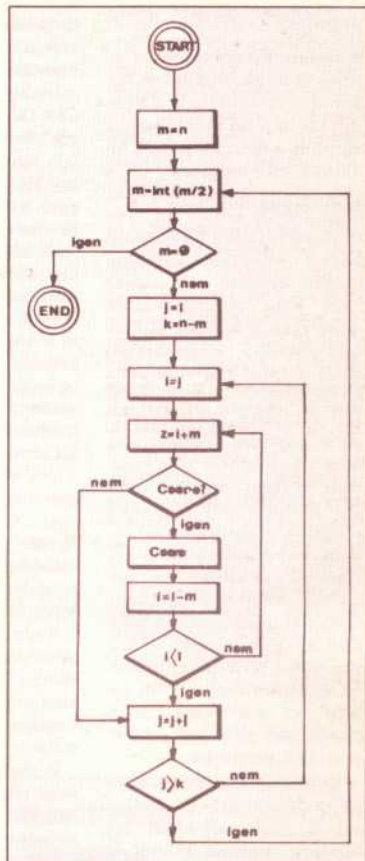
m := n; [ahol n a rendezendő tételek száma]
while m >= 1 do
begin h [j] := m div 2;
i := i + 1
end;
t := i;

```

A h tömb ilyen feltöltése N. Wirth szerint nem a legjobb. Ezt írja ugyanis könyvében: talán megfelelő, hogy optimális esetben a növekmények nem lehetnek egymás többszörösei.

Ennek ellenére a bemutatott rendező algoritmust bátran ajánlhatjuk C-64 tulajdonosok részére.

Érdekes befejezésül egy pillantást vetni arra a táblázatra, melyben N. Wirth összefoglalja a különböző rendező algoritmusokak végzett végrehajtásiidő-mérések eredményét:



A „Shell-Metzner-sort” algoritmus folyamatára

Az algoritmus neve	A kiindulási állapot					
	rendezett	véletlen	fordított	fordított	fordított	fordított
Közvetlen beszúrás	12	46	366	1129	704	2150
Bináris beszúrás	56	76	373	1105	662	2070
Shell-rendező	58	186	127	373	157	435
Gyors-rendező	31	55	60	137	37	75

A táblázatban megadott számok a futási időt mutatják n=256 tétel rendezésére egy CDC 6600-as gépen ezred másodpercben. A bal oldali számok csak kulcsok rendezésére, a jobb oldali kulcsot és adatokat is tartalmazó tételekre vonatkozik.

Itt is látszik, hogy a bináris beszúrás alig javít a közvetlen beszúrásnál. Az igen jónak bizonyuló „gyors rendezőt” az érdeklődők és a szerencsések megtalálhatják a könyvben.

egy „a[0]” komponenssel, hanem h darab újabb komponenssel fogjuk kiegészíteni. Ezért n darab tétel rendezéséhez az alábbi módon deklarált tömböt fogjuk használni:

```
var a: array [-h..n] of tétel.
```

Ha tehát például egy n=100 tételből álló tömböt kívánunk rendezni úgy, hogy az első menetben az egymástól 49 „távolságra” lévő elemeket szeretnénk összehasonlítani (azaz, ha h[1]:=49 értékkel dolgozunk), akkor a rendezéshez 100+1+49=150 tételnek megfelelő méretű tömbnek kellene a tárbán helyet biztosítani.

A h [1..t] tömbben tárolt lépések optimális meghatározása néhány igen nehéz matematikai problémát vet fel. Nem ismeretes, hogy milyen sorozat adja a legjobb eredményt. Knuth a következő sorozatot javasolja:

```
..., 121, 40, 13, 4, 1
```

Ajánlja még a

```
..., 31, 15, 7, 3, 1
```

sorozatot is.

Az alább bemutatott programban a

```
..., 9, 5, 3, 1
```

sorozattal fogunk dolgozni. Íme a program (helyesebben a procedúra — mint a PASCAL szintaxisából ismeretes, a procedúra csak egy program keretében fut le, önállóan nem futtatható —!):

```

procedure Shell-rendező;
const t = 4; [négy „menetben” fogunk rendezni]
var i, j, k, s: index;
x: tétel; [az előző példából ismert módon]
m: 1..t; [a h tömb indexe]
h: array [1..t] of integer;
[h tartalmazza az egymás utáni rendező menetekben alkalmazott „távolságokat”]
begin [először a h tömb elemeinek adunk értékeket]
h [1] := 9; h [2] := 5; h [3] := 3; h [4] := 1;
for m := 1 to t do [a különböző távolságok beállítására]
begin k := h [m]; s := -k; [a strázsá helye]
for i := k + 1 to n do
begin x := a [i]; j := i - k;
if s = 0 then s := -k; s := s + 1;
a [s] := x;
while x.kulcs < a [j].kulcs do

```

Sínstruktúrájú háló

Centralizált vezérlésű sín-topológiájú helyi hálózatok esetén az alábbi hálózatelési módszerek terjedtek el

- CSMA/CD
- Round robin
- Vezérelt jelzés (token bus)

A topológiájú hálózat az egyes munkaállomások párhuzamosan csatlakoznak. A hálózat kábelét a két végén illesztve zárják le a zavaró reflexiók kiküszöbölése érdekében.

A hálózati kábel lehet koax vagy sodrott érpár. Az ilyen struktúrák jellegzője, hogy a hálózaton egyidőben csupán egyetlen adatsomag lehet jelen. A sínstruktúrák egyszerűen bővíthetők (átkonfigurálhatók), megbízhatóságukat a kábel folytonos volta nagymértékben növeli (nem kell a kábelben állomásonként aktív elemeket közbeiktatni, mint a gyűrűs hálózatok esetében), az egyes állomások hibája nem teszi üzemképtelenné a rendszert. A sínstruktúrájú hálózatok leggyakrabban véletlenszerű hálózateléréssel működnek. Az ilyen jellegű hálózatelési módszerek a hálózathoz való hozzáférés, az üzenetsomag-ütközés felismerése, az üzenet megismétlése, valamint az üzeneteknek a hálózathoz való eltávolítása szempontjából vizsgálják.

Sínstruktúrájú hálózatokban léteznek időosztásos rendszerek is, ezek működését egy központi ütemező időzíti, az egyes munkaállomások üzenetei a hálózathoz adott hosszúságúra törölve kerülnek továbbításra (csomag), a sikeres átvitelt a vevőoldal egy időkorláton belül nyugtázza, ennek hiánya (negatív nyugta) jelzi az üzenetütközést, ilyenkor a megsemmisült üzenet forgalmazását meg kell ismételni.

Időosztás nélküli rendszerekben a továbbított üzenetek hosszúsága tetszőleges, az ilyen rendszer teljesítménye valamivel kisebb az időosztásosnál.

CSMA/CD

A CSMA/CD rendszer a „figyelj, mielőtt beszélsz” elvet valósítja meg, azaz a forgalmazni kívánó állomás forgalmazás előtt „befigyel” a csatornába és csak akkor kezd el forgalmazni, ha a csatornát szabadnak találja, nehogy az üzenetek megsérüljenek. Innen származik az eljárás neve is, CD: Collision Detection = ütközésértékelés. Ha a vonal foglalt, úgy többfelét is tehet, pl. egy kis kivárás után újra befigyel a vonalba. A megoldás hátránya, hogy a kivárások miatt a csatorna átbocsátóképesége csökken, előnye, hogy az ütközések száma csökkenthető (ha két üzenet egyszerre kerül rá a közös átviteli közegre — kábelre — azok ott ütköznek és megsérülnek, ezért mindkét üzenet forgalmazását meg kell ismételni). A megoldás ott alkalmazható előnyösen, ahol az átvitel lassabb, mint amit a rendszer egyébként megenged.

Léteznek olyan rendszerek is, melyek a vonalfoglaltság észlelése utáni kivárás idejét valamilyen algoritmus (szabály) szerint változtatják (pl. egy-egy sikertelen kísérlet után növelik a kivárás idejét), illetve egy meghatározott számú kísérlet után abbahagyják a sikertelen forgalmazási kísérletezést.

A csatornafoglaltságot alap-sávú rendszerekben a csatornán forgalmazáskor jelenlévő impulzussorozat, szélessávú rendszerekben a vivőfrekvencia megléte jelenti, az ütközés esetén a csatornán feszültségcsúcs jelenik meg, melynek nagyságát a kábelhossz is befolyásolja; ez a tény adott esetben a maximális kábelhosszat is korlátozhatja.

Szélessávú rendszerekben az adatsomagok ütközését az elküldött adatsomag, illetve a visszakapott nyugta bitenkénti összehasonlításával érzékelik.

Round robin eljárás

Az ilyen elven működő rendszerben az egyes munkaállomások adási, illetve vételi pontjaikon kapcsolódnak egymáshoz. Működésük során az állomások lehetnek:

- tétlen állapotban (nem akar forgalmazni)

- a forgalmazás megkezdésére váró állapotban

- a forgalmazás folytatására váró állapotban.

A forgalmazásra váró állomások csatornakérőjelet tesznek a csatornára, majd „befigyel”-sel ellenőrzik annak foglaltságát. A csatorna szabaddá válása után a vonalat, ezzel a forgalmazás jogát mindig a legkisebb indexű állomás kapja meg (a magasabb indexű állomás érzékeli a kisebb indexű csatorna kérőjelet, így a csatornát foglaltnak érzékeli.)

Az ismertetett eljárásnak létezik egy prioritáson alapuló változata is, ebben az egyes állomások különböző prioritási szintekhez vannak hozzárendelve és a csatorna-hozzáférés jogát a forgalmazásra váró állomások közül mindig a legmagasabb prioritási szinthez tartozó kapja meg.

Vezérelt jelzés (token bus)

A sínre az egyes munkaállomások egy telepítés által meghatározott sorrendben (fizikai sorrend) kapcsolódnak rá, de ugyanakkor minden állomás rendelkezik egy logikai azonosítóval is (a kettő nem szükség-szerűen azonos). Az egyes állomások hálózathoz való hozzáféréset azok logikai sorrendje szerint meghatározott időzítéssel hozzáférési jog szabályozza (token). A hálózat úgy működik, hogy a vezérelt jelzést birtokló állomás adott ideig birtokolja a hálózatot, ezalatt forgalmazhat, majd ezt követően a hálózat birtoklásának joga továbbkerül a következő logikai című állomásra.

A hálózati csatolóegység

A helyi hálózat munka- és server állomásai a hálózati kö-

zetre (kábel) egy csatolóegység segítségével csatlakoznak. A csatolóegység a munkaállomás (személyi számítógép vagy képernyős terminál) belső buszára vagy szabályos interfészére (V24) csatlakozik. A személyi számítógépes helyi hálózatokban az első megoldás az elterjedtebb, itt a csatolóegység nyomtatott áramkörtárcsát tartalmaz, amelybe a PC-be dugaszolják be.

A csatolóegység nagyintegráltságú elemeket (speciális protokolvezérlő áramkörök, mikroprocesszorok, nagykapacitású tárelemek) tartalmaz, működését ROM-okba ágyazott mikroprogram vezérli.

A csatolóegység feladata a közvetlen hálózati forgalomvezérléssel kapcsolatos funkciók ellátása, pl. az adás, illetve vételirányú csatornacsatlakoztatás megvalósítása, vonalfoglaltság-figyelés, az adásismétlés szervezése, nyugtázás, címfelismerés, az üzenetsomag összeállítás, adatpufferelés, adatátvitel a PC, illetve a hálózat felé.

Ezen funkciók megvalósításával a csatoló tulajdonképpen a PC központi egységét tehermentesíti a periférius funkciók ellátása alól.

Az adatátvitel a helyi hálózatok esetén is az OSI nyílt architektúra 7 rétegű modelljének megfelelően valósul meg, a modell első 2—4 (fizikai, adatkapcsolati, hálózati, szállítási) rétegfunkcióját a csatolóegységben valósítják meg.

Érdekesebb alkalmazások

A világon jelenleg a legtöbb helyi hálózatot irodai és ügyviteli rendszerekben alkalmazták, de jelentős számban találhatók helyi hálózatok ipari alkalmazásokban is. Mivel az irodai alkalmazások a közismertebbek, a cikk inkább né-

k elérési módszerei

hány érdekesebb, megvalósított ipari alkalmazást mutat be, ezzel is szemléltetve a helyi hálózatok sokoldalú alkalmazhatóságát.

A General Motors 15 részlegét kapcsolta össze helyi hálózat segítségével. A hálózattal futószalagot, szerelőrobotokat vezérelnek, továbbá ezen hálózattal oldották meg a szereléshez szükséges alkatrészek teljes nyilvántartását is.

A General Electric rendszerre irodai, tervezést segítő és gyártási alhálózatokból áll. Az irodai rendszer közismert feladatokat lát el, a tervezést segítő rendszerrel gyártmánytervezést végeznek, ennek leterhelése a rendszer használatából kifolyólag lőkészítési és benne a gépi válaszdők is ennek megfelelően alakulnak (50—500 ms). A gyártási rendszerrel robotokat, különböző gépeket és kamerákat kapcsolnak össze, a hálózatot a fenti

funkciókkal egyidőben hangátvitellel is használják. A rendszer gépi válaszideje szélsőséges körülmények között 5—5000 ms között változhat.

Az egyes rendszerek esetében jelentős szerepe van a teljesítmény/ár hányadosnak, ami a legmegfelelőbb hálózat-elérési módszer megválasztásával is befolyásolható, ez is az egyik oka a CSMA/CD eljárás közkedveltségének (egyszerű, hatékony műszaki megoldás, készen kapható CSMA vezérlő tokok stb.). Ezt a megoldást alkalmazza egyébként a General Electric, a Data General, a HP és a Siemens is rendszereiben.

Az ipari alkalmazásbeli hálózatok zömmel szélessávúak és sinstrukturájúak, a hálózat az egész intézmény területére kiterjed, továbbá a fő funkciók (folyamatvezérlés) mellett számos egyéb szolgáltatást is nyújtanak az üzemeltetőknél

(kép- és hangátvitel). Az ilyen rendszereknek további előnye, hogy a főhálózat a kisebb alhálózatokkal egyszerűen összekapcsolható komplex rendszerrekké. Az ipari alhálózatok általában alapsávúak és 4-10 csomópontot (munkaállomás) tartalmaznak.

Az alhálózatokból felépített struktúra azért előnyös, mert vele a felhasználói igények egyszerűbben elégíthetők ki.

Az elkövetkezendő időszakban kialakítandó munkahelyeken — főleg valós idejű alkalmazásokban — várhatóan a nagysebességű adatcsatornák mellett több kisebb sebességű csatornát is tartalmazó hálózatokat fognak üzemeltetni. Ipari rendszerek esetében igen jelentős szempont a rendszer jól definiált válaszideje, főleg valós idejű alkalmazásokban, ilyen esetben célszerűbb a vezérelt jelzéses rendszer alkalmazása. Léteznek olyan rendszerek is, melyekben a forgalomtorlódás soha sem fordul elő, így állandóan maximális teljesítménnyel üzemelnek, a válaszdők

minden esetben jól definiáltak. Az IBM ipari alkalmazásokra készített rendszere sinstrukturájú, irodai alkalmazásokra készített hálózata gyűrűs struktúrájú. Az IBM megjelenése a helyi hálózatok területén egyúttal a megoldás „defacto szabvány”-nyá választ is jelentheti.

Ami az elkövetkezendő 4—5 év várható perspektíváját illeti, a kapcsolatteremtésben elsősorban a vállalatvezetés, az egyes részlegek és a vállalati nagyszámítógéppont összekapcsolása várható főleg alapsávú helyi hálózatokkal. Az alkalmazások terén elsősorban a megrendelés-felvétel, megrendelés-nyilvántartás, raktárkészlet-nyilvántartás, valamint a termelésütemezés „hálózatosítása” várható leginkább; főleg nagyobb intézményeknél ezekhez a rendszerekhez csatlakozhatnak a különböző géppel segített tervező-gyártó rendszerek (CAD—CAM), illetve az irodai apparátus munkáját kiszolgáló alrendszerek is.

CSEH KÁLMÁN

Tervezőintézet

országos
számítógépes
grafikai
nyilvántartási
rendszerek
fejlesztéséhez
keres
rendszertervező,
tervező
és programozó
szakembereket.

Jelentkezés:

569-122/218-as mellék

**Fejlesztési Intézetünkbe
felvételre keresünk
korszerű mikroprocesszoros terminálok
és berendezések HARDWARE—SOFTWARE-
fejlesztéséhez villamosmérnököket,
készülék konstrukciós kialakításához
felsőfokú végzettségű szakembereket.**

**RUGALMAS MUNKAI DŐ
KIEMELT FIZETÉS
MAGAS JÖVEDELEM**

Jelentkezés:
TELEFONGYÁR

1143 Bp. Hungária krt. 126-132.
Tel.: 634-330.



DÖNTÉSÉT MEGKÖNNYÍTJÜK!

**Széles választék, kedvező ár,
sokoldalú szolgáltatás!**

8 bites ALAPKONFIGURÁCIÓK

PROPOS—8 (CP/M vagy ezzel kompatibilis) operációs rendszerrel

M08X	250 000,— Ft
PROPER—8	250 000,— Ft

A konfigurációk érdekeltségi alap nélkül megvásárolhatók

16 bites ALAPKONFIGURÁCIÓK

PROPOS—16 (IBM PC XT vagy ezzel kompatibilis) operációs rendszerrel

PROPER—16/m	266 000,— Ft
PROPER—16/W1	439 000,— Ft
PROPER—16/W2	549 000,— Ft
PROPER—16/MEGA	699 000,— Ft
MÁTRIXNYOMTATÓ (MP—80, kvázigrafikus jelkészlettel)	30 000,— Ft
PROPOS—16 OPERÁCIÓS RENDSZER	29 000,— Ft

RENDELÉSEKET VESZÜNK FEL 1986. II. FÉLÉVRE

PROPER—16/MT (IBM PC AT kompatibilis operációs rendszer)

850 000,— Ft

PROPER—16/G (nagy teljesítményű grafikus munkahely)

1 500 000,— Ft

SZOLGÁLTATÁSOK:

12 hónap garancia, további 12 hónapra a hardver árának 6%-áért
hardver-szervizszolgálat
Világszínvonalú perifériák
Széles körű szofterválaszték
Kulcsrakész alkalmazói rendszerek
Hálózatba integrálási lehetőség
Országos szervizhálózat

KÉRJE RÉSZLETES ÁRJEGYZÉKÜNKET!



Számítástechnikai Kutató
Intézet és Innovációs
Központ
Budapest 1251 Pf. 19.

Információ:

SCI—L Számítástechnikai
Informatikai Fejlesztő Leányvállalat
Budapest, Iskola u. 10. 1011
Telefon: 153-204

AZ SZKI STABIL PARTNER!

SCITEL Számítástechnikai
Fejlesztő Leasing Leányvállalat
Budapest, Donáti u. 35—45. 1015
Telefon: 350-180/142

PTA-4000 ROM-leírás

A gép alapkiépítésben az M0 lapon az A000—FFFF címterületben tartalmaz ROM-ot. Ezt cím szerinti sorrendben tárgyaljuk.

Az A000—A01F címterületen a kiíratáshoz használt 1. segéd tábla található. Ezt a táblát a tábla címmutatója követi az A020—A07E címterületen. A 2., ugyanilyen célú segéd tábla a folytatás az A07F—A28A címterületen.

A következő nagy terület a gép alaplakódését kiszolgáló szubrutinoké az A28B—AFFF címterületen.

A B000—B0E9 az 1. kulcsszó táblázat helye. A táblázat belsejében jelentős nagyságú „üres” hely található, valószínűleg a későbbi bővítések céljára fenntartva.

Ezt egy rövidebb, újabb rendszerprogramokat tartalmazó rész követi a B0EA—B152 címterületben.

BASIC szubrutinok I.

A kulcsszavakhoz tartozó 1. szubrutincsoport a B153—B7FF címterületen található. Az itt levő szubrutinok kezdő-és végcímei az 1. táblázatban ismertettem.

A folytatás a 2. kulcsszó tábla a B800—B887 címterületen, ismét jelentős „üres” résszel.

Ezt az ún. ugrótábla követi. Bizonyos szubrutinok ezeken keresztül, kétlépcsős ugrással érhetők csak el. A címterület: B888—B8A5.

BASIC szubrutinok II.

A B8A6—BBD5 területen újabb kulcsszavakhoz tartozó szubrutinok vannak, melyek kezdő-és végcíme a 2. táblázatban található.

Ezután néhány, a kazettával dolgozást szolgáló szubrutin következik. A BBD6—BBF4 címterületen a szalagon az ún. „fejréz”-t előállító rész, a BBF5—BCE7 között a parancsok kivétel/beolvasását vezérlő szubrutin, a BCE8—

BD3B területen a „fej” kiírató/beolvasó, a BD3C—BDA3 területen az állományt kiírató szubrutinok vannak.

A BDA4—BDCB területen egy szubrutindarab található, bizonyára csak így fért el. A BDCC—BDE6 területen az egy bájtot kiírató szubrutin, majd a BDE7—BDEF területen ismét egy szubrutindarab, a BDF0—BDF8 területen az egy bájtot beolvasó szubrutin lelhető fel.

Ezután a külső készülékeket vezérlő parancsok következnek. Először az RMT a BEF9—BF10 címterületen, ezt követik a külső készüléket bekapcsoló (REMOTE ON), BF11—BF42 és kikapcsoló (REMOTE OFF), BF43—BF75 parancsok, majd újra egy szubrutindarab: BF76—BFFF.

3. kulcsszó tábla

Néhány rövidebb szubrutin következik; az Xreg töltése a 78F1 címen tárolt értéktől függően C001—C00D (a C000 címen egy állandó található), az Ureg töltése a 7883/7884 és a (7A07) 7885 címeken levő értéktől függően C00E—C01C, a TRACE-vektor, C01D—C01F. Ezt a parancsok kezdőcím táblája követi C020—C053, szintén jelentős bővítési lehetőség.

A 3. kulcsszó tábla az ezután következő rész a C054—C34E címterülettel.

Ismét rövidebb csoportokat találunk ezután. A BASIC üzenetei C34F—C36A, a DEF-billentyűk táblázata C36B—C385, egy adat C386—C38A, a vezérlőkódok (00—0F ASCII-érték) ugrótáblája, C38B—C3AA, egy általán nem ismert célú táblázat: C3AB—C3FF.

Utasítások

Indítás:

C400.

A BASIC-fordító kulcsszavak az Ureg-ben:

C401—C457.

Ezt követik az utasítások szubrutinjai, melyek kezdő-és végcímei a 3. táblázatban találhatóak:

C458—C9E3.

Az EDITOR-hidegindítás követi ezt:

C9E4—CA54.

A CLEAR ALL (CA) billentyű szubrutin belépőpontja:

CA55—CA57.

EDITOR-melegindítás (RUN-módban):

CA58—CA63.

CL-billentyű szubrutin:

CA64—CA79.

EDITOR-ugrás a mutató törlése nélkül:

CA7A—CA7F.

EDITOR-ugrás:

CA80—CB60.

RCL-billentyű szubrutin:

CB61—CB68.

A RESERVE-mód bekapcsolása:

CB69—CB9B.

A SHIFT-billentyű bekapcsolása:

CB9C—CB9F.

A MODUS-billentyű bekapcsolása:

CBA0—CBC6.

Az INS-billentyű bekapcsolása:

CBC7—CBCE.

A DEL-billentyű bekapcsolása:

CBCF—CBE3.

A CURSOR RIGHT-billentyű bekapcsolása:

CBE4—CC21.

A CURSOR LEFT-billentyű bekapcsolása:

CC22—CC37.

A CURSOR UP-billentyű bekapcsolása:

CC38—CC47.

A CURSOR DOWN-billentyű bekapcsolása:

CC48—CC00.

Az ENTER-billentyű bekapcsolása:

CCC1—CD70.

Az OFF-billentyű bekapcsolása:

CD71—CD88.

Az ERROR 1-üzenet ugráshelye:

CD89—CD8A.

Az ERROR-üzenetek (a kódok az Ureg-ben):

CD8B—CDE5.

Egy karakter beadása az INPUT PUFFER-be INSERT-módban:

CDE6—CE0F.

Egy karakter beadása az INPUT PUFFER-be normál módban:

CE10—CE37.

Egy karakter törlése az INPUT PUFFER-ből:

CE38—CE9E.

A rendszer által lefoglalt terület kezdőcímeinek meghatározására szolgáló szubrutin:

CE9F—CFC A.

A rendszercímeik inicializálása:

CFCB—D02A.

Az INPUT PUFFER törlése:

D02B—D03D.

A rendszerüzemek mutatóinak meghatározására szolgáló szubrutin:

D03E—D064.

BASIC veremműveletek szubrutinja:

D065—D078.

Az Ureg tartalmának betöltése a BASIC verembe:

D071—D07F.

Változók törlése:

D080—D0D1.

Az ún. AR—X és AR—Y tartalma közt fennálló egyenlőség vizsgálata:

D0D2—D0F8.

Két karaktersorozat egyenlőség vizsgálata:

D0F9—D2E5.

Egy adott sorszámú sor kikeresése:

D2E6—D3D4.

Decimális/hexadecimális konverzió:

D3D5—D406.

Kódolt változónevek meghatározása:

D407—D45C.

Az Yreg-t mutatónak használva a változót keres szubrutin:

D45D—D460.

Az Ureg-ben tárolt változók kezdőcímeinek meghatározása:

D461—D5F8.

Egy 16 bites paraméter Ureg-be töltése:

D5F9—D6AC.

Egy adat:

D6AD—D6BF.

A változó átvitele az AR—X-ből vagy az INPUT PUFFER-ből:

D6C0—D6DE.

Az AR—X feltöltése egy változóval:

D6DF—D924.

Két karaktersorozat összekapcsolása:

D925—D992.

PEEK/PEEK #:

D993—D9A9.

INKEY S:

D9AA—D9B0.

CHR\$:

D9B1—D9CE.

STRS:

D9CF—D9D6.

VAL:

D9D7—D9DA.

A ROM további részeinek ismertetését folytatjuk.

DR. SIMONYI ENDRE

1. táblázat

Megnevezés	Címterület
SORGN	B153—B159
ROTATE	B15A—B169
COLOR	B16A—B17F
CSIZE	B180—B198
GLCURSOR	B191—B1B3
LF	B1B4—B200
LCURSOR/ TAB	B201—B221
LINE	B222—B223
RLINE	B224—B2EB
LPRINT	B2EC—B753
LLIST	B754—B7FF

2. táblázat

Megnevezés	Címterület
CSAVE	B8A6—B8F8
CLOAD	B8F9—B993
MERGE	B994—BB55
CHAIN	BB6A—BBB8
UP	BBC0—BBD5

3. táblázat

Megnevezés	Címterület
LET	C458—C4B5
STOP	C4B6—C50C
INPUT	C8FA—C967
LOCK	C968—C969
UNLOCK	C96A—C96D
LIST	C96E—C987
DIM	C988—C9E3
END	C50D—C514
GOTO	C515—C5B3
IF	C5B4—C5DF
ON	C5E0—C64D
GOSUB	C64E—C675
REM	C676—C67B
USING	C67C—C683
DATA/ ARUN/ AREAD	C684—C68B
TRON	C68C—C692
TROFF	C693—C696
DEGREE	C697—C6A3
RADIAN	C6A4—C6A7
GRAD	C6A8—C6AB
RETURN	C6AC—C703
NEXT	C704—C710
FOR	C711—C776
POKE	C777
POKE #	C778—C7A1
RESTORE	C7A2—C7B7
READ	C7B8—C809
NEW	C80A—C85E
CLEAR	C85F—C862
CALL	C863—C8B3
RUN	C8B4—C8C6
CONT	C8C7—C8F9

16 k-ról 17 k-ra

A ZX81 bővítése

A ZX81 alapgépnek 1 k vagy 2 k belső memóriája van. A belső memória a 16 k-s bővítő felhelyezésekor lekapcsolódik, mert a RAM C.S. vonal H szintre (+5 V) kerül. Egyszerű átalakítással elérhető, hogy ez a belső memória használható legyen a 16 k-s bővítő rákapszolása után is úgy, hogy a gép akkor is működőképes maradjon, ha a bővítő nincs rákapszolva.

Az átalakítás egy címdekódoló és néhány kapu (két IC) beépítéséből áll. A címdekódoló (SN 74LS139) azért szükséges, hogy ha a 16 k-s bővítőt rákapszoljuk, akkor a belső memóriaterület más címet kapjon. A kapuk (SN 74LS00) feladata, hogy a belső RAM az „eredeti” formában akkor is működjön, ha a bővítés nincs rákapszolva.

A ZX81 belső címdekódolása miatt a 8 k-nyi ROM 16 k címterületre foglal el, mert olyan a dekódoló rendszere, hogy a 8 k kétszer jelenik meg. A második 8 k címterület (árnyék ROM) a BASIC nem használja. Így ha ide cimezzük a belső memóriát, az a RESET

(RESET vonal L-re kapcsolása) hatására nem törlődik.

A bemutatott áramkör megoldja a ROM és a belső RAM címdekódolását. A rajzon megtalálható a gép hátulján levő csatlakozósáv (EDGE CONNECTOR) bekötése is. Segítségével elég könnyen megke- reszhetők az átalakításhoz szükséges vonalak. Az x-szel jelölt helyeken a fóliát át kell vágni.

Az áramkör működése

- ROM C.S. előállítása az első 8 k területen (θ —1FFF h).
- RAM C.S. előállítása a belső memória számára.

A belső RAM bővítés nélkül két címterületben jelentkezik: egyszer az eredeti helyen és egyszer az új helyen a 2000 h—2FFF h tartományban abban az esetben, ha a 3. kapu bemenet \bar{Q} -re van kötve. Ha \bar{Q} -ra van kötve — szaggatott vonal —, akkor a 3000 h—3FFF h tartományban jelentkezik. Ez mind a két esetben 4 k tartomány, így a nem teljes

címdekódolás miatt a belső memória 1 k esetén négyezer és 2 k esetén, tehát ha 2 k van a gépben, kétszer fordul elő egymás után a címterület.

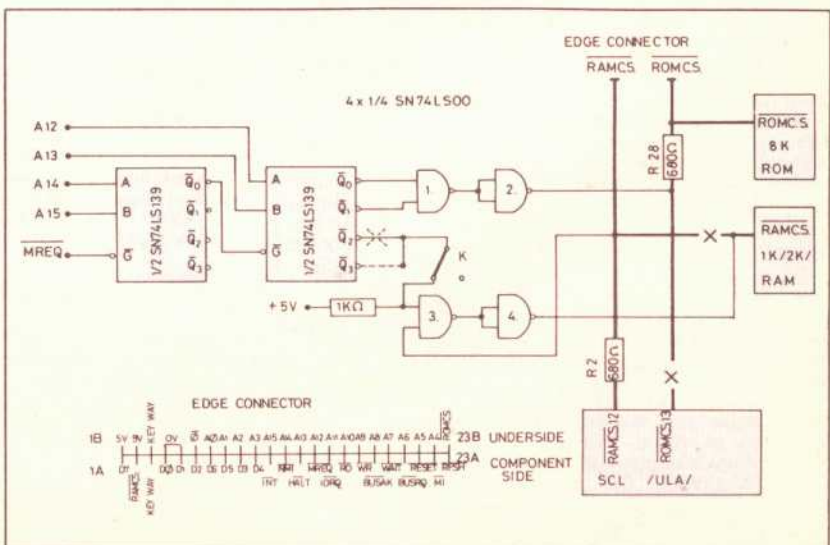
Bővítés esetén, mivel a bővítő a csatlakozósáv RAM C.S. vonalát H-ra kapcsolja, a belső RAM már csak az új helyen jelenik meg, az eredeti helyen nem, így nem zavarja a bővítő működését.

Az átépítéshez lehetőleg gyakorlott amatőrök fogjanak hozzá, mert egy hibás forrasztás vagy fóliátvágás tönkretetheti a gépet.

Nagyobb, 32 k, 64 k memóriabővítők használhatják azt a címterületet, ahová a belső memóriát cimeztük, ezért ezekkel a memóriabővítővel az átalakított gép csak úgy használható, ha a k kapcsolót megszakítjuk!

Az áramkör más jelű, de hasonló funkciókkal rendelkező IC-kből is felépíthető. A helyes lábbekötést a típusnak megfelelően katalógusból kell kikeresni.

KOVÁCS SZILVESZTER



PRIMO

Botkormány

A botkormánykezelő rutin (1. lista) a 16443-as tükörregiszteren keresztül lépteti a botkormány számláló áramkörét (lásd ábránkat). Ugyanezen

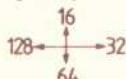
a tükörregiszteren keresztül olvassa be a dekódoló áramkör helyzetét, és az eredményt az 50000-es decimális címre helyezi. Ha két botkormányt használunk, akkor a második lekérdezésének eredménye az 50001-es címen található.

A rutin alkalmas arra is, hogy két nyomógomb egyidejű megnyomása esetén további irányításra alkalmas értékeket kapjunk.

A Rajz botkormányával neví programban (2. lista) a DATA-k tartalmazzák a botkormánykezelő rutint. A 200-as sorban történik a rutin hívása. A lekérdezés eredményét a 210-es sorban kapjuk meg. Ettől függően vezérelhetjük négy irányba a rajzoló képpontot. A program a lövőgombot nem használja, de annak megnyomása esetén k=8 értéket kapunk. Ezzel további vezérlési lehetőségekhez jutunk.

GERENDÁS SÁNDOR

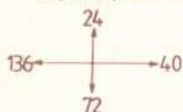
1. eset: lövőgomb lenyomva



2. eset: 144, 48, 192, 96



3. eset: lövőgomb lenyomásával együtt



4. eset: lövőgomb lenyomással együtt



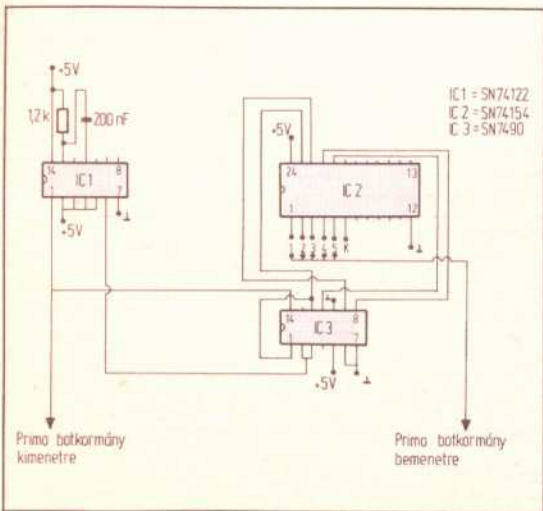
```

1
2
3 9C40 FD213B40
4 9C44 0E08
5 9C46 2150C3
6 9C49 3600
7 9C4B FD7E00
8 9C4E CBBF
9 9C50 CBF7
10 9C52 D300
11 9C54 FD7700
12 9C57 FD4600
13 9C5A CBB0
14 9C5C ED41
15 9C5E FD7000
16 9C61 DB64
17 9C63 CB47
18 9C65 37
19 9C66 2801
20 9C68 3F
21 9C69 CB16
22 9C6B 23
23 9C6C CB57
24 9C6E 37
25 9C6F 2801
26 9C71 3F
27 9C72 CB16
28 9C74 2B
29 9C75 FD4600
30 9C78 CBF0
31 9C7A ED41
32 9C7C FD7000
33 9C7F 0D
34 9C80 20D5
35 9C82 FD4600
36 9C85 CBF8
37 9C87 FD7000
38 9C8A ED41
39 9C8C C9
40

ORG 40000T
LOAD 40000T
LD IY.16443T
LD C.B
LD HL.50000T
LD (HL).0
LD A.<IY+0>
RES 7.A
SET 6.A
OUT (0).A
LD (IY+0).A
LEP: LD B.<IY+0>
RES 6.B
OUT (C).B
LD (IY+0).B
IN A.<100T>
BIT 0.A
SCF
JR Z.EGY
CCF
EGY: RL <HL>
INC HL
BIT 2.A
SCF
JR Z.KET
KET: RL <HL>
DEC HL
LD B.<IY+0>
SET 6.B
OUT (C).B
LD (IY+0).B
DEC C
JR NZ.LEP
LD B.<IY+0>
SET 7.B
LD (IY+0).B
OUT (C).B
RET
END
    
```

1. lista

2. lista

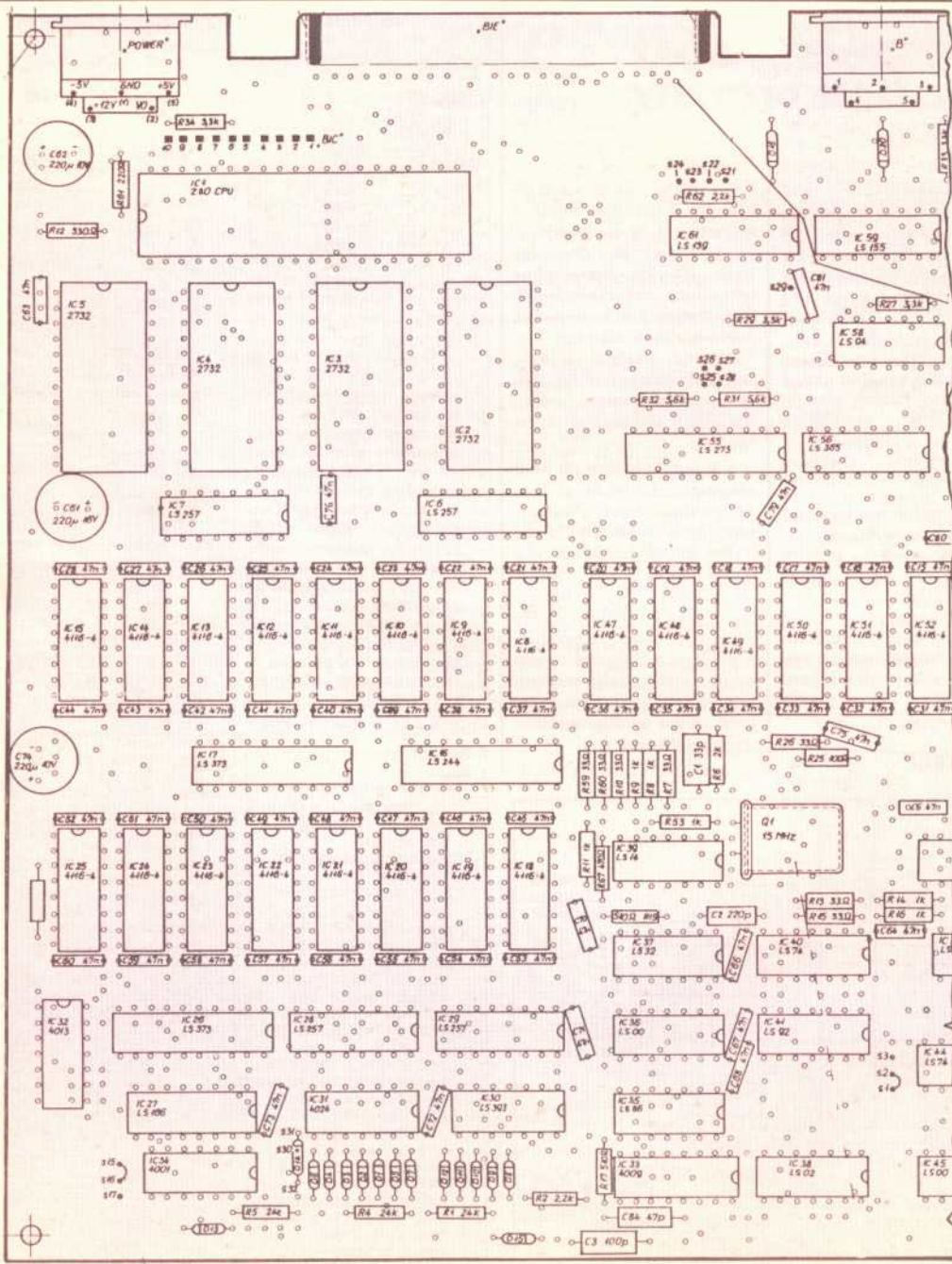


```

10 REM Rajz botkormannyal
20 FOR I=0TO74
30 READR
40 POKE30000+I,R
50 NEXT I
100 DATA253.33.59.64.14.8.33.80.195.54.0
104 DATA253.126.0.203.191.203.247.211.0
110 DATA253.119.0.253.70.0.203.176.237
115 DATA65.253.112.0.219.100.203.71.55
117 DATA40.1.63.203.22.35.203
120 DATA87.55.40.1.63.203.22.43.253.70.0
125 DATA203.240.237.65.253.112.0.13.32
130 DATA213.253.70.0.203.248.253.112.0
135 DATA237.65.201
150 CLS
200 C=CALL (30000)
210 K=PEEK (50000+65536)
300 IFK=16.A=A+1
310 IF K=64.A=A-1
320 IF K=128.B=B-1
330 IF K=32.B=B+1
340 IFA>191.A=191
342 IFA<0.A=0
344 IF B>255.B=255
346 IFB<0.B=0
350 SET(B.A)
400 GOTQ200
    
```

PRIMO

1. ábra



Príma ötlet

Minden számítógép-felhasználó életében elkövetkezik az a nap, amikor kinövi készülékét. Ekkor vagy komolyabb gépet vásárol, vagy a meglevőt bővíti.

Mi ez utóbbi mellett döntötünk, amikor egy Primo A-32-es (16 k ROM, 16 k RAM) számítógépet kívántunk kibővíteni A-64-esre (16 k ROM, 48 k RAM). Először a javítással és átalakítással megbízott, Fehérhajó utcai Gelka szerviznél érdeklődtünk, ahol megtudtuk, hogy a bővítést haterzet (1) forintért végeznék el (ebből kb. 2000 forint az alkatrészár), ha lenne hozzá alkatrészük.

Maradt tehát a régi mondás: „Magad uram, ha szolgád nincs!” Íme a recept. Végy 16 db MCM 1618 vagy 4116 típusú integrált áramkört, 32 db 47nF-os tárcsakondenzátort és 1 db kétállású, 4 áramkörös izosztát kapcsolót. Az utóbbira azért van szükség, hogy a számítógépet a kapcsoló állásától függően 16 k vagy 48 k memóriával lehessen használni. Ha csak 48 k memóriával kívánjuk használni, felesleges a kapcsoló.

Az 1. ábrán levő beültetési-részleten feltüntetett (IC 8-15, IC 18-25) helyekre beforrasztjuk a 16 db IC-t és al-

fölé 1-1 db 47nF-os kondenzátort (C 21-28, C 37-44, C 45-52, C 53-60). Az A-32-es típus alapállapotában az S 21-S 24, S 25-S 29, S 28-S 29 pontok vannak összekötve. Bővítéskor ezeket oldjuk, és helyettük az S 23-S 24, S 25-S 26, S 27-S 28 pontokat zárjuk rövidre.

Az izosztát kapcsoló egyik lehetséges bekötését a 2. ábra mutatja. A kapcsolót olyan helyre kell helyezni, hogy véletlen átkapcsolás ne következhesen be. A bekapcsolt számítógépet ugyanis tilos átkapcsolni! A véleményünk szerinti legvédehetőbb helyet a számítógép jobb oldalán a billentyűzet íve alatt találtuk meg (3. ábra).

Ha eddig eljutottunk, bedobozolhatjuk és bekapcsolhatjuk a gépet. A teljes képernyőt betérítő ponthalmaz jelzi, hogy jó munkát végeztünk.

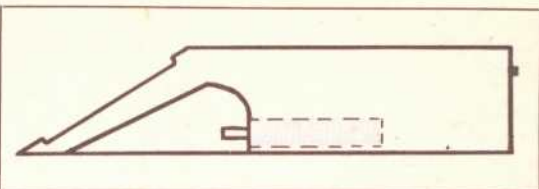
Az A-32-es számítógép képernyője ugyanis a 26 624-32 767 címeken található. Memóriabővítéskor ez feljebb csúszik, pontosan 32 k-val. Az új képernyőhely meghatározását a gyártómu EP-ROM IC cseréjével végzi el. Mivel nekünk nem volt ilyen, más címeket tartalmazó IC-nk, szoftver úton kell megtanítanunk a számítógépet az új képernyő helyére.

Ezt kétféle módon tudjuk megtenni. Az egyik megoldás, hogy „vakon” a billentyűzet hangvisszajelzését figyelve begépeljük a következő közvetlen utasításokat:

```
POKE16458,232:
POKE16562,231:
```

CLEAR50:CLS és nyomunk egy RETURN-t. Hatására a képernyő megtisztul, és a bal felső sarokban megjelenik az „OK” felirat és a kurzor. Az egyik POKE a képernyő elejét, a másik a változómező elejét írja át.

A másik megoldás, hogy a még 16 k-s gépbe beírjuk az



3. ábra

alábbi programot, és futtatás nélkül magnóra kimentjük:

```
10 REM XXXXXXXXXXXXX
20 REM X X
30 REM X 48 K X
40 REM X X
50 REM XXXXXXXXXXXXX

60 POKE16458,232:
POKE16562,231:
CLEAR50:CLS

70 PRINT $0,0,CHR$(2)“PRIMO”CHR$(18)“ BASIC SYSTEM '85.11.”TAB(38)“48K”

80 DELETE10-80
```

Ebben az esetben az átkapcsolt gépbe már csak az I LOAD és a RUN parancsot kell „vakon” beírni, és elindítani a magnót. A sorszámmal ellátott

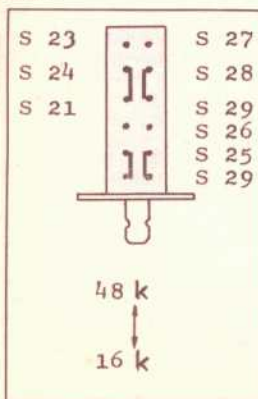
LOAD parancs futtatásakor a számítógép automatikusan indítja a betöltött programot.

A RESET kapcsoló megnyomásakor a kurzor visszazugrik a régi helyére. Ekkor újra be kell írni a POKE16458,232-t, de már csak ezt, hogy a kurzor visszatérjen.

Tudjuk, hogy ez a tárbővítés elég kezdetleges, de több mint fél éve hibátlanul működik, és megnyugtat bennünket az a tudat, hogy pár ezer forint többletkiadást meg tudunk takarítani. Amennyiben a leírtak megvalósításával bármilyen további kérdés, probléma merülne fel, szívesen állunk bárki rendelkezésére.

DEME ATTILA
HADNAGY ÁKOS

2. ábra



Ki ad magyarázatot?

A Primo számomra érthetetlen rendellenességeket produkál az adatok magnetofonszalagról történő beolvasása után.

1. Bizonyos programoknál a beolvasást követően READ-DATA utasításpárt használva, szöveges változóknál olyan jelenség lép fel, mintha minden READ utasítás után RESTORE utasítás lenne. Mivel a programban a hibát nem találom meg, ezekben az esetekben a következő „kerülő” megoldásokat használom:

- READ utasítást csak a magnetofon használata előtt alkalmazok.
- Az állandókat LET utasításban adom meg.
- Minden adatot szalagról olvasok be.

2. Az adatok beolvasása közben FD ERROR jelzést kapok esetenként olyan programoknál is, amelyek már több alkalommal hibátlanul működtek.

Ebben az esetben csak egy kerülő megoldást találtam. Az adatok szalagra történő rögzítésénél a folyamatot az adatok képernyőre íratásával lassítom, és a CLOSE utasítás elé lassító ciklust írok.

SOMOGYI GYÖRGY

Azt mondják, hogy nyáron mindenki leáll, az emberek inkább a nyaralással törődnek, mint a munkával. Olvasóink levelei ezt az elméletet erősen cáfolják; azt hiszem ez a szokás az informatikával foglalkozók körében nem érvényes.

Fáy András, Miskolc,

Széchenyi u. 6. 3530

Azt szeretném megkérdezni, hogy Miskolc-on miért nincs valamilyen számítástechnikai szakbolt, vagy valami ehhez hasonló. Miért minden Budapesten van. Az ország második városában rendkívül gyenge a számítástechnikai ellátottság. Szakkönyvekből tényleg az ellátás, most nyílt egy bolt a Tanácsház téren a Novotrade gondozásában. Számítógép-ellátás borzasztóan gyenge, és az árak magasak. Az OFOTÉRT-ban egy C16 kb. 21 000 Ft, C64 a Bizományiban 30 000 Ft, VC—20 kb. 20 000 Ft, C116 24 000 Ft. Kazettás magnó COMMODORE-hoz 6000—10 000 Ft. Igen gyenge „fejlődést” lehet tapasztalni, annyit mond össze, hogy a CENTRUM-ban lehet kapni TVC-t 12 800 Ft-ért és ATARI800-at 20 000 Ft-ért. ZX81 kapható 5500 Ft-ért, ez sem túl olcsó.

A CENTRUM-ról olyan hírek terjedtek el, hogy a lakosság részére fog árusítani számítógépet (lásd pl. *µM* 1986/5. számát, 11. oldal).

Az egyik decemberi HÉT-ben láttam, azt hiszem, hogy az egyik illetékes azt mondta, szervezik C16 behozatálát kb. 50 dolláros áron.

Az újsággal alapszabán véve meg vagyok elégedve, a hirdetés kicsit sok.

Levelet abban a reményben közlöm, hogy a miskolci számítógép-kereskedelemért felelős illetékesek is olvasni fogják.

Valkó Csaba, Miskolc,

Vörösmarty u. 66. VII/3. 3530

Nagyon örülök, hogy a lap az idejévtől havi formában jelenik meg. Ahogy olvastam az Olvasó írja rovatot, láttam, hogy az évi hat szám megjelenésével kapcsolatban nem csak nekem volt problémám. De ezek a problémák, mint ahogy Önök előre látták, egy csapásra megszűntek a havi megjelenéssel. De nemcsak ezért tartozom Önöknek köszönettel, hanem azért is, mert így minden hónapban hozzájutok kedvenc újságomhoz.

Látva a Magazinban a sok vitát a Magazin tartalmi és formai oldaláról tekintve, nagyon szorított Önöknek, hogy olyan lapot tudjanak készíteni, amely valamennyi olvasó igényét ki tudja elégíteni. Nekem például nagyon tetszik a lap mostani formája, nem tudom, hogy egyeseknek miért vannak

maximalista véleményeik a lappal kapcsolatban. Hiszen a lap úgy jó, hogy minél több dolog megtalálható legyen benne, ez pedig ilyen terjedelem (45-50 oldal) mellett lehetetlen feladat.

Mindenesetre várom a fejleményeket.

Min is, az elismerést valamennyiünk nevében köszönöm.

Liptay László, Szabadkigyós,

Nefelejcs u. 13. 5712

Van egy Commodore VC—20-as számítógépem. Sajnos a memóriája csak 3,5 kb-át, és bővíti nem bírja. Ezért a segítséget szeretném kérni. Ha lehetséges legyen szíves és küldjön nekem a géphez memóriabővítési rajzot, ugyanis szeretnék építeni egyet. A rajzért küldök egy kazettát.

A levelet átadtam az illetékes rovatserkesztőknek. Kérését azért közlöm, hogy másoktól is kaphasson ötletet vagy segítséget.

Tóth István Gábor, Békéscsaba,

Tanácsköztársaság u. 52. I/5.

Sajnálattal meg kell állapítanom, hogy a korábban írt észrevételeket, melyeket a *µMagazin* '85/6. számában megjelent C64 UNIN rutin programmal kapcsolatban írtam, az 1986/4. szám HELYREIGAZÍTÁS-a után ki kell egészíteni az alábbiakkal:

1. A közölt formában a stringekben szereplő „betűköz” karakterek nem egyértelműen értelmezhetők („...vagy”?). Ugyancsak hibás működésre vezet ha a melléklet szerinti 265-ös sorban a T\$ cursor vezérlés helyett üres karakterre végez vizsgálatot.

2. Bizonyára nyomdahiba és nem szándékos elírás egy új változó a 140-es sorban a BI az egyébként szükséges B1 helyett.

Ugyanebben a sorban az AND B> helyett szedett AND D> már az igazán kezdők számára is annyira szembetűnő, hogy biztosan nem fogják szolgai módon „bepotyógtetni”.

Mindezt figyelembe véve az összesen nyolc sor „HELYREIGAZÍTÁS” nyolc újabb hibát tartalmaz.

Úgy látszik ez a „szuper program” a nyomda ördöge miatt csak nem akar hibátlan lenni.

Egyébként biztosan igen jól hasznosítható program, különösen ha az alapfeladat néhány száz sor, és 30-40 különböző hosszúságú és fajtájú adatot kell a feldolgozás során bevinni.

A helyreigazítás helyreigazítása a következő:

```
105 T1 = PEEK(51):T2 = PEEK(52)
: T6 = T1 + 256 * T2 - 41: A$ = "" +
: AS: T4 = INT(T6 / 256): T3 = T6 - T4 *
: 256
140 B = LEN(A$): IF B1 > 15 AND B < 6
AND B
> 0 THEN FOR B = B + 1 TO 6:
```

```
J$ = "" + A$: GOSUB 540: NEXT: GO
TO 160
210 Z = (55296 + L * 40 + C): FOR I = Z
TO Z + P
: POKE I, CO: NEXT: Y = 1024 +
L * 40 + C: X = Y + P: IF B7 < > 70RB
= 0 THEN Z20
265 POKES1, S3: S3 = 0: S4 = 80:
IF B8 = 7 AND T$ = "□" AN
DA = 1 THEN GOSUB 535:
J$ = CHR$(14): GOSUB 540: A = 0: GO
TO 195
340 A = A + 1: GOSUB 535:
J$ = LEFT$(A$, A -
2) + T$ + RIGHT$(A$,
(AV - A + ABS(AV - A)) / 2):
GOSUB 540: IFA > AV THEN AV = A
495 IF B0 < > 7 AND (B1 = 0 AND
LEFT$(A$, 1) = "")
OR (B1 < > 0 AND RIGHT$(
(A$, 1) = "")) THEN A = 1: AV
= B + 1: GOTO 240
500 IF B6 = 0 THEN FOR I = X TO Y STEP -
1:
POKE I, ASC(MID$(A$, 1 - X + B, 1))
AND 63 OR 128: NEXT: GOTO 195
540 L$ = L$ + "" + POKE 820, PEEK(72):
POKE 821, PEEK(72):
POKE(PEEK(820) + PEEK
(821) * 256), 41 - LEN(J$)
```

A grafikus jelek jelentései az alábbi sorokban:

```
265 kurzormozgatás fölfelé
355 kurzor jobbra lép
360 kurzor jobbra lép
395 INSERT
415 kurzor balra lép
420 kurzor balra lép
425 HOME
430 CLEAR
```

Sajnos erre már semmi mentességünk nincs.

Szilvay Gyula, Budapest

XVIII., Nagyenyed u. 21. 1182

Kérem, hogy lapjokban szorítsanak helye „piaci” tudósításoknak is, érdekesebb, szívesebb híreknek külföldi gépjárműsárgóról (ajánlom megtekintésre a Haditechnikai Szemle egy számát mintának), mert különben kénytelen leszek visszamondani előfizetésemet. Unalmasak a hexadecimális listákkal kitöltött 10 oldalak! Akinek ez kell, vegye a Magyar Elektronikát, vagy Önök adják ki azokat külön mellékletben, de én még az első szám alapján szavazatom bizalmat a lapnak, és ha nem változik a tartalma (mint pl. volt a bécsi tudósítás. — Mint írták, megjelenésekor már elavult — de ki jár havonta Bécsbe?) én bizony lemondok az előfizetést

Már nagyon sokszor leírtam, hogy a µM olvasóinak az érdeklődése igen széles körű. Ahogyan Ön elvéri rajtnak a port a hexadecimális listák miatt, mások a túl sok szöveg miatt panaszkodnak. Annak persze örülök, hogy az ausztriai piacról szóló cikkem tetszett, de az is igaz, hogy sokaknak, főleg

megszállott programozóknak nem, mert elvetem a helyet néhány programotlet elől. Új kiadónkkal megkíséreljük a lap eddigi 3-4 hónapos áfutását idejét lecsökkenteni, ez a feltétele annak ugyanis, hogy érvényes piaci információkat tudjunk közölni.

Nagy László, Ináncs

Széchenyi út 13. 3851

Kb. 1 éve olvasom lapjukat és ezalatt egyre kevesebb a ZX—Spectrumra írt programok száma. Jó volna ha ebből is annyi volna, amennyit a C64-esre közölnek.

Második kérdésem, hogy hol lehet ZX—Spectrum-ot kapni? És miben különbözik a 48 kb-átos Spectrumtól? És végül, hol lehetne megvásárolni az Ada-Winter Péter—Ada-Winter Dávid: A ZX—Spectrum című könyvet? Ha volna, részemre utánvétellel elküldeni szíveskedjenek.

Valóban kevesebb Sinclair ZX, illetve Spectrumra írt programot készítenek és küldenek szerzőink, így mi is kevesebbet közlünk. Sinclair gépeket legjobb tudomásom szerint magánimportból lehet vásárolni a bizományosi boltokból (Bizományi Áruház, Foto-Optik, Ramovill, Skála-Metro stb.) Az Ada-Winter könyv még nem kapható. Amint megjelenik, megrendelhető a Műszaki Könyvruházban, Bp. V., Liszt F. tér 9.

Dr. Jankó Domonkos,

Közlekedéstudományi Intézet

Örömmel olvastam a Mikroszámítógép Magazin f. évi 5. számában Laczka Miklós „Mikroszámítógépek a tavaszi fesztiválon” c. cikkében KRESZ-elsősegély és munkavédelmi vetélkedő programunk kedvező értékelését, mivel a program tartalmi, szakmai részét, tehát a KRESZ-kérdéseket én készítettem és a lemezek forgalmazását is én irányítottam.

Szakmai lap lévén, szeretném azonban, ha helyesbítést tennének közzé, mivel a software-t és az arra épülő programokat Dobos Imre és software-teamje (Bodnár Tibor, Dobos Katalin és Kovács József) fejlesztette ki, ill. készítette.

A „munkahelyi vetélkedő” c. lemezt egyébként továbbra is a Közlekedéstudományi Intézet forgalmazza. Szívesen közöljük a helyesbítést.

Hargitai Tihamér, Budapest,

Nappfény u. 16. 1039

Commodore 16-os számítógépet kapott a fiam ez év elején. A szükséges könyvek, folyóiratok és a témával foglalkozó hetilapok segítségével autodidakta módon eljutott egy szintre, oly annyira, hogy egy egyszerű játékprogramot is meg tud már írni. Nos

ezek után körülnéztem, hogy lehetne továbbfejleszteni azoknak a gyerekeknek, akik már nem kezdők?

Természetesen a saját erőből való továbbfejlődés lehetősége továbbra is megvan, de hogy egy példával éljek, egy tornásznak készülő gyermek első időben, amíg izomfejlesztés a cél, megítélésem szerint nem igényel különösebb irányítást, de amikor már megfelelő izmokkal és képességgel rendelkezik, nem hiszem hogy ugyanolyan eredményt ér el az a gyerek, akivel célirányosan szakember foglalkozik, mint akivel nem.

A számítástechnikára is vonatkozik, hogy ahová saját erőből két év alatt jut el egy gyermek, oda jó irányítással fél év alatt eljuthat.

Itt kéne találkozni az általános hangoztatott elveknek és a gyakorlatnak, de ezt az utat én még keresem.

Azt hiszem a magyar oktatásügy is. A fejlődés jó iskolájának tartom a μ klubokat (már amelyiket), és ha megindul az informatikai távoktatás, akkor az ott elhangzó előadások mindenki számára biztosítják a továbbtanulási lehetőségét.

Bordás Bence, Tiszaföldvár,

Rákóczi u. 2. 5430

Bordás Bencének hívnak, gimnáziumban tanulok Tiszaföldvár Cityben. Mivel eddigi próbálkozásaim sikertelenül jártak, és a lapot 1984 óta járattam (őszintén szólva akkor többet értemtem volna egy kínai regényből), úgy tapasztaltam, hogy sokak problémáját sikerült megoldaniuk, én is elszántam magam, hogy leporolom az írógépet, és betöltöm magamba a gépelő programot. Nos, rún.

Mivel a VC 20 ún. számítógépre nem sok játékprogramot ismerek, nem is hiszem, hogy nagyon sok lenne, elhatároztam, hogy minden BASIC tudásomat és rutinomat latba vetve, csinálók már egy „normális” játékot.

Nézzük. Végy egy jó, 3,5 kbyte-os ötletet! Van. Végy egy marék sprite-et! Hoppá, ezen a gépen egy megveszekedett fia sprite-et nem találtam, nem hogy egy marék. Nem baj, van definiálható karakter. Kérelj jó hátteret! Ready. Bejelentkezés, hanghatások, pontszámtáblázat. Nem téma. Lásuk, mi van még? Scroll négy irányba. Itt esik a fene... Lefelé, — kész van. (Kímültöttem másfél óras, komplikált, de sikeres műtét során egy másik játékból.) Felfelé, — kész. (Print utolsó pozícióba való állítása.) Jobbra és balra. Ez az én gondom. Megkértem az iskolán egyetlen (krónikus időhiányban szenvedő) számítástechnikához értő tanárát, segítsen. Sajnos csak elméleti tanácsokat tudott adni (amiket már tudtam), mivel a híres, nevezetes HT—1080Z számítógéppben szunnyadó Z80 processzor lelkének bökdöséséhez van gyakorlata (pokeoláshoz). Sebaj, van egy barátom, aki BA-

SIC-ből már 3 danos, de gépi kódból még csak sárga öves. Ő is nekilátott, hiszen ő tudja, hol kell bökdösi a 6531-est, hogy elérjünk nála valamit. Végén öneki sem sikerült. Nagyon örülnék, ha küldenének nekem egy ilyen gépi kódú rutint. Előre nem köszönöm meg, mert ezt a gesztust nem tartom az értelmes ember sajátjának, másrészt ez „thank without program” hibajelzéshez vezethet. Szóval nagyon örülnék, ha elküldenék nekem a rutint, és ha igényt tartanak rá, szívesen elküldeném a kész programot. Más. Mióta jár a lap, nagyon sok érdekes, felhasználható cikket, programot olvastam benne, nagyon meg vagyok vele elégedve. Tény, hogy volt egy-két gyengébb szám is, de általában minden számban található cikk vagy érthető program mindenki számára. Mióta olvasom a lapot, egyre többet értek meg belőle, és most már ott tartok, hogy most már nem lapozok el egyetlen oldalt sem azzal, hogy „ezt előbb fordítsák le magyarra”. Nem értek egyet azokkal, akik kritizálják a lapban található hirdetések mennyiségét. Ez hozzátartozik a magazin-szerű újsághoz. Egyébként biztos elfogynának az ezt panasoló levek, ha az újság 60 forintba kerülne, (akkor viszont elfogynának azoknak a levelei is, akik nem tudják megszerezni a lapot).

Örömmel vettem, hogy az újság most már havonta jelenik meg. Drukkolok, hogy a sűrűbb megjelenés dacára a magazin ugyanolyan tartalmas tudjon maradni mint ahogy azt eddig szerencsére megszoktam. End. Ready.

Ehhez a levélhez nem szabad semmit sem hozzáfűzni, így elköszön a rovat vezetője

KOVÁCS GYŐZŐ

Helyreigazítás

A μ M 1986. augusztusi számában közölt Commodore 64 segéd táblázatában sajnálatos hiba miatt az alábbi javítások szükségesek:

- A D7 mezőben DCP zp helyett helyesen DCP zp,x áll.
- A cikkben külön felsorolt — új — utasítások **kék színnel** jelzettek.
- A vízszintes vonallal jelölt utasításokat a processzor nem tudja értelmezni.

A téves színjelölések sajnos nehezítik, de nem gátolják a táblázat használatát. A hibákért olvasóink szíves elnézését kéri a szerkesztőség.

Kedves leendő Szerzőink!

Kérjük, hogy cikkeiket az alábbi kívánalmakkal megfelelően küldjék be:

- 2 példányban gépelve,
 - a programlisták jól olvasható (nem halvány, elmosódott), normál szélességű karakterekkel, legfeljebb 40 kar/sor formátumban legyenek kinyomtatva,
 - az esetleges ábrák feliratait egyértelműleg legyenek.
- A szerzőtől az alábbi adatokat kérjük:
- név,
 - személyi szám,
 - lakás cím,
 - munkahelyének neve, címe.

A szerkesztőség

SZÁMALK az ORGTECHNIK

F/60-as standján
minden kedves látogatóját
szeretettel várja!

SZÁMALK – ORGTECHNIK

Termelésirányítási és
vállalatirányítási
rendszerek
Mikrogepes rendszerek,
alkalmazások
Számítástechnikai
alkatrészek, segédanyagok
Oktatás – Szakkönyv



SZÁMALK ORGTECHNIK

Csak két – párosításban igen merész – példa a



SZOFTVERKERESKEDELMI ÉS FEJLESZTÉSI BETÉTI TÁRSULÁS

által forgalmazott IBM PC és azzal kompatibilis számítógépeken alkalmazható szoftverekre.

3 dimenziós építész tervező rendszer:

A Nyugat-Európába már 1984 óta exportált 3 dimenziós speciális építészeti tervező program által biztosított lehetőségek közül csak néhány:

- Alaprajz interaktív tervezése képernyőn grafikus tablettel vagy „mouse”-zal
- Automatikus méretezés, terület- és térfogatszámítás
- Magasságadatok numerikus bevitelle
- Homlokzati, metszeti és perspektív nézetek automatikus szerkesztése
- Kiviteli terv szintű rajzok készítése plotterre tetszőleges méretarányban
- Konzignációs listák és költségkalkuláció automatikus nyomtatása

Ára: 400 000,- Ft

VARYNFO

interaktív adatnyilvántartó rendszer

Egy program – több tucat probléma megoldása:

- személyzeti és munkaügyi nyilvántartás
- raktárkészlet, rendelések, számlák nyilvántartása
- könyvtári alkalmazások
- kérdőívek, felmérések feldolgozása
- telefonkönyv, üzletfelek nyilvántartása
- vezetői információk

Használata számítástechnikai előismeret nem igényel, kezelése a legegyszerűbb kérdésekre adott legegyszerűbb válaszokkal történik.

Ára: 39 000,- Ft

Új címünkön várjuk érdeklődésüket:

1137 Budapest XIII., Kun Béla rakpart 8. (Újpesti rakpart) Telefon: 129-230

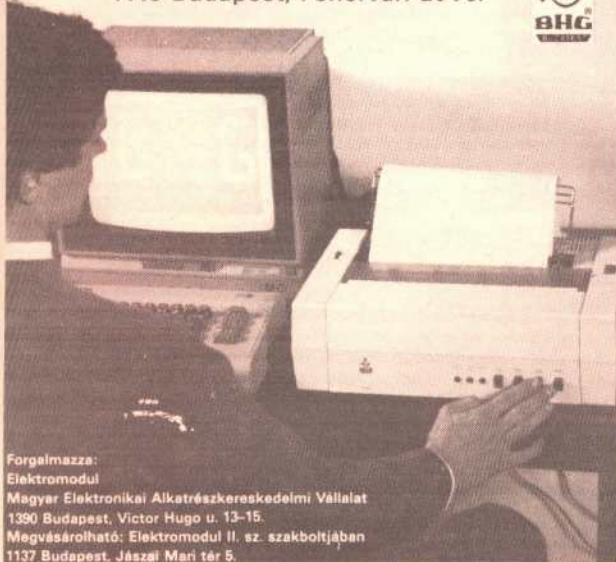
Nemcsak adat, szöveg,
hanem grafikák,
grafikonok
képi megjelenítésre is kiválóan alkalmas
a **PRT-80 GS**
grafikus mátrixnyomtató.

Sokoldalúan felhasználható, asztali kivitelű, első-
sorban személyi számítógépek, mikro- és minigé-
pek, intelligens terminálok, vezérlő- és mérőkészü-
lékek adatkiviteli perifériája.

Főbb alkalmazási jellemzői:

- kétirányú nyomtatás, logikai kereséssel,
- mikroprocesszoros vezérlés,
- grafikus karakterkészlet,
- karakterkiemelés,
- normál, dőlt és dupla széles karakterek,
- szövegtömörítési lehetőség,
- automatikus soremelés,
- változtatható papírszélesség, formátumhossz,
sorszámság és sorköz,
- traktoros vagy gumigörgős papírtovábbítás,
- 80/132 kar/s sebesség.

Gyártó: BHG Híradástechnikai Vállalat
1119 Budapest, Fehérvári út 70.



Forgalmazza:
Elektromodul
Magyar Elektronikai Alkatrészkereskedelmi Vállalat
1390 Budapest, Victor Hugo u. 13-15.
Megvásárolható: Elektromodul II. sz. szakboltjában
1137 Budapest, Jászai Mari tér 5.



**Fiatalok!
Figyelem!**

Indul az 1986. évi 8.,
szeptemberi számunkban
meghirdetett rejtvénypályázat!

Feladatok:

1. Milyen célt szolgál
az alábbi program?

```
10 DEFDBLT,S,X
20 INPUTX:X=X*0.0174532925
30 N=1:T1=X:S1=T1
40 T2=-T1*X[2/(N+1)]/(N+2)
50 S2=S1+T2
60 IFABS (S2-S1)
   <1E-12THENPRINTS2:END
70 T1=T2:S1=S2:N=N+2:GOTO40
```

2. Készíts programot,
amellyel a bérkifizetéskor
a szükséges címletek
meghatározhatók
(a „címletezés” a lehető
legkevesebb bankjegyet,
ill. érmét eredményezze)!

A program
változtatás nélkül futtatható legyen
a két iskolaszámítógépen,
a HT-1080Z-n és a C16-on.

Beküldési határidő:
1986. október 20.

**A megoldásokat írásban kérjük,
szerkesztőségünk címére:
Mikroszámítógép Magazin
Szerkesztősége,
Bp. Fő u. 68.
IV. em. 452. 1027**

**A borítékra írjátok rá:
DIGITÁL rejtvénypályázat.
Sok sikert kívánunk!**

A szerkesztőség

(Mottó: Mit ér, ha az ember egyik lábával a tudomány legmagasabb fokára hágott, ha közben a másik beheragadt a sárba. — Konfuciusz)

Kis barátunk ismertetését mindig úgy kezdik, hogy — felbomolják. Lám, ez is van benne, meg az is! Remélem, hogy az orvosudományi egyetemen a leendő orvost először megtanítják arra, hogy mi az ember. Majd a sokadik évfolymon boncolnak csak.

Elterjedt az a tévhit, hogy az „információs rendszer”, egyáltalán a rendszer, a számítógép függvénye. Mostanában napvilágot látott tankönyvünkben is a rendszert programok együttesének tekintik. A rendszertervezés = programkészítés. Ennél már Pistike is többet tud, mióta anyu receptjeit számítógépen tárolja.

Válasszuk végre szét a hobbit és a szakmát! — Hogy miért mások? Hiszen ott van a hegy! — mondják a hegymászők. Sok programozó is úgy véli, hogy azért kell programozni, mert ott a gép. Én máséknél vélekedem. Ezért — vizsgálva türelmükkel — e cikkben az alapokhoz megyek vissza. Ha unalmasnak is tűnik, ajánlom, hogy először ezt a hegyet mászzuk meg, hogy utána edzeten és könnyedén jussunk fel dombokra.

Az információról

Kora reggel mondta a rádió, hogy a múlt évben az x. y. térszében a tengeri átlagos hozama 60 mázsa volt hektáronként. Mi tagadás, ennek hallatára nem vágtam meg magam a borotvával. Már elfeledtem az egykor tanult gazdaságföldrajzból az átlagos hozamokat.

Ugyanebben a pillanatban valahol Józsi bácsi levelete a pipát szép száma agyaráiról, és ennyit mondott: No! Többet nem szólt, hanem várt. (Hiába). Józsi bácsi ugyanis nem nyilvánít véleményt addig, amíg nem tudja, hogy arrafelé milyennek a földek, mekkora volt a belvízkár, miként alkalmazták a „meliorizációt”. (Az utóbbi kifejezés többnyire a tragédiás irodalmi kifejezése a rádiós rovatokban, de erről a téhen nem tud.)

Az információ pontosan olyan, mint a sült csirke. A delikvens elkészült, és előre tu-

dom, hogy erre a véznácska jószágra nagyobbik lányom azt mondja, hogy zsíros. A papa majd kijelenti, hogy sótlan, miközben én dörögöm: Drágám, ezt picit élesztőd. A kicsi szerint a csirke égett, szerintem lehetne ropogósabb is. Ha kiváló hallaná az elhunyt ilyen méltatását, azt hihetné, hogy itt mindenki mást eszik.

Ez a kiváló nem számítástechnikai szakember. Nem tudja, hogy a csirke az csak „adat”, és megevésevel születik az „információ”. (Pontosabban már színével is illatával is, de e tényezők is, az ízhez hasonlóan, eltérő benyomásokat keltenek bennünk.) E ponton az olvasó kimehet harapni valamit a konyhába, ha van utána sőre. Folytathatom?

Szóval azt akarom mondani, hogy az információ legszemélyesebb tulajdonunk. Bennünk és csakis bennünk születik abban a pillanatban, amikor egy hír, egy közlés, egy új ismeret elér hozzánk, azt érzékeljük és felfogjuk. Illetve ilyenkor még nincs szó információról. Csak akkor, amikor a hírt értelmezzük. Azaz összevetjük a vele érkező egyéb hírekkel, vagy — ami technikailag ugyanaz —, a már bennünk létező ismeretekkel.

Mivel más a hátterem, mint Józsi bácsinak, a tévesz-híradó nekem mást mond, mint neki. A közlésből mindenkinél többet tud meg a szomszédos térsz vezetöje, aki ismeri a körülményeket. Ha pedig a nyilatkozó is hallgatja a rádiót, akkor legfeljebb saját kellemes baritonjában gyönyörködhet, mert információt nem kap. Hiszen ő adta.

Tévesek azok a vélemények, melyek szerint az adat forma, az információ pedig a forma köntösébe öltözött tartalom. Ne lenne az elfogyasztatlan sült csirkének tartalma? De mennyire van! Az adat a valóság valamilyen jelenségének tükörképe, míg az információ: értelmezett adat. Mivel mindenki másként értelmez, az információ valóban személyes jellegű. Én ne tudnám, aki őt főnöknek adta mindig ugyanazokat a (tartalmas) adatokat — mindig más következmény-nyel?

Barátunk, a számítógép

Beszéljünk a feladról!

Az adatfeldolgozásról

Régi, értetlen tankönyvekben olvastuk, hogy van adatfeldolgozási célú és műszaki-tudományos számítógép. Folyosón hallottam: „Ezek azt a nagy böhöm gépet csak adatfeldolgozásra használják.” Idé-

ahol apa közölte a főszakccsal, hogy három és fél csipet majoránával kéri a sült csirkét, amely kizárólag zsályán és nem halliszten nevelkedhetett. Volt az asztalon só, bors, paprika, és kicsit lehetett korigálni az ízeken, de nem sokat. Hiszen már a kiindulás-



Az adat és az információ lényege

zet vezetői beszámolóiból: „Eltársak, mi nem adatfeldolgozásra, hanem vezetői információk előállítására fogjuk használni a számítógépet”.

Osztán miből lesz a vezetői információ? Vagy tekinthetjük-e értelmes és a dolgokat a szó valódi értelmében „értelmezni” tudó bírótunk, és számítógépet? Tetszik vagy sem, a számítógépek semmi másra nem valók, mint adatfeldolgozásra. Ha majd eleget dolgozt kis barátunk, és közli velem a részben emésztett adatokat, akkor majd én szülök információt. Ezt a jogot fenntartom az emberiségnek!

Nem kívánok itt szörszálakat hasogatni. Hiszen jómagam is meglehetősen pongyolán alkalmazom — szinonimaként — az adat és az információ megjelöléseket. Időnként azonban gondolok a lényegre is. Mert ha ezt nem tenném, akkor reményem se lehetne arra, hogy valaha is jó számítástechnikai szakember válik belőlem. A most BASIC-et tanuló Pistinek ezt a két dolgot hamar meg kellene ismernie!

Pisti még nem járt mostanában olyan magyar étteremben,

ra sincs befolyása a fogyasztónak.

Aki érteni akar szakmánkhöz, annak tudnia kell, hogy vannak ún. alapadatok. Jé, ez a formás lény előttem hölgy. 99 forint a paradicsom ára, vagy annak legnagyobb közös többszöröse — kell? Az az autó piros. E mondatokban alapadatokkal közöltem.

Vannak különböző fokig feldolgozott adatok. Egyszerűbb adat az egy személyes átlagjövédelmet mutató, mint a vállalati átlagjövédelmet tükröző. Így van ez a csirke esetében is: bontott csirkét, bontott csirkét...

Az adat egy bizonyos fokon érik meg a fogyasztásra, válik képessé az információgenerálásra. Kedvenc, saját zargonomat kell ennek kapcsán bemutatnom.

A feldolgozási lánc

Van ilyen. Azt a szegény csirkét ettették, leöldösték, felkészítették, hűtötték addig, amíg nejem tovább boncolta, és most kétségek dűlnak bennem: nejem finom fűszerezését dicsérjem-e vagy az alapvető-

halaszt sziđjam. Milyen jó lenne, ha a folyamat egységes és általam meghatározható lenne...!

Van adatfeldolgozási lánc is. Ismerek egy vállalatot, ahol kedves hölgyek kiszámítják a táppénzt, a levonásokat, a pótlékokat stb. Majd az apróra darált „információt” beadagolják barátunknak, és az okos masina jövedvében még a bérlistát is kiírja. Mint a mosógépem. Fél nappal a mosás előtt beáztatom a zoknit. Majd jól kidörzsölöm a gyermekeim életképét tanúsító nyomokat. Zutty bele a mosógépbe! Egy óra múltán vegyesnek mondott színekben kapok vissza pár zoknit, amit már csak napokig kell szárítani az automatikus centrifugálás után.

Tanfolyami szünetben mondja mérnök ismerősöm (számítógépet gyártó vállalat alkalmazottja) kérdésemre, hogy mit tesznek a kimenettel (vagyis outputtal): — Egy alkalmas targoncán az üzem olyan sarkába toljuk, ahol nem zavar. A nevezett lista tájékoztat arról, hogy a múlt dekádban melyik gép volt lerobbantva, milyen következménnyel. Mérnök barátom pedig beadagolhatja a gépbe azt, hogy valójában(!) mi történt a múlt dekádban. Elvégre az egyéb ki-



Sokan úgy vélik, hogy a számítógép is a közismert reklámnak megfelelően viselkedik:

„A számítógép akkor érzi jól magát és akkor takarékos, ha jól megtömi”

mutatásoknak konzisztenseknek kell lenniük. Miért nem lehet a változás pillanatában új termelési programot adni a gépnek? Vagy ilyenre javaslatot kérni tőle?

Nem vagyok naiv. Ismerem a gépi korlátokat. Szentül meg vagyok győződve arról, hogy a

feldolgozási lánc elején is és végén is elkenyézettjük barátunkat, a számítógépet, mert lusták, tehetetlenek, pontatlank, megbízhatatlanok vagyunk. És ezért gyors barátunk helyett kétszer, tehetetlenül, pontatlanul és megbízhatatlanul kell előre felkészítenünk és utólag kiigazítanunk pontos

kis barátunk „információt”. És bérelszámoló hölgyek, termelésprogramozó urak szídják barátunkat, aki munkát ad.

No, a nagygépek kisöccse, a személyi számítógép végre megérkezett! Aki picit is komolyan jártsik vele, annak szemérlől le hull a fátvol. Kollégáim, reszkessetek! Ha nem tartjuk távol a mikrogépektől a felhasználót — amiért sokat megteszünk —, akkor kiderül: nem vagyunk már a távoli számítógépipsten titokzatos papjai, akik nehezen érthető áldozatokra kényszeríthetik a felhasználót. — Jé, hiszen itt valóban alapadatot lehet bevinni, és egy kicsit tovább lehet feldolgozni, mint azt nekem mondták, mindenféle ürügyekre hivatkozva — szól majd Kovács egy napon. Kovács, a végső felhasználó.

Azon a napon Szabó, a rendszerfejlesztő, rájön arra, hogy miközben a programozás magaslataira hágott, a másik cizmájára egyre terhesebb lett. Keményen kell kihúzni. Keményen kell lépni annak érdekében, hogy kényelmünket feledve, ellássuk feladatunkat. Amelynek lényegét már Pistike is érzi: feladatunk a szolgálat. A kényelemteremtés.

Ennyit az információkról és a sült csirkéről.

DR. HALASSY BÉLA

data manager

MINDEN EGYSZERŰ, HA VAN EGY JÓ MENEDZSERE!

- Aki:**
- **gazdag piaci és szakmai ismerettel rendelkezik,**
 - **ismeri a korszerű gépeket és alkalmazásukat,**
 - **ért a szervezéshez és a műszaki fejlesztéshez,**
 - **gyakorlott programozó,**
 - **széles választékból rövid határidőre szállít,**
 - **pontosan tudja a feladatát és nagy rutinnal irányítja a többiek munkáját.**

ÖNNEK CSAK

DÖNTENI KELL, A TÖBBIT ELVÉGZI A

data manager
SZAMITASTECHNIKAI KISSZOVETKEZET
1134 Budapest Dózsa György út 150
Postacím: 1553 Budapest Pf. 41
Tel: 202-850/247
Tx: 22-6741

KOMPLEX SZÁMÍTÁSTECHNIKAI VÁLLALKOZÁS

Kulcsrakész ügyviteli rendszerek a **MICROCONTROLL** számítógépcsalád elemeivel

MICROCONTROLL '86-'87 számítógépcsalád

Teljes IBM XT és AT hardver-szoftver kompatibilitás
Több munkahelyes és hálózatos rendszerek kialakítása
Széles körű perifériaválaszték
Grafikus és NYÁK-tervező rendszerek
Mágnesszalagegység és videomagnó-illesztés
COMMODORE floppy illesztés

MICROCONTROLL '84 számítógépcsalád

CP/M kompatibilis operációs rendszer
Nagy teljesítményű lokális hálózat
Kimagasló adatfeldolgozási sebesség
Igen kedvező teljesítmény-ár viszony.

INTELLIGENS **ADAT-ELŐKÉSZÍTŐ RENDSZER**

Tetszőleges számú munkahely
Kezelőtechnikai kompatibilitás
ismertebb rendszerekkel

KOMPLETT **FELHASZNÁLÓI RENDSZEREK**

Egy- és több munkahelyes
felhasználói rendszerek
Szervezés, programozás
Betanítás, adaptálás



PROFESSIONÁLIS SZÁMÍTÁSTECHNIKAI SZAKÜZLET

1132 Bp., Visegrádi u. 6. Tel.: 128-064
Szaktanácsadás
Adás-vétel
Lízing



Számítógépeinket 1-3 hónapon belül szállítjuk
Gyors, megbízható szerviz, installálás
Oktatás, szervezés

Elektronikai és Számítástechnikai Kiszövetkezet
1027 Bp., Szász Károly u. 2. Tel.: 158-428.

a számítástechnikában hagyományos és már megszokott cikkein kívül

bemutatja **ÚJDONSÁGAIT**

**TEDI
AUTOMATIKUS
TELEFONHÍVÓ
ÉS HÍVÓSZÁMTÁROLO
KÉSZÜLÉKÉT**

helyi, belföldi és nemzetközi
távhívásokat
automatikusan bonyolít,
40 hívószámot tárol,
hálózatkimaradás
esetére akkumulátoros
memóriavédelemmel
van ellátva.

IBM kompatibilis
mikroszámítógépét
különböző
felhasználásokra,
teljes
programellátottsággal

**IRODAI
HORDOZHATÓ
SOKSZOROSÍTÓGÉPÉT**

A/4 méretű
4 szín ellátottságú
programozható
folyékony kristály
kijelzésű

**SZEMÉLYI SZÁMÍTÓGÉPEK
PROGRAMJAIT
ÉS TARTOZÉKAIT**

fényceruza
botkormány
alkalmazási programok
játékprogramok
perifériaillesztők
memóriabővítők



Minta utáni árusítás

*MINDEZEKET
A HELYSZÍNEEN
ÁRUSÍTJUK,
illetve előjegyzést
veszünk fel*

BITEK ÉS FIGURÁK

Állásértékelés V.

Figuratámadás és -védelem

Az értékelőfüggvénynek eddig ismertett komponensei a sakkjáték stratégiai szempontjait ölelték fel. Ezek a stratégiai szempontok biztosítják az állás felmérését és képessé teszik a programot az állásban lezajló folyamatok lényegi megértésére. A figurák támadásának és védelmének figyelembevételével a programunkat valamelyest taktikailag is felvértezzük, és így realisabban értékeli az állás dinamikáját és taktikai lehetőségét. Mint a későbbiekben látni fogjuk, ez a taktikai játékhoz önmagában még nem elegendő, de a hozzá vezetőlépcső első foka.

A figurák támadása és védelme — mint kvantitatív mennyiség — meghatározza az egyik félnek az ellenség figuráira kifejtett támadási esélyeit és a saját figurák védelmének mértékét. Tudjuk, hogy az a játékos van előnyben — figuratámadás és védelem szempontjából —, amely figurái jobban védettek és az ellenfél bábjait aktívabban támadja. Ez nem jelenti azt, hogy ebben az állásban az előnyben levő félnek azonnali taktikai esélyei vannak, hanem csak azt, hogy kombinációs lehetősége nagyobb. David Lavy megemlíti könyvében ezeket az értékeket is, amit az ábrán látható példán mutat be.

Ez az állás a francia védelem egyik változata, amely a következő lépések megtételével alakul ki:

- e2 – e4, e7 – e6
- d2 – d4, d7 – d5
- h1 – c3, f8 – b4

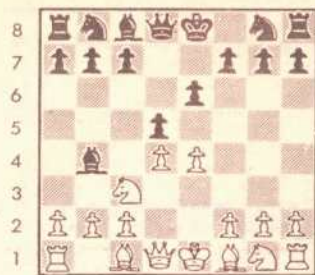
Látható, hogy világos csak e4 gyalogjával támadja az ellenfél

d5 gyalogját, így egy támadó értéke van. Itt nem kell figyelembe venni, hogy a d5 gyalogot sötét biztonságosan védi az e6 gyaloggal és a d8 vezérrel. Ez az értékelőfüggvény egy másik komponensében realizálódik. Sötét d5 gyalogjával világos e4 gyalogját támadja, b4 futójával a világos c3 huszárt. Tehát világos egy támadásával szemben sötétnek két támadó értéke van.

A figuratámadás kvantitatív meghatározása több módon történhet. Meghatározhatjuk mindkét fél esetében az ütések számát, figyelembe véve a kötéseket is. Jelen esetben nem a félig legális lépésekből adódó ütések számát kell figyelembe venni, hanem csak a legális lépésekből adódókat. Ez azért lényeges, mert nagyon sok taktikai lehetőség éppen a kötések kihasználásán alapul. Ezért ezt a programnak feltétlenül figyelembe kell vennie. Az előbbi állásban is észrevehető, hogy b4 futó lekötöti a c3 huszárt, és így világos e4 gyalogja védtelen marad. Sokkal pontosabb értékhez jutunk, ha a támadó figura értékét a megtámadott figura értékéhez viszonyítjuk, és figyelembe vesszük, hogy a megtámadott figura védett-e. Ezek alapján a támadás értékét a következő módon is meghatározhatjuk:

$$\frac{\text{a „megtámadott” értéke}}{\text{a támadó értéke}} \\ \text{védők száma} + 0,25$$

Ezt a képletet felhasználva az előbbi diagramon a következő figuratámadási értékeket kapjuk:



a b c d e f g h

- e4 gyalog támadása
d5 mezőre:
(1:1) : 2,25 = 0,4
- d5 gyalog támadása
e4 mezőre:
(1:1) : 0,25 = 4,0
- b4 futó támadása
c3 mezőre:
(3:3) : 1,25 + 0,8

Láthatjuk, hogy sötétnek 4 x 4 egység a figuratámadási előnye.

A figuravédelem meghatározásához a védő figura értékét is figyelembe kell venni. Ezt a legegyszerűbben a védő figurák reciprok értékének összegével állapíthatjuk meg, amiből kiderül, hogy a legelőnyösebb, ha a saját figurát a lehetséges legkisebb értékű figurával védjük meg. Ezek szerint a megtámadott figurát elsőként gyaloggal, ha nem lehet, akkor huszárral, futóval, bástyával, vezérrel, majd királlyal célszerű megvédeni. Ez azért fontos, mert ha a védelemre nagyobb értékű figurát használunk, amely egyben hatékonyabb is, akkor ez a támadásban hátrébb szorul és passzívvá válik. Ezt a módszert az előző állásra alkalmazva, a következő értékeket kapjuk:

Világos:

- $$\begin{aligned} \text{Fe1: } 1/5 + 1/9 &= 0,3 \\ \text{Vd1: } 1/4 &= 0,3 \\ \text{Ff1: } 1/4 &= 0,3 \\ \text{Hg1: } 1/5 &= 0,2 \\ \text{a2: } 1/5 &= 0,2 \\ \text{b2: } 1/3 &= 0,3 \\ \text{c2: } 1/9 &= 0,1 \\ \text{f2: } 1/4 &= 0,3 \\ \text{g2: } 1/3 &= 0,3 \\ \text{h2: } 1/5 &= 0,2 \\ \text{Hc3: } 1/1 &= 1,0 \\ \text{d4: } 1/9 &= 0,1 \end{aligned}$$

Sötét:

- $$\text{Hb8: } 1/5 = 0,2$$

- $$\begin{aligned} \text{Fc8: } 1/9 &= 0,1 \\ \text{Vd8: } 31/4 &= 0,3 \\ \text{Hg8: } 1/9 &= 0,1 \\ \text{a7: } 1/5 &= 0,2 \\ \text{b7: } 1/3 &= 0,3 \\ \text{c7: } 1/9 &= 0,1 \end{aligned}$$

Az egyes figurák értékeinél tisztán látható, hogy ha a gyalogot egy egységnek vesszük, akkor a huszár és a futó 3 egység, a bástya 5 egység, a vezér 9 egység. Kérdés, a királyt hogyan értékeljük anyagi szempontból. Mivel ez kiténtelt báb a sakkjátékban, anyagi értékéről nemigen szoktunk beszélni. Jelen esetben kivételt teszünk, és a királyt is anyagi értékben részesítjük. Ezt megtehetjük dinamikusán úgy, hogy amíg a táblán levő tiszték száma egy bizonyos értéket meghalad, addig a király értékét száznak vesszük, a végétkéba történő átmenet esetén pedig négygyel számolhatunk. Ezzel elérjük, hogy a középjátékban, amikor királyunkat könnyen megtámadhatják és mattveszélybe kerülhet, ne hagyja el biztonságos helyét, hogy valamelyik figuráját megvédje. A végétkéban viszont — ahol a király centralizálása igen fontos, és nyugodtabban is mozoghat a tábla közepén bántódás nélkül — az anyagi értéket vehetjük négy gyalogegységnek. Ezzel értéke egy könnyű tisztnél nagyobb lesz, de a bástyánál kisebb, ami általában reális is.

A konkrét figuratámadás kiszámításának módjával később foglalkozunk, miután a királytámadás és királybiztonság kvantitatív meghatározását ismertettük.

KOVÁCS P. ATTILA

NEMZETKÖZI FOTÓ- ÉS RAJZPÁLYÁZAT

A Nemzetközi Távközlési Egyesület (UIT) nemzetközi fotó- és rajzpályázatot hirdet a 160 tagországban élő fiatalok számára. Hazánk részvételét a nemzetközi versenyben a Magyar Posta szervezi. A verseny általános témáját „A távközlés a fejlődésért” címmel jelölték meg a kiírók. A pályázók gondolkodását négy altémával segítik:

A műhold csökkenti a távolságokat.

A telefon a hétköznapi életben.

A televízió oktatásra is alkalmas.

Mire jó a számítógép?

A verseny résztvevői fotók, rajzok, festmények, egyéb illusztrációk (pl. kollázsok) segítségével dolgozhatják fel témáikat. A versenyben 8-18 év közötti fiatalok vehetnek részt három korcsoportban: 8-12, 13-15, 16-18.

A magyarországi pályázók-
nak 1986. november 30-ig kell
eljuttatniuk műveiket a Ma-
gyar Posta Központja tájékoz-
tatási és marketing osztályá-
hoz, személyesen a Budapest
XII. Krisztina krt. 6-8. vagy

postán az 1540 Budapest cím-
re. A Magyar Posta által össze-
hívott nemzeti zsűri dönti el,
hogy a beérkezett munkák kö-
zül melyek kapják a Magyar
Posta díjait és közülük melye-
ket küld el a posta (kategórián-
ként maximum 10-et) a genfi
nemzetközi versenyre.

A tagországokból Genfbe
küldött pályaműveket 1987.
október 20-27. között mutat-
ják be a TELECOM '87 világ-
kiállításon. A nemzetközi zsűri
itt választja ki a három korcso-
port díjazására javasolt pályá-
munkáit.

1983-ban egyébként 53 or-
szág nemzeti pályázata nyo-
mán 300 ezer alkotás született,
a nemzetközi zsűrinek pedig
1500-ból kellett választania.
A magyar fiatalok rendkívül
eredményesen szerepeltek, a
16-18 évesek között Budai Tá-
más világső lett.

A pályázattal kapcsolatos
kérdésekkel a Magyar Posta
Központjának tájékoztatási és
marketing osztályát kereshetik
meg az érdeklődők az 555-575
vagy a 754-893 telefonszámon.

PRODUKTORG
SZERVEZÉSI VÁLLALAT

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

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

1027 Budapest, Fő u. 68.
Telefon: 154-090

*Bővebb
felvilágosítással
készséggel állunk
rendelkezésükre!*

Beruházók vállalkozók figyelmébe!

*A fejlesztések, beruházások,
fővállalkozások sikeres irányításához
nélkülözhetetlen a különféle
költségek alakulásának állandó
nyomon követése és a költségek
folyamatos ellenőrzése.
Ezt a követelményt elégíti ki*

a KFR-MICRO

költségfigyelési programcsomag!

Alapvető szolgáltatások: költségek
alakulásának nyomon követése,
költségösszesítések készítése, százalékos
előrehaladások számítása, kerettüllépés
figyelése, várható bekerülési összeg
számítása stb.

**A KFR-MICRO a következő számítógépeken
üzemeltethető: TAP-34, M08X
PROPER-8, PROPER-16, COMMODORE-64,
IBM PC**

*Tekintse meg a SYSTEM Szervezési Vállalat
kiállítását a Budapest Sportcsarnokban
1986. október 3-8. között*

*a C/25-ös standon
az ORGTECHNIK kiállításon*

86

orgtechnik
hungaria
budapest '86

Keresünk megvételre

jó állapotban lévő

VIDEOTON

gyártmányú

VDT 52103, VPC,

VPPC displayket.

Villamosenergiaipari

Kutató Intézet

Ügyintéző:

Papp György osztályvezető

Telefon: 178-698

ADOK-VESZEK-CSERÉLEK

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

■ Atari 800 XU új személyi számítógép magnóval, német nyelvű leírásokkal, áron alul eladó. Járfás, Ajka, Márvány u. 1. 8451.

■ ZX81-et keresek megvételre, vagy árának megfelelően könyvet, alkatrészt, barkács-, fotószkózt adok érte. Lehetőleg bővítővel! Varsányi Gábor, Kiskörös, Árpád u. 12. 6200.

■ C64-re készült programokat cserélek. Szomor László, Bp., Báthori u. 18. 1054. Telefon: 320-807.

■ Commodore 16 (módosítással C64) idegen szavakkal tetszés szerint feltölthető *nyelvtesztizotár* programkazetta utánvétellel (300 forintért) eladó vagy/és ZX81-hez turbo 81; 1 k-s autóverseny, BASIC nyelvoktató kazettákat cserélek C16 programokra. Dakó Csaba, Dunaújváros, Martinovics u. 31. I. 1.

■ Commodore 64-hez MC 381 típusú, BRG gyártmányú adatmagnó 4000 forintért eladó. Gyári garancia novemberig. Debreceni László. Telefon: 353-900/543-as mellék.

■ Laser-210 számítógép magnóval, botkormánnyal eladó vagy bérbé adó. Tanulásra kiválóan alkalmas. Bártafi Barnabás, Agárd, Széchenyi u. 74. 2484. Telefon: 239.

■ PL-150 típusú szalaglyukasztó elektronikával eladó. Várszegi Sándor, Bp., Beregszász u. 77. 1118.

■ Olcsón eladó vagy erősítőre cserélhető egy TI 99/4A 16 k-s színes számítógép programokkal. Jelinek András, Bp., Mogyoródi út 117. 1141.

■ Commodore 64+VC 1541 lemezegység + 38 db lemez tele programokkal + 2 db botkormány 57 ezer forintért eladó. Strelli József, Bp., Török Flóris u. 28. VI. 36. 1203. Telefon: 451-350, 8-16 óráig.

■ C64-es játékprogramokat cserélek magnót használókkal. Soós Sándor, Szeged, Gyöngyvirág u. 18/A. V. 12. 6723.

■ Eladó ZX—Spectrum 48 k — 14 000 forint, interface II. — 3000 forint, Computone magnó — 3000 forint, kb. 50 db program — 1500 forint, kb. 1500 oldal irodalom — 1500 forint. *Az egész együtt csak 22 000 forint!* Bendik István, Bp., Harmat u. 71/B. 1104. Telefon: 572-824.

■ Eladó a Mikroszámítógép Magazin 1984-ben és 1985-ben megjelent összes száma. Csatlós Béla, Mezőtúr, Mező 1. 47.

■ TI 99/4A számítógép botkormánnyal, magnóval és játékprogrammal 20 000 forintért eladó. Telefon: 62/56-607.

■ VIC—20-hoz oktató- és játékprogramok olcsón eladók. Krusó Oszkár, Tát, Néphadsereg u. 21. 2534.

■ Eladó ZX—Spectrum, 48 k, sok magyar nyelvű dokumentációval (BASIC és gépi kódú programozás, hardversegédlet, szervizkönyv), GP—50S nyomtató és sok program. Varga Zoltán, Bp., Murányi u. 36. IV. em. 1078. Telefon: 216-975.

■ ZX81 16 k-s bővítővel eladó. Id. Krasznai László, Bp., Sáfány u. 67. 1116. Telefon: 867-745.

■ 64 kbájtos számítógép floppyval, programokkal olcsón eladó. Telefon: 410-158.

■ C64-re játékok és másolóprogramokat cserélnék. Elsősorban lemezen, de kazettán is. Lucz Géza, Kaposvár, Kinizsi ltp. 5. 7400.

■ C64-es programokat cserélek bel- és külföldi partnerrel. Tóth József, Bp., Bercsényi u. 2. 1041. Telefon: 890-226.



HÁLÓTERVEZÉS = HSZR-MICRO

Ismerjék meg és használják Önök is a HSZR-MICRO hálótervezési programcsomagot,

melyet eddig több mint száz hazai vállalat, illetve intézmény alkalmaz sikeresen. Referenciákkal rendelkezünk: beruházásszervezés, kivitelezésszervezés, karbantartás-szervezés, termelésirányítás, mezőgazdaság, oktatás stb. területeken. A HSZR-MICRO 8 és 16 bájtos mikroszámítógépeken üzemel a COMMODORE-tól az IBM PC AT-ig.

A HSZR-MICRO FŐBB SZOLGÁLTATÁSAI:

- hálószerkesztés és rajzolás képernyőre, sornyomatóra,
- kombinált MPM/CPM módszer,
- vonalas ütemvert változatható időtengellyel,
- erőforrásterv, aggregáció, hisztogram,
- aktualizálás, nyomon követés, újraütemezés,
- szabadon változatható output tablók.



Tekintse meg a SYSTEM Szervezési Vállalat kiállítását a Budapest Sportcsarnokban 1986. október 3-8. között a C/25-ös standon az ORGTECHNIK kiállításon

PRO-KONTRA GM.

1074 Budapest,
Csengery u. 7 fszt. 1/a
☎ 417-893

Cégünk ajánlataiból:

- 64 k-s memóriabővítő Commodore C16-hoz
- Személyi számítógépek és összes tartozékaik garancián túli javítása átalánydíjas szerződéssel is
- FAST-VC-1541 kommunikációgyorsító rendszer a C64-hez (minden gép—floppy közötti és floppy belüli műveletet 4-12-szeresére gyorsít)

Konfiguráció beszereléssel együtt 7000,— Ft

- Abszolút programvédelem C64-hez: cartridge-ben, műgántával kiöntött EPROM-ba égetéssel, MÁSOLHATATLANNA teszi programjait
- IEC buszról vezérelhető méréspontváltó egység 2 x 30 vagy 1 x 60 bemenet, 2 ill. 4 kimeneti csatornával
- Digitális kijelzésű elektronikus óra 8 x 14 cm-es digit-mérettel, különféle színekben
- Egyéb, közösen sorozatú fejlesztések igény szerint

Pro-Kontra Automatizálási,
Műszaki Tanácsadó és Közvetítő GM

Budapest, Csengery u. 7. fsz. 1/a 1074
Tel.: 417-893
Levél cím: Budapest, Pf. 72. 1581
Telex: 22-7770

Nemzetközi Számítástechnika — Internacia Komputado

Az 50 országban olvasott magazin példányonként 30 forintért kapható az SKV Könyvesboltban.
Bp. II., Keleti Károly u. 10.
Telefon: 158-018

Szidarovszky Ferenc—Molnár Sándor:
Játékelmélet műszaki alkalmazásokkal. Többcélú programozás, klasszikus és differenciáljétekek.
 (Budapest, 1986. Műszaki Kiadó, 240 oldal. Ára: 75,— Ft.)

Bonyolult rendszerek elemzése során gyakran előfordul, hogy a vizsgálati szempontok egymásként ellentmondóak, esetleg nem összemérhetők. A konkrét feladatok jellege szerint játékelméleti, többcélú programozási módszert kell választani, vagy a differenciáljétekek valamely módszerét kell alkalmazni. A könyv ezzel a három, egymással szoros kapcsolatban álló témakörrel foglalkozik.

Minden C64-tulajdonos, programozó, felhasználó örömmel üdvözölheti a Novotrade Rt. vállalkozását, a Data Becker kiadó C64-gyel foglalkozó sorozatának folyamatos, magyar nyelvű megjelenítését. A kötetek sok kipróbált, ellenőrzött BASIC és assembly nyelvű példaprogram listáját tartalmazza fénymasolt, nem nyomdai úton szedett formában. Az aszsembler lista mellett mindig megtalálható a BASIC betöltőprogram is. Az így közölt programrészeket, programok garantáltan jók, hibátlanok. Úgyanez nem mondható el a szöveg közben magyarizáltaként szereplő néhány soros programokról, parancsokról, utasításokról. Ezek között több sajtóhibát találtam. Tanár vagyok, így az ismertetések sorrendjét egyben didaktikai sorrendnek is ajánlom. Az egyes könyvek átfedései elkerülhetőek lettek volna ugyan, de sosem árt a téma magasabb szinten való, más szempontú ismétlése, újratárgyalása. Az árakról nem beszélek.

Angerhausen — Englisch
— Gerits — Schellenberger:
A BASIC programozás
magasiskolája C64-esen
 (Budapest, 1985.
DATA BECKER — NOVOTRADE RT,
 183 oldal. Ára: 308,— Ft.)

Az első fejezet a programozásmélet és az ösztönös programkészítés összevetése gyakorlati problémák tükrében. Nem mondja ki ugyan, de világosan szemlélteti, hogy a programkészítés lényegesen szélesebb, általánosabb fogalom, mint a kódolás fogalma. Az ötlet születésétől a feladat precíz megfogalmazásán keresztül, az algoritmus-készítésen át a változók megválasztásával, a képernyők, a számítás, az adatfájlok, a nyomtatási képek tervezésével, a folyamatos és részletes dokumentálás szükségességével foglalkozik. Az egyes részek tárgyalásai sok hasznos, többször alternatív megoldási lehetőséget, segédeszközt ajánl, mutat be.

A második fejezet a program részeivel foglalkozik a klasszikus felosztás — bevétel, feldolgozás, kivétel — csoportosításban. Ezekkel kapcsolatban olyan kisebb-nagyobb, de rendkívül általános és hasznos tudnivalót, eljárást közöl, melyek nélkül nem képzelhető el egy jó program. A bevétel tárgyalása közben a szerzők megadnak egy általános, egységes „bolondbiztos” és

barátságos képernyőmaszk-készítő és adatbeolvasó rutint, melyet mindenki beépíthet(ne) saját fejlesztésű programjába. A feldolgozás tárgyalásakor a változók helyes típusának megválasztása és a számítási pontosság problémája mellett felhívják az olvasó figyelmét a különböző rendezési eljárások hely- és időigényére. Ismertetik a buborék- és gyorsrendezést. Hiányolom itt legalább néhány keresési, kiválasztási algoritmus, illetve a memóriában való rendezések szélesebb körének ismertetését, nem beszélve a háttérrolón őrzött adatok összefuttatásáról.

Az adatok kiírásával foglalkozó rész összevonja az eredmények nyomtatón való megjelenítésének és a lemezen történő tárolásának problémáit. Ismerteti a VC 1541-es lemezegység SEQ és REL típusú fájljainak műveleteit, majd sajnálkozva közli, hogy a sokkal hasznosabb ISAM fájl megvalósítását az operációs rendszer nem támogatja. Ezt követi a könyv leglényegesebb része, a szerzők által létrehozott QUISAM fájl részletes ismertetése a hozzá szükséges HASH kód leírásával együtt. Végül a nyomtató kerül sorra. Az összegzés mottója: a felhasználót az eredmények érdeklik! Egy általánosan használható, a programozó által paraméterezhető nyomtató rutin zárja a fejezetet.

A harmadik fejezet a programozástechnikai kérdések tárháza, bár itt fejeződik be az első fejezet gondolatmenete is. Tárgyalja az élő (általános problémakörre készi-

tett) programok problematikáját: a paraméterezést. Ismerteti a menütechnika és funkcióbillentyűk használatának előnyeit, mindez a programok arculatának egységesítése miatt. Visszatérve a programozás lépéseire, röviden vázolja a „kész” programok tesztelésének szükségességét, tanácsokat adva a lehetséges segédeszközökre, bővítésekre vonatkozóan. Sajnos a különböző tesztelési módszerek ismertetésére a szerzők nem térnek ki. Végül hangsúlyozzák a rövid, világos, közérthető kezelési utasítás szükségességét. Itt-ott megemlítik ugyan a programlista olvasható voltának, a program fejleszthetőségének fontosságát, azonban ennek tárgyalására nem vállalkoznak. A fejezetet az építkező, (alulról felfelé) és az elemző (felülről lefelé) programozási módszerek megemlítésével, valamint három memóriatakarékossági eljárás megoldásával zárják.

A negyedik fejezet az alkalmazásoké. A szerzők konkrét példákon keresztül mutatják be az eddig ismertetett elvek megvalósításait. Tárgyalnak egy paraméterezett raktárnyilvántartást, egy általános soros fájl listázót, egy egyszerű szövegszerkesztőt, a QUISAM fájl alkalmazását irodalomjegyzék nyilvántartására, valamint egy ismerőseink címnyilvántartását végző programot.

A ötödik fejezet a programkészítéshez felhasználható segédprogramok egy részét foglalja össze. Ismerteti a MASTER 64 és a STRUKTO 64 programokat.

ÉNEKES FERENC

Beruházók figyelme!

A beruházások pénzügyi nyilvántartásához

ORGTECHNIK
hungaria
budapest'86

COMMODORE 64
számítógépre
ajánljuk

a számla-nyilvántartási, forráskezelési és tőkésgep-nyilvántartási programcsomagokat.

Alapvető szolgáltatások:
a kötelező nyilvántartások elvégzése, adatszolgáltatás a jelentésekhez, információ a beruházások irányításához.

Beruházásszervezést, komplett beruházásirányítási rendszer bevezetését is vállaljuk.

Tekintse meg a SYSTEM Szervezési Vállalat kiállítását
a Budapest Sportcsarnokban
1986. október 3—8. között
a C/25-ös standon

az ORGTECHNIK kiállításon

PÉNZÜGYI SZERVEZŐ ÉS TANÁCSADÓ VÁLLALAT

több évtizedes tevékenységei – szervezés – tanácsadás mellett újabb szolgáltatásokkal segíti a hatékony vállalati gazdálkodást:

- értékelemzés
- gazdasági elemzés
- könyvszakértői tevékenység
- lízing tevékenység

Szervezési tevékenységünk alapja a hazai és külföldi elektronikus adatfeldolgozó berendezésekre, kiszámítógépekre kidolgozott programcsomagok, egyedi szervezések.

A hagyományos ügyviteli gépek (R. 1355, R. 1711, R. 1720, R. 5100-as gépcsalád) mellett profilunkba illesztettük a hazai piacon megjelent géptípusok közül a következőket:

- **COMMODORE 64**
- **ROSY 80F**
- **VT 16**
- **R. 1715**
- **IBM PC XT**
- **ATARI 800XL**

Programcsomagjaink, mintaszervezéseink az ügyviteli folyamatok teljes körét felölelik, elsősorban az ipar, mezőgazdaság, költségvetési ágazatok területén. Szakmai tanácsadással, konzultációs lehetőségekkel, bemutatókkal segítjük a helyes gépkiválasztást. Megbízóink igényére egyedi szervezési munkákat is vállalunk a felsorolt géptípusokra. Szoftve-reink bevezetését folyamatosan szervezett oktatással biztosítjuk.

Rendszeres felvilágosítás:



Bemutatóterem
Bp. VI., Paulay Ede u. 15.
Tel.: 421-764, 220-607



HAVI AJÁNLATUNK:

- EPSON illesztőkártya
- Winchester illesztőkártya
- 14,5 Mbájtos Winchester
- olcsó hajlékony lemez
- Streamer (backup)
- mágnesszalagok

VÁRJUK ÖNT:

hétfő	
kedd	9—16 óráig
szerda	
csütörtök	9—17 óráig
péntek	9—15 óráig

LEGYEN A PARTNERÜNK!

Keresse fel bizalommal a Műszaki Árut Értékesítő Vállalat és az „5G” Számítástechnikai Szolgáltató Kiszövetkezet Számítástechnikai szakboltját!

Címe: Budapest XIII., Victor Hugo u. 33.
Tel.: 494-782

*Vevőszolgálat és
szaktanácsadás!
Eladás és lízing!
IBM PC/XT, IBM PC/AT
COMMODORE, ATARI
számítógépek*



Menetjegyet mikrogéppel

A Nyugati pályaudvaron megjelent az országban az első nyolc mikroszámítógép, amely a MÁV-menetjegyek kiadását végzi. Ezek a célgépek Intel 8085 típusú mikroprocesszoron alapulnak, s egybeépítve tartalmazzák a megjelenítőt, a nyomtatót, a billentyűzetet és egy hajlékonylemez tartózt is. A célorientált gépialakítás a kezelést egyszerűsítést szolgálja.

A mikrogép alkalmazása végső soron nem gyorsítja ugyan magát a menetjegy-kiszolgáltatást, viszont közvetve jelentős, az utasok számára is kedvező változást eredményez. Eddig a pénztárosnál kint lévő jegyállomány igen nagy értéket képviselt. Ennek a kötelező, rendszeres leltározása vitte el a havi munkaidőalap jelentős részét. A számítógép alkalmazásakor a pénztárosok csak üres tekerceket kapnak, ami értékük csak a rányomatás során válik. A forgalom összesítését egy gombnyomásra azonnal elvégzi a gép, amely komoly munkaidőalapot szabadít fel. Az utasok számára ez több pénztárablak egyidejű nyitva tartásával válik érezhető előnnyé.

GYSEU 106F 86.03.19
Budapest-Nyugati pu.
Kiséri kútba egy ótra
ERVENYTELEN
Bármely vonat 2. oszt.
Érv: 86.03.19 +1nap
0 5 0 km SFt 02806

Mintaként kinyomtatott kutyajegy

86.03.19	106F	011
Db. 02806	Ft.	0194964
Db. 02806	Ft.	0194964
Bevétel:	Ft.	0000000
Rontott:		
Db. 00000	Ft.	0000000
Terhelés:	Ft.	0000000

1061	Ft.	0000000	000
1062	Ft.	0000000	011
1063	Ft.	0000000	000
1064	Ft.	0000000	000

Elszámolási összesítő minta a gépen dolgozó négy pénztárosra lebontva

Új házi nyomtató

VT 21 100 néven, Centronics licenc alapján új mikronyomtatót kezdett gyártani a Videoton. Az alig 3 kg súlyú mátrixnyomtató egy sorba maximum 80 karakter nyomtatására képes. Háromféle papír befogadására is lehetővé teszi: a szabványos A/4-es papírt gumihengerrel, a leprellőt úgynevezett traktorral továbbítja, de tekerespapír-adaptere is van. Nyomatási sebessége 90 karakter/perc, ami kategóriájában igen jó, hiszen megfelel a hazai professzionális mikronyomtatók sebességének. Külön piszkos-tisztázati nyomtatási lehetősége is van. A mikroszámítógéphez annak mind a soros, mind a párhuzamos illesztőjén keresztül csatlakoztatható. Az új nyomtatót a Videoton már be is mutatta saját házi számítógéphez, a TV-Computerhez csatlakoztatva. Ára: 25 ezer forint.

GENERALCALC

Háromdimenziós táblakezelőt készített az NDK-beli robotron cég. A 256 x 30 x 15 elemmel kezelő program már elkészült. Jelenleg integrált szoftverrel alakítását végzik, azaz ki-egészítik szövegkezelőt, adatbázis-kezelőt és grafikus funkciókkal is. A fejlesztést az IBM PC-vel kompatibilis, robotron gyártmányú A 7100 típusú gépen végzik. Már folyik a GENERALCALC-nak nevezett termék — várhatóan 1987 elején történő — külföldi piacra hozásának előkészítése is. Ez ügyben a robotron nemcsak amerikai, hanem magyar forgalmazókkal is tárgyal.

A pénz számolva...

A pénztételek nagyobb hivatalaiban több száz ezer forint az érmeforgalom. Az ebből eredő feldolgozási munkát könnyíti meg a kiskörösi PRO-TOKON kisvállalkozás tőkés importot helyettesítő mikroszámítógépes érmeszámoló készüléke, mely a tavaszi BNV-n mutatkozott be először

a nagyközönség előtt. A gép eddigi eladási sikerei arra a reményre jogosítanak fel, hogy a KGST és a tőkés piacon is lesz rá kereslet. A posta és az OTP után jelenleg a Magyar Nemzeti Banknál folyunk a gép üzemi „nyüzőpróbiái”.

Összefogásból számítógép

Bács-Kiskun megyében nagy hagyománya van az „Egy üzem — egy iskola” mozgalomnak, melynek keretében a megye vállalatai rendszeresen segítséget nyújtanak egy-egy iskolának. A nyári vakáció előtt új felhívás született: a számítógépes oktatás feltételeinek megteremtésére szólított fel. Hatására sok településen megmozdultak a termelői kollektívák. Orgoványon például a községi tanácsnál nyitottak és célra számlát, s olyan ütemben gyarapodott a pénz, hogy szeptemberben már nem egy, hanem két számítógépet vásárolhattak az általános iskolának. A kezdeményezésnek, a társadalmi összefogásnak köszönhetően ezen az őszön már az orgoványi gyerekek is megkezdhetik a számítástechnika alapjainak elsajátítását.

Urányszennyeződés

Nagy-Britanniában a Harwell laboratórium kutatási programot kezdeményezett az integrált áramkörök katasztrofális meghibásodását előidéző urán és egyéb szennyezők eredetének kimutatására. Ezek radioaktivitása következtében keletkeznek ilyen hibák. A kibocsátott alfarészecskék által képviselt villamos töltések megváltoztatják a memórialaparamkörök állapotát, vagy az aritmetikai áramkörök hibás működését idézhetik elő. Már többször előfordult, hogy a radioaktív szennyezőelemek következtében létrejövő számítógép-meghibásodás miatt egész üzemek működése automatikusan leállt.

A Harwell laboratóriumban kidolgozták azt az elemzési eljárást, amely minimális mennyiségű urán jelenlétét ki tudja mutatni. A vizsgálat során a felvezető anyagot polimidd főléval vonják be és neutronugrás hatásának teszik ki. A hasadási termékeket fényszerűen réteg regisztrálja, amelyet előhívás után mikroszkópon vizsgálva, pontosan meghatározható a jelenlevő urán mennyisége. Az eljárást felhasználják az integrált áramkörök gyártásához szükséges szilícium, gallium-arszenid és müanyag tokozások vizsgálatára.

Fuvardíj-elszámolás

1983—85 között a hazai intézmények tízezres nagyságrendben vásárolták a Commodore 64-eket professzionális célokra. Ismerve e gép korlátait, a FÜTI Mikroergo leányvállalata keresett egy olyan alkalmazást, amely nem igényel túl nagy adatforgalmat, a nyilván tartott tételszám százas nagyságrendű, s mégis gyakori, tipikus feladat. Így jutottak el a fuvardíj-elszámoló rendszerhez. Ez 120-150 gépkocsira elvégzi a tarifóri, statisztikai feladatokat, és a fuvarokmányokat is elkészíti. Ha a gépkocsialomány nagyobb, akkor még egy hajlékonylemez tartólt kell a géphez csatlakoztatni.

A rendszer gyorsan népszerűvé vált; alig néhány hónap alatt elkészülte után már tucatnyi helyen megvásárolták és alkalmazták. Ez motiválta a kidolgozókat arra, hogy már IBM PC-vel kompatibilis gép-re is átdolgozták.

Bővíthető hálózat

A végtelenségig bővíthető a SZÁMSZÖV Kiszövetkezet által az IBM PC/AT gépekre kifejlesztett, ötletes lokális hálózat. Két kategóriába tartozó AT-gepeket tartalmaz: van közöttük egyszerű munkahely és csoportvezérlő. Az utóbbihoz legfeljebb négy munkahely tartozhat, de lehet közöttük egy újabb csoportvezérlő is. A hálózatban belül bármely gépről bármely periféria elérhető.

LEGÚJABB
SZÁMÍTÁSTECHNIKAI KÍNÁLATUNK
NAGY VÁLASZTÉKÁVAL
VÁRJUK ÜGYFELEINKET
AZ ORGTECHNIK '86
KIÁLLÍTÁS
58-AS STANDJÁN

Fotoelektronik ISZ





PRONET LOKÁLIS HÁLÓZAT

HARDVER

- A PROPER—16/A, PROPER—16/W, PROPER—16/m IBM PC kompatibilis számítógépcsaládból összeállítható hálózat kétirányú busz szervezésű
- A hálózat kiépítéséhez a PROPER—16 számítógépbe bedugaszolható lokális hálózati kártya szükséges
- A hálózat állomásai közös átviteli közegre kapcsolódnak
- Könnyű a hálózat konfigurálása és újabb állomások beiktatása

Főbb műszaki jellemzők

- adatátviteli sebesség: 1 Mbit/s
- adatátviteli közeg: csavart érpáru — vagy telefonkábel
- a főkábel hossza: maximum 500 m (ajánlott hossz max. 300 m)
- Az állomások elméleti maximális száma: 255 (a hálózatba kapcsolt erőforrásoktól és a felhasználás jellegétől függően ajánlott szám: 2-10)

SZOFTVER

- PROPOS V.3.02 vagy ezzel kompatibilis operációs rendszer
- Többfeladatos, többfelhasználós környezetben PROMOS operációs rendszer kiegészítés
- A hálózat szolgáltatásainak elérése külső parancsok segítségével, operátori konzolról vagy programból (felhasználói interfész) biztosított

AZ SZKI – STABIL PARTNER!

Számítástechnikai Kutató Intézet és Innovációs
Központ

1251 Budapest, Pf. 19.



Információ: SCI—L Számítástechnikai Informatikai Fejlesztő
Leányvállalat
1011 Budapest I., Iskola u. 10.
SCITEL Számítástechnikai Fejlesztő Leasing
Leányvállalat
1015 Budapest I., Donáti u. 35—45.