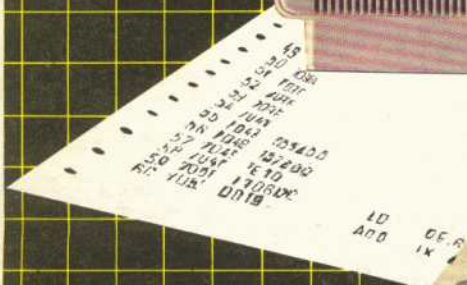




# mikro *Bevezetés* 36 Ft számítógép magazin 12



14. oldal

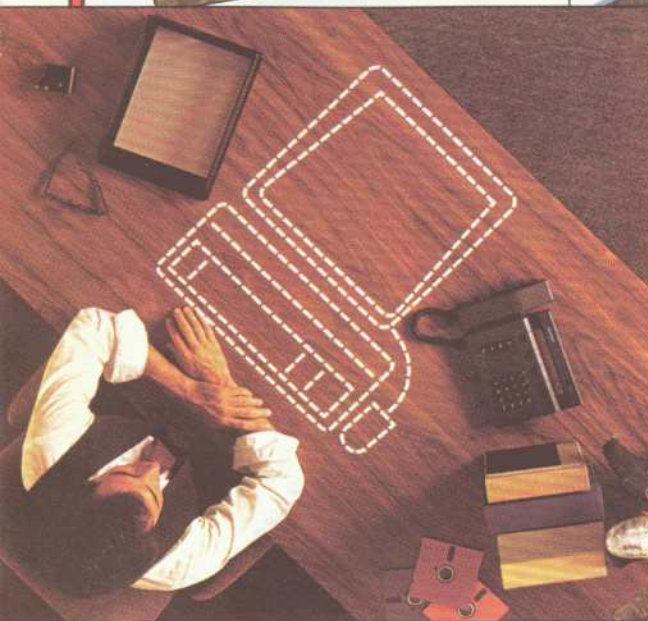


**Vigyázat!  
Jelvaj!**

15. oldal

Az új színhely

# SICOB





# mikro számítógép magazin

5. ÉVFOLYAM  
1987/2. SZÁM

## A NEUMANN JÁNOS SZÁMÍTÓGÉPTUDOMÁNYI 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)**

**Petrőczy Judit  
(könyvek)**

**Simonyi Endre  
(klub)**

**Varga András  
(iskola-számítógép)**

**Címképünk:  
Perényi József munkája**

**mikro számítógép  
magazin**



**Hátso borítónk képe  
Kozák Gábor munkája.  
Olvasóink hasonló jellegű  
képeit, rajzeit, grafikáit  
szívesen fogadjuk a szerkesztőség**

**Felelős szerkesztő:  
Könyves Tóth Pál**

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

**Levél cím  
1371 Budapest  
Pf. 433.**

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

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

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

**Terjeszti a Magyar Posta  
Előfizethető a hírlapkézből  
hivataloknál  
és a Posta Hírlapelőfizetési  
és Lapellátási Irodáján  
(1900 Budapest V.,  
József nádor tér 1.)  
vagy átutalással a 215-96 162  
pénzforgalmi jelzőszámra.**

**Megjelenik havonta  
Egy szám ára 30,— Ft  
Előfizetési díj:  
egy évre 360,— Ft  
fél évre 180,— Ft  
Külföldön terjeszti  
a Kultúra,  
1389 Budapest, pf. 149,  
és a Magyar Média  
1932 Budapest, pf. 279.  
86-253**



**Szakra Lapnyomda  
Budapest (86-4731)  
Felelős vezető:  
Csöndeb Zoltán vezérigazgató**

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

### Tartalom

Alkalmazás '86	2
A programozás Trabantja	14
Vigyázat! Tolvaj!	15
Rajzológépek	16
SICOB — volt és lesz!	18
Adok-veszek-cserélek	19
µNFORM	27
Interaktív kompakt lemezek	39

### ISKOLA — SZÁMÍTÓGÉP

Meghalt a király, ki lesz az új király?	3
Ismerős gondok az Egyesült Államokban	4
Másolóprogram	8
PRIMO-primátus	8
Gépi kódban	9
Tv-foci	10

### PROGRAMOZÁSTECHNIKA

Z80 programozási gyakorlatok 7.	12
BASIC és gépi kód	13

### µKLUB

QLub	20
Adom a magyarázatot!	20
A Spectrum operációs rendszere ZX81-en	21

### µPROGRAMOK

A Beta BASIC és alkalmazása III.	30
Mandelbrot-program	32
VC20 programok beolvasása	34
Floppy monitor	35
Disassembler program	36
HELP bővítő	37

### SAKKPROGRAMOZÁS

Bitek és figurák	38
------------------	----

### JÁTÉKPROGRAMOK

Kétórás játékok — miért ne?	40
Labirintus	41
Kapcsoltam?	42

### AZ OLVASÓ ÍRJA

	43
--	----

<b>KÖNYVEK</b>	45
----------------	----

<b>HÍREK-ÉRDEKESSÉGEK</b>	47
---------------------------	----

# Alkalmazás '86

## A Neumann János Számítógéptudományi Társaság III. Országos Kongresszusa Szolnok, 1986. november 17—20.

*„Kiváló emberekhez valóban nem méltó, hogy rabszolgá módra órákat vesztgessenek el olyan számítások elvégzésével, amelyeket bárkire nyugodtan rá lehetne bízni, ha gépet használna.”*  
(Gottfried Wilhelm Leibniz — 1646-1716)

A kedves olvasó bizonyára furcsállja, hogy egy múlt évi eseményről négy hónap után írok, de a nyomdai átfutási idő mindenre magyarázatot ad. Amikor megpróbáltam az országos kongresszus tapasztalatait a számítástechnika iránt érdeklődő olvasók részére összefoglalni, még egy hete sincs, hogy a kongresszust befejeztük, így meglehetősen frissek az élményeim, és talán februárra sem avulnak el.

Szolnok fogadta az Alkalmazás '86 több, mint négy száz résztvevőjét, szívesen és barátságosan; a budapesti és helyi rendezők mindent megtekintettek az összejövetel sikere érdekében. Három teljes nap alatt 96 előadás hangzott el négy szekcióban. Így — mondta összefoglalójában Gergely Csaba, aki a program- és a szervezőbizottságot is képviselte — még a legszorgalmasabb résztvevő is legfeljebb az előadások egynegyedét hallgathatta végig. Én azt hiszem, hogy a rendezők jól tették, hogy szinte valamennyi előadót a pódiumra engedtek, nem csak a „sztár” előadókat, de a sokszor motyogó és nem kevésszer az előadást felolvasó kezdőket is. Amikor jó néhány évvel ezelőt a Neumann-kongresszusok megrendezését javasoltam, az egyik céloom éppen az volt, hogy a kezdők is lehetőséget kapjanak az előadói készség és gyakorlat megszerzésére. Erre itt a lehetőség megvolt, legfeljebb az idő volt kevés (pl. az oktatási szekcióban). Szerencsére jó volt a hallgatóság, nagyon sokszor vitatkozott, vagy szeretett volna vitatkozni az előadókkal, ha nem vágta volna el a szigorú elnöki szó a vita fonálát.

A kongresszus másik fontos célja — véleményem szerint — a találkozás volt. Nagyon sokan eljöttek, akiket az ember — a munkája során — csak nagyon ritkán, vagy sohasem lát. Sajnos sokan nem jöttek el, mert vagy az anyagiak hiányoztak (valta-keppen drága négy napot töltöttünk Szolnokon, hiszen 3500 Ft volt a részvételi díj, 1—2000 Ft a szállás, 4—500 Ft volt az étkezés, de — sajnos — ennyiért lehet ma egy ilyen összejövetelt megrendezni), vagy pedig a kongresszus témája nem találkozott jó néhány Neumann-tag érdeklődésével. A kongresszus záróülésén azt javasoltam, hogy ezután valamennyi kongresszust nevezzük „Alkalmazás'-nak, de a felhívásban hívjuk meg előadóként a hardver és szoftver konstruktorokat is, ne csak az alkalmazásokat, hiszen az ő munkájuk is része, sőt fontos része az alkalmazói rendszereknek. Az sem baj, ha nem négy, hanem nyolc szekció fut párhuzamosan, az a lényeg, hogy a tagságnak minél nagyobb része ott legyen ezen a háromvenkinti összejövete-

len. Ebből a szempontból a megelőző kongresszus témaválasztása még szerencsétlenebb volt (Adatbázisokon alapuló információrendszerek), mert a szakemberek egy igen szűk rétegéhez szólt, így nagyon sokan voltak távol.

A szekcióelnökök referátumából feljegyeztem néhány gondolatot, amelyeket — némi kommentárral — most közreadok.

— Elég sok volt az igazán jó és tartalmas előadás, de nem volt kevés az olyan sem, ami nagyon nagy munkával előállított szerényen eredményről adott számot. A legfeltűnőbb talán az volt — az előadásokból világosan látszott —, hogy nálunk csak a saját fejlesztésű rendszerekről illik (sikk!) beszámolni, olyasmiről nem, amikor pl. valaki egy kész rendszert (dBASE, ... calc stb.) alkalmaz egy feladat megoldására. Nem hiszem, hogy kiszámolta már valaki, hogy mennyi időt, pénzt és energiát használunk el feleslegesen, mert vagy nem ismerjük a kész és konfekcionált rendszereket, vagy pedig egyszerűen nem akarjuk használni ezeket, mert az nem elégíti ki teljes mértékben a megrendelőt vagy éppen a rendszervevőt előzetes elképzeléseit. Dömöklő Bálint mondta zárszavában, idéve egy, az IFIP kongresszuson elhangzott előadásból, hogy a jövőben, de már ma is az alkalmazás technológiája azt jelenti, hogy az alkalmazó kiválasztja az elképzeléseihez legközelebb álló rendszert, esetleg több ismert rendszerből próbál meg egy komplexet előállítani, és a saját szervezetét próbálja meg ehhez az alkalmazói rendszerhez passzítani.

Már a kongresszuson is suttozták, hogy ez a „gombhoz a kabát” technológia — lehet, de hogy a költségeket csak így lehet alacsony szinten tartani, abban teljesen biztos vagyok.

— Néhány előadás után úgy éreztem, mintha a költségek alacsony szinten tartása egyáltalán nem szerepelt volna az előadók által készített műszaki feladattervben. Kerül amibe kerül, legfeljebb magasabb lesz az üzem, a vállalat, az intézet költsége, a lényeg, hogy a rendszer menjen. Nemegyszer került az előadó meleg helyzetbe, amikor a kérdezők a megoldás költségeit feszegették.

— Viszonylag sok előadó mondta, hogy az általa bemutatott és esetenként nagyon kritikusan fogadott alkalmazói rendszer bevezetése után az nagyon gyorsan befogadták és megszerették, különösen az interaktív programokat. Néhány hónap múlva el sem tudták képzelni a munkát számítógép nélkül. Én azt hiszem, és ebben a hitemben a kongresszus megerősített, hogy a hazai alkalmazók most jutottak el annak a felismerésére, hogy vannak olyan problémák, amelyeknek a megoldására eddig azért sem kerülhetett sor, mert nem volt eszközök hozzá. Egyre több olyan példát láttunk, amikor nem eddig kézi eszközökkel végzett munkát akartak gépesíteni, hanem teljesen

új és manuálisan kezelhetetlen alkalmazó területeken próbáltak szerencsét.

— Számomra nagy és igen kellemes meglepetés volt, hogy az oktatási szekció iránt mennyien — és amint utólag kiderült, voltaképpen nem csak pedagógusok — érdeklődtek. A programbizottság egy megbícsátható tervedése volt a szekcióit a kisebbik terembe tenni, ráadásul az előtte levő szekció „túlcsordult”, így Szelezsán János szekcióelnök kénytelen volt az egyes előadások idejét 20 percre redukálni, ráadásul a vitát sem engedélyezte. (Bosszúságomban esti találkozóra hívtam a hallgatókat, akik — örömröme — el is jöttek, szinte ismét megtelt a terem, és így nagyon kellemes körülmények mellett, számomra igen tanulságos beszélgetésen hallgathattam meg a nyílt egyetem létrehozásával kapcsolatos bátorító véleményeket).

— Föltétlenül baj volt, hogy a „maszek kerekasztalhoz” hasonló esti vagy délutáni találkozók, beszélgetések nem szerveztünk. Pedig biztosan jól megtárgyalhattuk volna a már említett problémákat, de azt a sok, a szakmát izzató kérdést is, amelyek így a résztvevőkben „bennszorultak”.

— A nyitó előadásokat Vámos Tibor, Pál László, Reinger Péter és dr. Udovecz Gábor tartották, a műszaki fejlesztés, az ipar és a mezőgazdaság számítástechnikai, elektronikai terveiről és gondjairól beszéltek. Sajnos, ezeket az előadásokat sem követte vita; nagyon sokan vették úgy, hogy talán a főhatóságoknak és a résztvevőknek is hasznosabb lett volna egy ilyen beszélgetés, mint a kétségkívül kitűnő táncegyüttes műsora.

A Neumann-kongresszusok emlékezetes pillanata a Neumann-díjak és a Kalmár emlérképek kiosztása. Az idén — ez a személyes véleményem — igen jól sikerült a díjak odaítélése, az erre érdemesek kapták a kitüntetés. Óriási taps köszöntötte a társaság tíz évig aktív, ma tiszteletbeli elnöket, Vámos Tibort, amikor a díjat átvette. Hasonló ünnepélyben részesült Könyves Tóth Pál, aki a magyar számítástechnikai szak-sajtóban végzett munkájáért és Jávor András, aki az első kórházi számítógépes rendszer kidolgozásáért vehette át a Neumann-díjat. A Kalmár emlérképet főleg szakmai eredményekért ítéli meg a társaság országos elnöksége. Az idén Hanák Péter az egyetemi oktatásban, Lócs Gyula a programozásban elért eredményéért és a két Lukács testvér — megosztott díjat — az első hazai házi számítógép kifejlesztéséért részesült az egész tagság elismerését kifejező kitüntetésben.

A III. Országos Kongresszus befejeződött. Inkább a problémákról írtam, mert csak ebből tanulhatunk és rendezhetünk olyan kongresszusokat, amelyek egyre jobban összekötnek bennünket. Ezt kívánja valamennyiünknek:

KOVÁCS GYÓZÓ

# Meghalt a király, ki lesz az új király?!

Az iskolákban közismert HT-1080Z hazai gyártmányú iskolaszámítógép elérkezett életpályája végéhez. Az utolsó darabok ez év elején hagyták el a gyártó műhelyeit, s a közelmúltban a maradék tőszázat is kiürítették, leszállított áron. Eddig 2364 darab került a különböző oktatási intézményekbe, s közülük 1861 kifejezetten az iskolákba.

Korai lenne mégis a búcsúztatás: e gépekre ugyanis elég sok program készült az idők folyamán, s várhatóan egyes szervezetek és gm-k még évekig vállalni fogják a javítást. A gép egyik előnye, hogy egyszerű felépítésű, különleges alkatrészek nélkül épült. Ma már túlhaladott, de segítségével indult meg az iskolaszámítógépes program, vele szerztük az első tapasztalatokat az oktatásban, így érdemei feledhetetlenek.

A HT-1080Z a TRS-80 típusú nyugati mikrogép klónjába tartozik. A Microsoft BASIC level-two verzióit használja, így BASIC szinten — a hangrész kivételével — kompatibilis volt külföldi elődjével. Ugyanakkor a gyártás folyamán is megváltozott, így ennek a BASIC-nek, de főleg az assemblernek különböző dialektusai alakultak ki. Az első, 1983-as kiadásban a magno felvételi színtjét még kézzel kellett állítani, s nem volt ékezetes betű sem. A következő évi sorozatnál már megjelentek egyes ékezetes betűk, sőt a színtszabályozást is automatizálták. Végül, 1985-ben megjelent a 64 kb-jaos változat — még több magyar karakterrel. De ez is képtelen volt a színes, finomgrafikus képmegjelenítésre, ami a korszerűbb személyi számítógépeknél, főleg, ha az oktatásban alkalmazták őket, elengedhetetlen.

De mi legyen ezután? Ennek a fogas kérdésnek az eldöntésére az iskolaszámítógép-program koordinátora, a Tudományszervezési és Informatikai Intézet pályázatot írt ki. Ezen a megmértetésen négy gép bizonyított alkalmasnak arra, hogy a nebulók oktatását szolgálja. Mint azonban majd látni fogjuk, valószínűleg csak egy hazai és egy külföldi típus kerül be ténylegesen az oktatási intézmények falai közé.

Előjáróban azonban érdemes egy pár szót szólni az árakról. A cikkben szereplő árak nem azonosak azokkal, amelyekért az üzletben bárki (nem oktatási intézmény) megvásárolhatja ezeket a berendezéseket. Külföldön már régen bevált az a gyakorlat, hogy az oktatási intézmények igen kedvezményes áron, forgalmiadó- és vámentesen kapják a berendezéseket. Hasonló feltételeket kell itthon is biztosítani. Az árak értékeléséhez azonban azt is tudni kell, hogy amennyiben ezeket a készülékeket az iskolák a TII-n keresztül szerzik be, további egyharmaddal olcsóbban juthatnak hozzá, mert a különböző oktatási számítógép-program gazdái költségvetési forrásokból fedezik.

Az egyik jövőendő új iskolagépnek a

HT3080C látszott, de kétséges, hogy sorozatgyártása egyáltalán megindul-e. Maga a gép nem kompatibilis az öt megelőző HT iskolagép-sorozat egyik tagjával sem. Lényegében a ZX-Spectrum programjai futtathatók rajta, így természetesen finomgrafikus lehetőséggel és színes megjelenítési móddal is rendelkezik. A BME kollégiumának számítástechnikai köre fejlesztette ki. Várható kedvezményes ára 27 ezer forint lenne; sokkal több, mint a trón más várományosaié. Csak igen nagy sorozatban lehetne olcsóbb, mert akkor érdemes lenne itthon kifejleszteni a berendezésorientált áramköröket és gyártásukat.

A másik jelölt a Primo utódgépe lett volna. Maga az eredeti alapgép — a billentyűs, illetve a fóliatászatúras — gyártása megszűnt, a maradék készleteket most árusítják ki árengedménnyel. A pályázaton szerepelt utód a PRO-Primo, amelynek van színes grafikája, és egy hazai BASIC „dialektust” alkalmaz. A megajánlott kedvezményes ár szintén magas: a 32 kb-jaos változat 13 500, a 64 kb-jaos pedig 17 ezer forintba került volna. Eddig csak 850 darabra van megrendelés, de a gyártó — a COSY — véleménye szerint csak minimálisan 10–15 ezres sorozatnál tarthatják ezt az árat, így nem érdemes belefogni a sorozatgyártásba.

A hazai utód, az egyik általánosan elterjedt iskolagép szinte biztosan a Videoton TV-Computer lesz. Az idén már 1000 darabot szállítottak le belőle, a 32 kb-jaost 11 ezer, a 64 kb-jaost 13 ezer forintot kedvezményes áron. A jövő évtől csak a 64 kb-jaos gépet fogják szállítani, 12 ezer forintot áron, és ebből 5000 darab kerülhet az iskolákhoz. Egyelőre még nincs hozzá szoftver, mert a Videoton csak a hardver gyártásával kíván foglalkozni. Így még mintegy fél évet kell várni az első oktatási célú programcsomagok megjelenéséig. Az ellátást az OPI programirodája szervezi. A koordinátori munkát és a sokszorosítást ez esetben is a TII intézi. A VT TV-Computer előnye, hogy teljes rendszert lehet összeállítani belőle, az oktatási intézmények költségvetésébe beférő árakon. Jövőre megjelenik a floppy és sornyomató — ez utóbbi Centronics interfésszel. A BRG által gyártott adatmagnó már megvan, kedvezményes ára 2900 Ft. Sikeres konstrukció, olyannyira, hogy Bécsben is lehet látni a számítástechnikai áruházak némelyikében, igaz, nem ezen az áron... (drágább).

A közkedvelt Commodore sem maradtott ki az iskolagépek közül. A C64-es az ára miatt nem versenyképes, de a C16-tal kompatibilis Commodore Plus/4 magnóval együtt 7900 forintot kedvezményes áron került be, amelyre a TII-től beszerzett gépekre további egyharmad kedvezményt kapnak az iskolák. Kapható lesz a közeljövőben a Commodore 1551-es floppyval is

— adatmagnó helyett —, 18 600 forintot kedvezményes oktatási áron. A Commodore gépekre igen sok szoftver készült az iskolákban. A Novotrade Octasoft és Delta-soft irodája szintén forgalmaz programokat ehhez a típushoz, közöttük kifejezetten oktatóprogramokat is. A TII és az OPI jövőtől rövidesen várható a tananyaghoz kapcsolódó programok megjelenése.

A TII és az OPI megrendelésére készült programokat az Akadémiai Könyvtárúdió Városház utcai Magiszter Könyvesboltjában, mintegy 200 forintot egységáron, programkattán bárki megvásárolhatja. Megjegyzendő, hogy ezek a programok nem másolásvédettek. A Novotrade által forgalmazottak sem, de turbósítva vannak, így igen nehezen másolhatók.

A TII-n keresztül eddig 3400 darab C16-os számítógép került forgalomba; iskolába ebből 3004 darab. A C Plus/4-ből eddig 5300 jutott az iskolákba, és megrendelés van további 1920 floppy gépre. Mind a C Plus/4-ből, mind a C16-ból az országban beszerzési forrástól függően több verzió van forgalomban. Ezek annyiban különböznek, hogy a TII által forgalmazott gépekben a ROM magyar karakterkészletre van kicserélve, a külföldről behozott gépek pedig vagy angol, vagy/és német karakterkészlettel dolgoznak. Ez galibát okozhat néhány külföldi eredetű játékprogramnál, amely a kisbetűt is használja: egyes félgrafikus karakterek helyett ugyanis a magyar ábécé betűi jelennek meg. A magyar oktatóprogramoknak — mivel ezek természetesen magyar ékezetes betűket is alkalmaznak — a direkt külföldről behozott gépeken való futtatása okozhat gondot. Athidalja ezt a Novotrade-nál megvásárolható, a gépbe betölthető magyar karakterkészlettel generáló szoftver bár hallani arról is, hogy egyesek a problémát hardverrel — egy másik ROM beépítésével — is megoldották. A C Plus/4 64 kb-jaos, tehát eleget tesz a korszerű iskolagép követelményeinek, s azt már mondani sem kell, hogy alkalmas finomgrafikus megjelenítésre. További szempont, hogy a közben már régebben forgalomban lévő C16-osok is bővíthetők 64 kb-ja. Vagy külső bővítő hozzácsatolásával, ami a nagyobb áramfelvétel miatt tápegységmódosítást tesz szükségessé, vagy egyes gm-k ugyanezt, hardver útján is egyszerűen és viszonylag olcsón elvégzik.

Hogy ki lesz tehát az új király az iskolákban? Az idő ad erre végleges választ; még sok minden történhet. Annak idején sokan nem hitték a HT sikerében. A jelenlegi helyzet kettős királyságra utal: néhány évig valószínűleg a VT TV-Computer és a C Plus/4 lesz az uralkodó az iskolák és egyéb oktatási intézmények kabinetjeiben.

Magasabb nívón

# Ismerős gondok az Egyesült Államokban

Simonyi Endre (rovatszerkesztőnk) egyesült államokbeli útja során találkozott többek között William S. Wagner úrral, az IBM oktatási osztályának tanácsadójával — korábban a Santa Clara megye számítástechnikai vezető szakfolyóírója, a Számítógép-használó Oktatók (CUE) nevű szervezet elnökségi tagja —, aki közlés céljára rendelkezésére bocsátotta egy konferencián tett felszólalásának anyagát. Mivel az iskolaszámítógép-programunkban részt vevő tanároknak, a program irányítóinak érdemes megismerkedniük az előttünk járó országok tapasztalataival, eredményeivel, közreadjuk Wagner úr előadásának érdeklődésükre számot tartó részleteit. Örülünk, ha olvasóink reflexiókat fűznének a közöltekhez.

## Visszatekintés

Először is perspektívába helyezném a jelenet a számítógépes korszak, a számítógépes oktatás kezdeteire való visszatekintéssel. Létezett 1952-ben egy csodálatos számítógép (de hát melyik számítógép nem csodálatos a maga idejében?), az IBM 704. Ez a gép nagy teljesítményű volt, gyors volt, és lehetőségeivel elbűvölte az embereket. Nem számított, hogy bérleti díja havi 40 000 dollár volt, hogy öt mérnök állandó jelenlétét igényelte, és hogy tisztecentként leállt. Nem számított, mert a gép addig nem tapasztalt műveleti sebességgel és teljesítménnyel dolgozott.

1966-ban egy kevéske FORTRAN-t oktatam — egy hétig az osztályomban, továbbá egy este az egyetem számítóközpontjában. A produktumunk egy bináris program volt, amely kétjegyű számokat adott össze, és kiírt egy nevet. Nem érdekes, hogy csak ennyi volt, a lényeg az, hogy programoztuk egy számítógépet.

1977-ben vásároltam négy számítógépet az iskolámnak. Darabjért 4000 dollárt fizettem, de higgyék el, az iskola vezetősége mégis meg tudta elégedve a vásárlással: 25 százalékkal olcsóbban vettük ugyanis, mint egy akkor korszerűnek számító, négytermiális mikroszámítógépes rendszer ára. Ugyanakkor rendelkezésünkre állt négy gép, mindegyik egy-egy monochrom monitorral, 48 k-ra bővíthető 32 k RAM-mal, 70 k lemezkapacitással és BASIC programozási nyelvel. Mondhatná bárki most, és igaza lenne, hogy ezek a számítógépek nemigen hasonlíthatók az Apple vagy JX gépekhez; mégis, én akkor nagyon boldog voltam, mert azok a gépek ott az osztá-

lyomban lényegesen nagyobb teljesítményűek voltak, mint a 25 évvel korábbi legjobb gépek.

Tradicionalis, veterán matematikatanárként el voltam bővölve a jó szerencsém: lehetőségem nyílt arra, hogy az addigiaktól eltérő módon oktassak! De nyilván Önök közül is soknak volt már ez a tapasztalata: a diákok egy része nem mutatott érdeklődést, és otthagyták a pácban. Elmarasztaltak emiatt, érdemtelenül. A többi diákban viszont annál nagyobb volt a buzgalom. A BASIC tanítása olyan helyzetet teremtett, amelyben a diákok vagy azok kis csoportjai egyénileg haladtak, próbáltak megoldásra jutni, és módszereiket illetőleg azonnali visszajelzést kaptak a számítógéptől.

1978-ban alkalmam volt előadást tartani a Második Nyugati-parti Számítógép Vásáron, San Franciscóban. Reméltem, hogy lesznek ott olyan pedagógusok, akikkel érdemes lesz találkoznom. Meg kellett tudnom, hogy vannak-e rajtam kívül, akik bosszús és örömteli órákat töltenek a gépek mellett, új oktatási módszereket fejlesztenek. Meglepetésemre több mint kétszázan hallgatták meg egy ismeretlen főiskolai tanár első előadását... Amikor az előadásom során újabb találkozót javasoltam, kiderült, hogy nagyon is sokan vannak a hallgatók között számítógép-kedvelők. Ez végül is a Számítógép-használó Oktatók (CUE) szervezetének megalakításához vezetett. 1978-ra ugyanis a következő felismerések fogalmazódtak meg:

- nincs követhető tananyag;
- nincsenek szervezetek, konferenciák, újságok;
- nincs tanárképzés; a legtöbb egyetem rengeteg időt fecsér el a számítógépes oktatás mikéntjének feltárására; nincs számítógépes oktatásról szóló diploma és nincs is hol megszerezni;

— szoftver alig létezik, és a fellelhető legtöbbjét tanárok vagy más amatőrök írták; nincs szakirányú kutatás;

— a nagy kiadók és hasonló intézmények mellőzik ezt a tárgykört;

— az illetékesek nem foglalnak állást, hogy támogatják-e az oktatásnak ezt az irányát vagy sem;

— a hosszú távú tervezésre csak elvétve van példa: kevés iskolának vagy iskolaközetnek van elképzelése, hogy milyen irányba tartson a számítógépes oktatásban.

Minthogy valamennyi felvetett probléma megoldásra várt, gazdagon kínálkozott alkalom a világ újrateremtésére. És bár valamennyi területen értünk el eredményt, a számítógépes oktatás olyan fiatal terület, hogy itt még mindig nagyon sok a tennivaló; nekünk, tanároknak még mindig szép számban lesz módunk ebben jelentős szerepet vállalni.

A 70-es években mindenesetre úgy láttuk, hogy korunkat meghaladva előbbre járunk, teljesen magunkra hagyatva. Ez végül soron természetesen is — gondoljanak csak a tinédzserek viselkedésére. És ahogy ők általában, mi is messze jártunk jövőbeli elképzeléseinkkel a valóságtól. Íme néhány példa.

- Azt vártuk, hogy a számítógépgyártó cégek szaporodni fognak — bár az IBM belépésére akkor még nem számítottunk. Jelenleg azonban jobbra csak négy géptípus: Apple II, TRS-80 Co-Co, Commodore 64 és IBM PC jellemző az egyesült államokbeli iskolák gépparkjára, azaz kevesebb, mint 1979-ben.
- Úgy véltük, a szoftverkérdés megoldását a tanárok által fejlesztett, mindenki számára hozzáférhető programok jelentik majd. Erre a válasz a kaliforniai SOFTSWAP és a minnesotai MECC cégek. Nem láttuk előre, hogy mennyire

nem egyszerű jó minőségű szoftvertermékeket előállítani és forgalmazni.

● Nagy jövőbeli jelentőséget tulajdonítottunk a jogvédelemnek. Tanárokból álló csoportok létrejöttét vártuk, amelyek kompekt leckeért írják majd, és ezeket lemezekre fogják a diákok számára hozzáférhetővé tenni. A jogvédelem magas költsége és a termékek gyenge minősége azonban nem tette vonzóvá ezt az eljárást.

● Két dolognak a megvalósulását sokkal korábban jósoltuk. Az egyik a lemezkölcsonzés. Egyre népszerűbb és egyre választékosabb ugyan, a technikai problémák és a szoftvervédelem miatt azonban lelassult az intézményesítése. A másik a részletes magyarázattal ellátott adatközlő programok. Ez a rendszer a későbbiekben kényelmetlennek bizonyult: az útmutatókat nem vették figyelembe, ez a vonal is befulladás. Most a nagyobb teljesítményű gépektől, a kevésbé költséges hálózatoktól várjuk az információhordozó programok újraéledését.

● Megemlítem, hogy az utópisták a 70-es évek végén vezető helyre sorolták a videorendszereket, a video-lemezjátsszókat. Számottevő darabszámban egyik sem áll még rendelkezésünkre, de el kell ismernünk, közeleg az áttörés ezen a területen.

● Két dologra egyáltalán nem számítottunk. Az egyik, hogy valamennyi iskolai évfolyamban általános lesz a szövegszerkesztők használata. De nem volt előre látható a többi alkalmazási szoftvernek tanterv szerinti használata sem. A Logo nem várt hatása a kicsikre és a tanárookra a másik erőmteli meglepetés.

Sokban tévedtünk tehát. Viszont a számítógépes oktatás forradalma már nem látomás a távoli messzeségbe, hanem itt és most zajlik. Búszkák lehetünk arra, amit az oktatók a számítógépekkel megvalósítottak. Nagy előrelépést tükröz, hogy

— erős nemzetközi és helyi szervezettek jöttek létre szerte a világban. Kiadványaik, konferenciáik kiváló fórumai az elképzelések terjesztésének;

— az egyetemeken megindult a tanárképzés kezdő és haladó fokon egyaránt. A helyi központok meghatározó forrásai a képzésnek, a tananyagra vonatkozó speciális információknak, szoftvereknek stb.;

— az oktatóprogramok egyre jobbak, nagy kiadók is bekapcsolódtak a forgalmazásukba;

— országos és helyi szoftverismertetőket nyújtanak segítséget a tanároknak a választáshoz az óriási kínálatból;

— mind egyetemi, mind helyi viszonylatban jelentős fejlődésnek indult a kutatás.

### A jövőre utaló észrevételek

Elsőként a kompakt lézerlemezre gondoltok. Jelenleg 1000 dollárba kerül az IBM PC-vel kompatibilis lemez meghajtó és hozzá egy 21 kötetes lexikon kompakt lemezen. A lexikon mindössze egyötödét foglalja a lemeznek! A lemezt kezelő szoftver igen hatékony, a gyorsaság elképesztő. És mindez olyan áron, hogy felsőoktatási intézményeink könyvtárába már ma is könnyen telepíthetünk egy-egy ilyen egységet. De tekintsük a kompakt lemezt most egy szoftvertároló médiumnak. Képzelnék el 1400 darab floppylemez egy kupacban: 4 méter magas oszlopot tenne ki. Tartalmuk ugyanakkor ráfér egyetlen kompakt lemezerre. Gondolják csak meg: az iskolájuk vagy iskolakörzetük egyetlen 5"-es lemezen kaphatná meg az összes szoftvert, amelyről az tán könnyen kiválogathatók lennének a sohasem használt és a nélkülözhetetlennek ítélt programok. (A kompakt lemezekkel egyébként részletesebben foglalkozik dr. Simonyi Endre cikke lapunk mostani számában, a 39. oldalon. — A szerk.)

És most a videóról. Az ideai vancouveri EXPO-n új információközlő állomásokat mutatnak be. Ezek valójában egy IBM PC-ből és egy új kereskedelmi termékből, az ún. IBM Infowindow Systemből álltak. Ezek az egységek feladatokat állítottak elő és küldtek a képernyőre. A feladatok, gyakorlatok számítógépes grafikából, fényképből, színtetizált vagy élő beszédből és mozgó videóból álltak. Izgatottan várom, hogy milyen lehetőségeket hoz ez az új rendszer a minőségi információközlés területén.

A jó minőségű hardver és szoftver azonban nem minden. A legfontosabb amit minden tanárnak személy szerint, egyedül kell véghezvinnie: keresni kell a megoldást, kipróbálni, újat és igényesebbet alkotni, egyszerűvel dolgozni azon, hogy miként lehet ennek a technikának az előnyeit valóban kihasználni. Hogy olyan iskolákban tanulhassanak az elkövetkező nemzedékek, amelyekben az eddigieknél jobban lehet tanulni és tanítani. A számítógépes oktatást favorizáló tanárok feszített tempóban igyekeznek is minél többet megtanulni a számítástechnikából. Újabb és újabb gépalkalmazásokat találnak ki az ismeretközlés javítására. A jövő iskolája valójában itt van már körülöttünk — részleteiben „fel van találva”.

### Néhány kísérletről

Az alaszka Juneau-ban Paul Berg egy éves időtartamú programot szervezett számítógépet nem használó olyan tanárok részére, akikben megvan az akarat, hogy az írást-olvasást a megszokottól eltérően oktasák. A hallgatók fejlődése a kezdők újító felélmétől a megegyező fejlődés felé halad magabiztoságáig hű tükrre legutóbbi történetének. Paul Berg erről szóló jelentésének címe: Hand-in-hand: The Writing Process and the Microcomputer (Kéz a kézben: Az írás folyamata és a mikroszámítógép).

Tabiona kis falu Utah állam kanyonregtetegében, az isten háta mögött. Az ottani diákok korábban nem kaptak idegen nyelvi oktatást, csak miután az egyetemek megemelték a felvételi szintet. Ekkor a polgármester a kérdés megoldására tervet dolgoztatott ki egy munkacsoporttal, amelyben az oktatási osztály, az IBM, egy videotechnikával foglalkozó társaság és az idegen nyelvi oktatás egy szakértője dolgozott. Jelenleg naponta sugározják a tananyagot műholdon keresztül, és rendelkezésre áll egy beszédszintetizátorral ellátott IBM PC, hogy segítse az anyag jobb elsajátítását. Az eredmény: a legtöbb diák megduplázta a tempót a spanyol nyelv tanulásában.

Marilyn Pollock ötödikeseket tanít egy Chicago melletti kisvárosban. 15 számítógép működik az osztályában. Ő a kedvenc példám a jó tradicionális tanárúpusra. A rendelkezésre álló valamennyi erőforrást felhasználja, beleértve a férjét is, hogy ismereteket szerezzen az új technikáról, és hogy a technikát az oktatás céljainak szolgálatába állítsa. Marilyn osztálya felfedezi a tudományokat, társadalmi ismereteket szerez adatközlő programok segítségével megtanul lemezer írni, és kapcsolatban tart távoli helyekkel a telekommunikáció segítségével. De a legfontosabb, hogy a gyerekek hozzásszokjanak, miként lehet és hogyan kell tanulni számítógépekben gazdag környezetben.

New Mexico egyik iskolájában megtalálták a módját, hogyan tehettek szert olcsón új számítógépekre: az iskola elektronikai osztálya építi a gépeket, majd átadja a programozással foglalkozó osztályoknak.

Oaklandban (Kalifornia) a Franklin bentlakásos iskolában az öt-hat évesek, akik több mint húsz nyelvet beszélő közösségek közül kerültek ki, négy számítógépet használnak az olvasóteremben. A számítógépes gyakorlás kiegészül kettőtára listázással, beolvasással, rajzolással, gépeléssel, levelek kinyomatásával. Ezek a gyerekek is az IBM Writing to Read programot gyá-

korolják, amellyel már jelentős sikereket érték el az ország több mint 2000 iskolájában, az írás-olvasás oktatásában. Számukra a számítógéppel semmi misztikus nincs: része a szoba berendezésének.

Edward Reese atya a San Jose-i (Kalifornia) katolikus iskola igazgatója. Amikor az egyik szülő felajánlotta, hogy kiépít egy terminálos rendszert az egyik osztályban, Reese atyának máris kész terve volt a gépek üzemeltetéséhez. Az egyes munkahelyeken alkalmazói szoftvereket futtatnak. Törölte a programozást a 9. osztályosok tantervéből, és szövegszerkesztők, táblázók (spread sheet), adatbázis-kezelők oktatását vette fel. Elsőként sajátította el a programok kezelését, majd az egyik szekciót oktatta.

Denver belvárosában van a George Washington iskola. A város iskolái közül ez a legnépesebb, és neki van országos viszonylatban a legkiterjedtebb számítógépes oktatási programja. Ezt a programot egy matematikatanár dolgozta ki. 750 diáknak 27 szekcióban 15 tantárgyat oktatnak. Túl az ügyvitel, a matematika, a nyolcféle programozási nyelv oktatásán, van egy olyan kurzusuk, ahol diák, tanár egyaránt számítógépet szerez el és szoftvert készít a többi osztály számára. Egy diákcsoport a közelmúltban olyan programot fejlesztett ki, amely az angol nyelvet oktatja második nyelvként, és amely a laoszi ábécét is megjeleníti. Ez a szoftver a kereskedelembe is kapható.

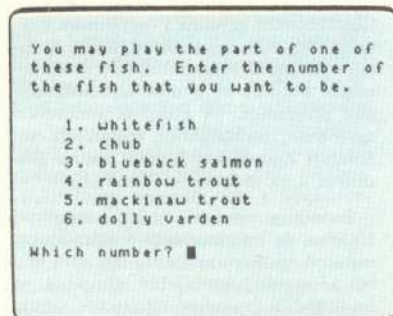
Az itt következő, utolsó példával a mikrogépek nagy teljesítőképességére kívánok rámutatni, amely önmagában is új lehetőségeket kínál az oktatásügy minden területén.

Kaliforniában van egy kies, mezőgazdasággal foglalkozó település, mindössze egy iskolával. Az iskolának rendkívül érdekes a tanmenete. A napi tananyagot az osztálytermekben felállított tévékészülékek sugározzák. A programozással foglalkozó osztályokon kívül két különleges osztály. Az egyik a számtanfeladatok megoldásában nyújt segítséget a Multiplan programmal, a másik szövegszerkesztési feladatokat lát el. Rám a legnagyobb benyomást az angol nyelvi csoport tette, amely az elsősök nyelvi problémáit hivatott megoldani. A nehezen boldoguló diákok két sorozatból álló nyelvi kurzuson vesznek részt. Ezen a tanfolyamon a tananyagot hét kategóriába sorolták: olvasás, írás, helyesírás, nyelvtan, beszélgetés, előadások tartása, előadások hallgatása.

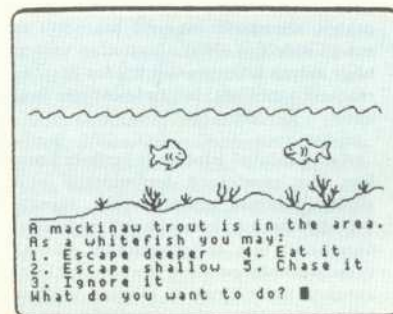
Ha szétnézünk magunk körül, láthatjuk, hogy az Államok legeludogottabb részein, a legszegényebb rétegekből származó gyerekek is szövegszerkesztőket használnak gon-

dolataik megfogalmazására. Aztán rádöbbenünk arra is, hogy azok az IBM PC-j-ek, amelyeket ezek a gyerekek használnak, legalább olyan teljesítményűek, mint egy nagyszámítógép volt a hetvenes évek elején. Felemelő gondolat, hogy a számítógépes oktatás hívei, a tanárok, mint Önök vagy én, a magas szintű technikát a legjobb cél szolgálatába állították. Olyan oktatási segédeszközt hoztak létre, amit egy mérnök sem tud kigondolni.

Mi, akik igyekszünk, de kénytelenek is vagyunk szakmailag továbbfejlődni, álmodozunk és alkotunk. Olyanok vagyunk, mint egy angoltanár egy kis kaliforniai faluban. Vannak diákok, akiket segítünk a megfelelő szintre feljutni, másoknak feltárjuk a lehetőségeket. És tudnunk kell, hogy a technika ki nem aknázott lehetőségei bennünk is rejlenek. Van-e ennél izgalmasabb?



1. kép



3. kép

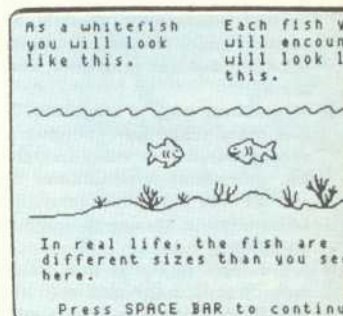
## Utószó

Lássuk, hogy körülbelül milyen is az a szoftver, amely kimondottan oktatási célra kerül forgalomba az USA-ban. Amint Wagner úr anyagából megtudhattuk, két nagy cég, a minnesotai MECC és a kaliforniai SOFTSWAP ezeknek a szoftvereknek az el-

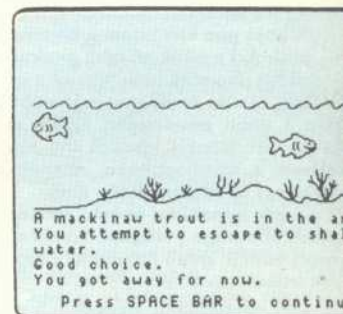
sődleges előállítója. A MECC cég (Minnesota Educational Computing Consortium) nemzetközileg elismerten a vezető helyre tehető az oktatóprogramok gyártói között. 12 éves múltra tekint vissza; az akkori kétfős stáb mára 128 alkalmazottal működő vállalattá fejlődött. 1985-ben választékuk meghaladta a 300-at, és jóval több mint egymillió példányzámban volt forgalomban szerte a világban. A programok mintegy 60 százalékát Apple gépre írják, mint-hogy az egyesült államokbeli iskolákban ez a legelterjedtebb géptípus.

Ennek a cégnek a kínálatából mutatunk be egy programot, melyet Simonyi Endre a gyártótól az ismertetés engedélyével kaptott.

A „Természet biológiája” tulajdonképpen két programból áll: az *Odell tó* a 3–9. osztályosoknak, az *Odell erdő* a 2–8. osztályosoknak készült. Az *Odell tó* programot



2. kép



4. kép

az Oregon állambeli, azonos nevű tó adatai alapján állították össze. A programnak egyszerű, de vonzó grafikája van. Az *Odell erdő*, amely a tavat körülölelő erdő állatvilágának életét szimulálja, grafikával nem rendelkezik.

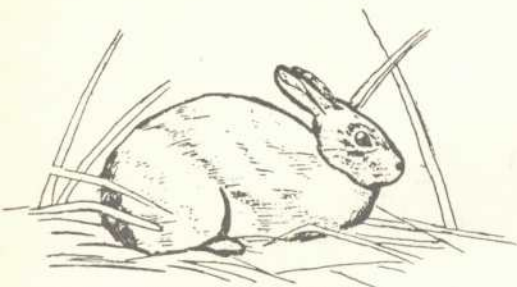
A „Természet biológiája” a tápláléklánc ökológiai összefüggését tanítja. A diák ki-



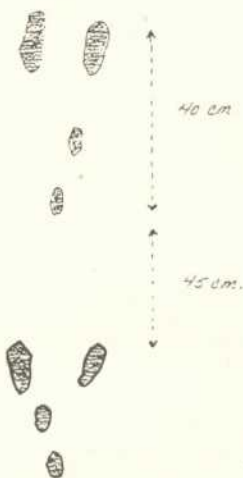
10c

## ODELL WOODS - THE ANIMALS

## RABBIT



Snowshoe Rabbit



ram egy futásban, és a diáknak a túlélésre kell játszania.

Ez egyszerűen hangzik, de nem mindig annyira magától értetődő a megoldás, mint az említett esetben. A növények, növényevők és hűsevők kapcsolata bizonyos esetekben sokkal összetettebb, mint ahogy várjuk — kivált a tó táplálékláncában (2., 3. és 4. kép). És a program még csak nem is tartalmazza a tápláléklánc valamennyi elemét!

Mindenesetre az embernek az a benyomása, hogy a program a tudásanyag észrevétlen fokozását szolgálja. Nem azt kívánja a gyerekekbe sulykolni, hogy nagyon járatlanok még a témakörben, de mégis sokszor le kell futtatniuk a programot ahhoz, hogy egy kis sikerélményben legyen részük. Sok magától értetődő dologhoz hozzátesz egy kevés nem kézenfekvőt, így a memorizálás nem követel különösebb szellemi erőfeszítést.

Mindkét programban beépített rutin szabályozza a diák által választott állattal szemben állók reakcióját. Például ha egy farkas találkozik egy farmerrel, a farmer rálő az állatra. A farmer találatának aránya, ha a farkas nem ugrik a farmernek, hanem futásnak ered: 75%-ban téveszt, 25%-ban megesbesíti, 15%-ban megöli az állatot. Egy másik példában: kevesebb esélye van egy rókának a medve elől elszaladni, ha rossz erőnléti állapotban van. Így tehát hibázik, ha a nyúl elől elszalad (bár a találkozást így is túléli), vagy ha elkerüli a figyelmét, azaz ha nem eszi meg.

A „Természet biológiájá”-t kitűnő kézikönyvvel látták el. Olvashatósága a megjelölt korosztályhoz igazodik. A 47 oldalas kézikönyvből húsz oldal a témával foglalkozó tananyag: leírja a tó kialakulását, ismerteti a terület állatvilágát. Az ábrák nem csak az állat képét, hanem lábnyomát, sőt annak méreteit és járás közben kialakuló formáját is tartalmazzák (5. kép). A kézikönyv többi része a programot ismerteti, és útmutatást ad a tanárnak a tárggyal foglalkozó biológiaóra megtartásához. Szép számban található a könyvben személyi adatokkal együtt kitöltendő tesztlapok is.

VÁGH ISTVÁN

*Szerkesztői megjegyzés.* A cikk nyomdai előkészítése közben kaptuk a hírt, hogy a Commodore cég szerződést kötött a MECC céggel az Odell Lake (Odell tó) elnevezésű programnak a Commodore 64C géppel együtt szállított programként történő forgalmazására. Ezenkívül a géppel együtt szállítják még az ugyanebben a lapszámunkban ismertetett Geost is, valamint egy távadatátvitelt vezérlő programot.

MECC

5. kép

választja, hogy milyen állat szerepét kívánja játszani (1. kép). Ha például vadnyúl lesz, és az erdőben más állatokkal találkozik, mondjuk egy éhes farkascordával, vá-

lasztania kell, hogy elkergeti-e, megeszi-öket, rájuk se hederít, vagy esetleg elszalad. Jelen esetben a futás a legjobb választás. Kilencféle találkozásra ad alkalmat a prog-

## PRIMO

### Másolóprogram

Az alábbi, saját készítésű programommal könnyen lehet BASIC és gépi kódú programokat másolni.

Betöltés után a program kiírja parancsait, amelyeket a kezdőbetűkkel választhatunk ki. A másolandó program betöltése után kiírja az indítási címet, amennyiben automatikusan induló gépi kódú program volt az. Bármelyik gomb érintésével visszatérhetünk a menühöz.

A másolóprogramot a BASIC programmal lehet felírni magnóra. Ha ez megvan, LOAD paranccsal tölthetjük be a másolóprogramot, amely automatikusan indul.

A program az eredeti LOAD és SAVE utasításokat átírja, ezért ha már nincs szükség a programra, a tápfeszültség ki-be kapcsolásával hidegindítás kell végezni.

JUHÁSZ ISTVÁN

```

0 **** Program: Juhasz Istvan ****
1 **** Print "HÖZÖL 06.10. ****
2 **** Nau "COPY betöltő ****
3 *****
10 CLS
20 PRINT#6,10,"Egy kis turelmet!"
30 FOR I=229967023517:READA:POKEI,A:NEXT
35 CLS
40 PRINT#2,5,"A masnot allisd felvetelre!"
42 PRINT#4,5,"Ha kesz inditsd a masnot,"
44 PRINT "      majd nyomj meg egy billentyu
t!"
46 IF INKEY#="" THEN#46ELSEPRINT#9,17,CHR#(2)
"OK"CHR#(18):PRINTCALL(23414)
49 DATA 0,0,0
50 DATA 33,15,69,205,117,43,205,37,0,254,76
51 DATA 40,26,254,108,40,22,254,83,202,143
52 DATA 63,254,115,202,143,68,254,66,202,25
53 DATA 26,254,98,202,25,26,24,223,33,124
54 DATA 69,205,117,43,33,92,64,54,131,33,1
55 DATA 0,205,143,0,205,117,60,33
60 DATA 140,69,119,87,205,158,0,35,119,205
61 DATA 158,0,35,119,205,158,0,35,119,205
62 DATA 158,0,35,119,71,205,158,0,35,119,16
63 DATA 249,205,158,0,35,119,205,117,60,35
64 DATA 254,177,40,6,254,185,40,2,24,207
65 DATA 119,205,158,0,35,119,205,158
70 DATA 0,35,119,205,158,0,35,119,205,158,0
71 DATA 35,119,43,43,43,126,254,185,194
72 DATA 238,67,35,35,94,35,86,235,205,175
73 DATA 15,205,102,56,254,0,40,249,195,238
74 DATA 67,33,92,64,54,131,33,98,64,34,13
75 DATA 69,33,126,69,54,16,205,146,0
80 DATA 33,140,69,126,254,177,40,47,254,185
81 DATA 40,57,205,221,58,126,205,244,58,35
82 DATA 126,205,244,58,35,126,205,244,58,35
83 DATA 126,205,244,58,35,70,126,205,244,58
84 DATA 35,126,205,244,58,35,16,249,126,205
85 DATA 244,58,35,24,204,50,92
90 DATA 64,35,126,50,94,64,205,155,0,195
91 DATA 238,67,245,205,221,58,241,50,92,64
92 DATA 205,244,58,35,126,50,94,64,205,244
93 DATA 58,35,126,205,244,58,35,126,205,244
94 DATA 58,35,126,205,244,58,195,238,67,0,0
95 DATA 0,12,2,32,32,32,32,42,42,42,13
100 DATA 42,42,42,42,42,42,42,42,42,42,13
101 DATA 32,32,32,32,42,32,32,66,89,46,32
102 DATA 74,46,73,46,32,42,13,32,32,42,42
103 DATA 42,32,32,67,32,79,32,80,32,89,32
104 DATA 32,42,42,13,13,13,13,9,9,32,4
105 DATA 76,20,32,79,32,65,32,68,13,13,9,9
106 DATA 32
110 DATA 4,83,20,32,65,32,86,32,69,13,13,9
111 DATA 9,32,4,66,20,32,65,32,83,32,73,32
112 DATA 67,13,13,9,9,32,0,73,78,68,73,84
113 DATA 93,83,73,32,67,73,77,32,58,32,0
120 DATA 33,212,89,17,234,67,1,166,1,237
121 DATA 176,33,92,64,54,131,33,205,91,34
122 DATA 203,91,33,202,91,54,16,205,146,0
123 DATA 33,92,64,54,249,17,238,67,33,146
124 DATA 69,1,0,0,205,149,0,22,0,205,221,58
125 DATA 62,185,205,244,58,58,94,64,60,205
130 DATA 244,58,62,238,205,244,58,62,67,205
131 DATA 244,58,62,205,244,58,33,25,26
132 DATA 229,201,0,0,0,67,79,80,89,32,32,32
133 DATA 32,32,32,32,32,32,32,32,32,0
    
```

## PRIMO — primátus

Keveset írnak a Magazinban a Primóról! (Az októberi szuperakció után remélhetőleg többet...) Engedjenek meg néhány elfoglalt megjegyzést a gépről.

A SUPER BITLET PRIMO-kinjában olvastam: "... a Primo rövidítve is elfogad BASIC-szavakat, ha megfelelően rövidítjük". Ezt a feltételt paraszti fordításomban kódozóként értelmeztem. A szoftverfüzet tartalmazza a kulcsszó-készlet kódjait, amelyeket a 128-225, ún. definiálható karakterek grafikus jelével hívhatunk elő. Mint tudjuk, a i vele egyidejűleg nyomott billentyű kódját 128-cal növeli, a CTRL pedig 64-gyel csökkenti. Alkalmazásánál a háromkezdű emberek előnyben vannak, ugyanis többször kell egyidejűleg 4 billentyűt is nyomni. Sőt a '84.2 verzió egyenesen zsonglőrködésre készített, ugyanis a kódmódosító billentyűk a klaviatúra két áttelenes pontján található. A megjelenített szavakból az is kiderül, hogy ez a kis gép megfelel a szélesebb periferiák követelményeinek, a szoftveres továbbfejlesztésnek. A ROM például tartalmazza a DEF, FN, MERGE, CMD, CVI, CVS, CVD szavakat.

Néhány gyakran használt parancs, utasítás, függvény képzése a következő:

ABS	Y	
AND	R	
ASC	v	
		ATN   d
		AUTO   7
		BEEP   CTR Ö
		CALL   A
		CDBL   q
		CHRS   w
		CINT   o
		CLEAR   8
		CLOSE   &
		CLS   CTR D
		CONT   3
		COS   a
		CREATE   +
		CSGN   p
		DATA   ←
		DEFINT   →
		DEFSGN   CTR Z
		DEFSTR   CTR X
		DELETE   6
		DIM   CTR J
		EDIT   CTR Á
		ELSE   CTR U
		END   CTR É
		ERL   B
		ERR   C
		ERROR   i
		EXP   é
		FIX   r
		FOR   break
		FRE   Z
		GOSUB   CTR Q
		GOTO   return
		IF   CTR O
		INKEYS   I
		INP   0
		INPUT   CTR I
		INT   X
		LEN   s
		LEFTS   l
		LIST   4
		LOAD   '
		LOG   ú
		MIDS   z
		NEW   ;

### I. lista

```

10 CLS:PRINT#(5)"R J Z O L O"
20 REM
30 REM
40 REM
50 REM
60 REM
70 REM
80 REM
90 REM SZABALYOK SOKAI
100 FOR I=0 TO 1000:NEXT I
110 PRINT:PRINT"BARHELV BILLYENTVU INDIT"
120 IF INKEY#="" THEN 120
130 CLS:PRINT"KEZDO KOORDINATAK:"
140 INPUT X,Y
150 IF X<0 OR X>255 OR Y<0 OR Y>191 THEN PRINT"HIBAI" GOTO 130
160 CLS:SET(X,Y):S=1
165 #S=INKEYS
170 IF #S="Z" AND X<255 THEN X=X+1
180 IF #S="R" AND Y<0 THEN Y=Y-1
190 IF #S="L" AND Y<0 THEN Y=Y-1
200 IF #S="F" AND Y<191 THEN Y=Y+1
210 IF #S="T" THEN 130
220 IF #S="V" AND S=1 THEN S=2
230 IF #S="V" AND S=2 THEN S=1
240 IF #S="S" THEN STOP
250 IF S=1 THEN SET(X,Y)
260 IF S=2 THEN RESET(X,Y)
270 GOTO 165
    
```

NEXT | CTR G  
 NOT | K  
 ON | N  
 OPEN | "  
 OR | S  
 OUT | space  
 PEEK | e  
 POINT | F  
 POKE | l  
 RANDOM | CTR F  
 READ | CTR K  
 RESET | CTR B  
 RESTORE | CTR P  
 RESUME | ↑  
 RETURN | CTR R  
 RIGHTS | y  
 RND | Ű  
 RUN | CTR N  
 SAVE | -  
 SCREEN | .  
 SET | CTR C  
 SGN | W  
 SIN | b  
 SQR | A  
 STEP | L  
 STOP | CTR T  
 STRS | t  
 STRINGS | D  
 TAB | <  
 TAN | c  
 TEST | )  
 THEN | =  
 TO | J  
 TROFF | CTR W  
 TRON | CTR V  
 USING | ?  
 VAL | u  
 VARPTR | É

A dokumentációellátás igen szegényes. Az ún. felhasználói kézikönyv kritikán aluli: hiányos, megjelenése előtt is elavult, szakmailag hibás. A hardvert és a szoftvert ismerető füzetek adnak ugyan komolyabb betekintést a Primo lelkivilágába, de az alapos ismerethez sok próbálkozás kell. Érvényesül a Murphy-törvény átírata: „Legjobban a káromkodás nyelvét ismerjük"! Jó tanácsom tehát a szalonképességért és az időrablások ellen küzdő gyakorlatlan primósoknak: mielőtt kikezdesz a Primoval, lépj házasságra a HT-vel!

Két kezdő színvonalú programjavaslatom van. Az egyik a „Rajzoló". A gépkönyv végén közölt ugyanilyen című program az ott leírt módon nem működik. Az 1. lista a négyírányú rajzoló javított változata, tehát nem új. A másik az Élet és tudomány 1985-ös zsebskólájának csirkevadászát c. feladványa, Primóra átírva (2. lista). Ez sem új. A listák C64-en készültek, „aki" erősen tőri a magyart.

Végül két észrevétel. 1. A Primo érti a gépi kódhívás USR-jét. Ezt semmilyen dokumentációban nem találtam, a ROM-ban sem. 2. Az egyik B-32-es gép hibás kazettatöltés utáni programindításkor „S Error" hibajelzést ad, és csak a RESET-re reagál. Nincs rá magyarázatom.

TAKÁCS DEZSŐ

Hónapokig tartó tanulásal lerövidíthető a gépelési idő. A programokon belüli karakterdefiniálás az előbbi szavak képzési módját nem befolyásolja.

## PRIMO

## Gépi kódban

## Értékkadás

A G1 jelű BASIC programban (1. lista) lévő gépi kódú rész a HL jelű regiszterben elhelyezi a négy értéket: LD (HL),04, és visszatér a BASIC programba: RET.

Az 1. sorban a program neve van.

A 100-as sorban a programban lévő skálár változóknak adunk értéket, mert ha a gépi kódú program helyének meghatározása után és működése előtt „új" skálár változót használunk, akkor a program hibásan működik.

A 110-es sorban K% (index) részére helyet foglalunk. E helyre fogjuk elhelyezni a gépi kódú programot. A foglalt hely természetesen lehet nagyobb a szükségesnél, de kisebb nem.

A 120-as sorban megkeressük, hogy a gépi kódú program részére lefoglalt hely hol van.

A 129-es sor csak magyarázó rész. Elhagyható.

A 130-as sorban betöltjük a gépi kódú programot.

A 140-es sorban meghívjuk a gépi kódú programot. A gép a HL jelű regiszter címét a C változóban adja meg.

A 150-es sorban kihatjuk ezt a címet és az ott tárolt értéket.

A program működését kipróbálhatjuk 0 és 255 között lévő egész számokkal. A programot kipróbálás előtt rögzítsük szalagra, mert hiba esetén törölheti magát.

A tanulás kedvéért rontsuk el a programot a következő módon:

- A HL regiszterbe tört számot töltünk.
- A HL regiszterbe negatív számot töltünk.
- A HL regiszterbe 255-nél nagyobb számot töltünk.
- A 120-as és a 140-es számú sorok között új skálár változóknak adunk értéket. Például így:

```
135 FOR I = 1 TO 100:NEXT I
```

## Adatátadás

A G2 jelű BASIC programban (2. lista) lévő gépi kódú rész adatot kap a DE regiszteren keresztül, át helyezi az A regiszterbe: LDA, (DE), majd át helyezi a HL regiszterbe: LD(HL), A és visszatér a BASIC programba: RET. A gép sajátossága, hogy az adatot a DE regiszter fogadja, és a HL regiszter adja vissza.

A 110-es sorban helyet foglalunk az adat számára is.

A 122-es sorban megkeressük, hogy az adat részére lefoglalt hely hol van.

A 132-es sorban a helyére rakjuk az adatot.

A 140-es sorban úgy hívjuk meg a gépi kódú részt, hogy egyben átadjuk az adatot is.

Az 1., 100., 105., 120., 129., 130. és a 150-es sorok a G1 jelű program után nem igényelnek magyarázatot.

Ezt a programot is próbáljuk meg elrontani a G1 programnál leírt módszerekkel!

SOMOGYI GYÖRGY

## 2. lista

```
5 CLS PRINT "CSIRKEVADASZT"
6 PRINT "A SAS SPACE-SZEL VADRASZIK"
7 PRINT "KILEPES "BPK"-VAL"
10 S=1:Z=0:G=1
11 INPUT "A SAS BEJESSEDE(1-10)";V
12 IF V<1 OR V>10 THEN 11
13 CLS U=(1+V)/2
15 FOR I=0 TO 255:SET(I,0):NEXT
95 C=20:RND(220)
98 FOR I=C-2 TO C+2:SET(I,2):NEXT
90 M=50:RND(100)
100 V=V1:R
105 FOR R=1 TO 255
110 IF U<255 THEN CLS:OOT0 20
115 RESET(U/R-1),M):RESET(U/R-1),M+1)
120 SET(U/R,M):SET(U/R,M+1)
130 IF I=KEY$="" THEN 170
135 FOR D=0 TO 10:NEXT D
145 NEXT R
150 FOR J=0 TO 1000:NEXT
155 CLS
160 GOTO 20
170 FOR I=1 TO 300
175 SWR(R-1)
180 IF X<255 THEN 240
185 RESET(X-U,M):RESET(X-U,M+1)
190 RESET(X-U,Y):SET(X-U,Y+1)
195 SET(X,Y)
200 Y2=2*Y1+Y-G
210 V=V1:V1=V2
215 IF V<4 AND ABS(C-X)<=3 THEN 260
220 IF V<1 THEN 240
230 NEXT I
240 PRINT "HEM JOI";S;"LECSAPASBOL";Z;"TRALAT"
245 S=S+1
250 GOTO 150
260 Z=Z+1
265 PRINT "JOI";S;"LECSAPASBOL";Z;"TRALAT"
265 S=S+1
270 RESET(C-1,2):RESET(C+1,2)
280 GOTO 150
```

## 1. lista

```
1 REM G1
100 C=0:CD=0
110 DIM X(20)
120 CD=VARPTR(ABS(0))
129 REM (LD(HL),04):RET
130 POFEC(54,4,201)
140 C=CALL(CD)
150 PRINT C,PEEK(C)
```

```
.....
A PROGRAM - 0 KOD
LD(HL),04 - 54,4
RET
```

## 2. lista

```
1 REM G2
100 C=0:CD=0:CD=0
105 INPUT V
110 DIM X(20),L(1,20)
120 CD=VARPTR(ABS(0))
122 CD=VARPTR(LX(0))
129 REM (LDA,(DE):(LD(HL),A):RET
130 POFEC(20,119,201)
132 POFEC(0,V)
140 C=CALL(CD,CD)
150 PRINT C,PEEK(C)
```

```
.....
A PROGRAM - 0 KOD
LDA,(DE) - 20
LD(HL),A - 119
RET - 201
```



# Tv-foci

A program abban különbözik a korábbi változatoktól, hogy mivel a labda sebessége 1 és 5 között szabályozható, többféle kategóriában teszi lehetővé a játékot. A bal kapust az A és Y billentyűkkel, a jobb kapust az ó és ú billentyűkkel lehet irányítani.

```
1 CLS
5 K=3
10 PRINTS
15,CHR$(3)CHR$(2);"***PRIMO-
foci***"; CHR$(18)
20 PRINTS
30,STRINGS(41,45):PRINT$ 6,2,
"BAL KAPUS":PRINT$6,28,"JOB
B KAPUS"
30 PRINTS 7,3,"FEL:A(a)":PRINT$8,3,
"LE:Y(y)"
40 PRINTS 7,29,"FEL:ó":PRINTS 8,29,
"LE:ú"
50 PRINTS 14,8,"NYOMJ MEG EGY
GOMBOT!!!"
60 IF INKEY$="" THEN 60
65 GOTO 500
70 CLS:PRINT CHR$(1):FOR I= 3 TO
252: SET(I,1):SET(I,191):NEXT I
80 FOR I= 1 TO 55:SET(3,I):SET(252,I):
NEXT I
90 FOR I= 143 TO 191:SET(3,I):SET
(252,I):NEXT I
100 U=0
110 S=4:W=10
120 N=1
130 PRINT CHR$(6)
140 H=2
150 RANDOM
160 U=U+H
170 C=C+H
180 PRINTS S,40,CHR$(138)
190 PRINTS W,0,CHR$(138)
200 PRINTS C1,U1," "
210 PRINTS C,U,"o"
220 U1=U
230 C1=C
235 FOR I=0 TO M:NEXT I
240 IF (C=W AND U=0) OR (C=S
AND U=40) THEN H=-H:GOTO
160
250 IF U=>40 OR U<=0 THEN
H=-H
```

```
260 IF C=>14 OR C<=0 THEN
N=-N
270 F$=INKEY$
275 IF F$="A" AND NOT(W=4) THEN
PRINTS W,0," ":W=W-1
280 IF F$="a" AND NOT(W=4) THEN
PRINTS W,0," ":W=W-1
290 IF F$="Y" AND NOT(W=10)
THEN PRINTS W,0," ":W=W+1
295 IF F$="y" AND NOT(W=10)
THEN PRINTS W,0," ":W=W+1
300 IF F$="ó" AND NOT(S=4) THEN
PRINTS S,40," ":S=S-1
310 IF F$="ú" AND NOT(S=10) THEN
PRINTS S,40," ":S=S+1
320 IF C>=4 AND C<=10 AND (U=0
OR U=40) THEN PRINTS C,U,
"";:GOTO 340
330 GOTO 160
340 IF U>20 THEN E=E+1 ELSE
R=R+1
350 PRINTS 0,0,E;"";R:CHR$(18)
360 IF E=10 OR R=10 THEN 410
370 IF INKEY$="" THEN 370 ELSE
PRINTS 1,13,"15 üres";
380 C=FIX(RND(0)+13+1):IF
RND(0)>0.25 THEN H=2*SGN(H)
390 GOTO 160
400 CLS:PRINTS 8,5,"REMÉLEM JÓL
SZÓRAKOZTAK": GOTO 470
410 FOR I=0 TO 1000:NEXT I
420 CLS:PRINT"JÁTSZANAK MÉG?(I/
N)"
430 F$=INKEY$:IF F$="" THEN 430
440 IF F$="N" THEN 400
450 IF F$="I" THEN RUN 70
470 GOTO 470
490 PRINT CHR$(12):PRINTS 8,7,CHR
$(2) "!!! 1...5!!!":CHR$(18):PRINT:
PRINT:PRINT:PRINT
500 INPUT"MILYEN KATEGÓRIÁ-
BAN JÁTSZANAK?(1...5)":M
510 IF M<1 OR M>5 THEN 490
520 IF M=2 THEN M=10
530 IF M=3 THEN M=20
540 IF M=4 THEN M=40
550 IF M=5 THEN M=60
560 GOTO 70
```

A program kérésre kiírja az információkat; ezután a változók értékadása, majd a pálya kirajzolása következik. Gombnyomásra elindul a játék. Kettén játszhatják; az egyik játékos 1 és 1 nyilakkal, a másik az O és L betűkkel tudja mozgatni „embereit”. Mindkét játékosnak két-két „embere”, azaz ötöje van: egy a kapujában, egy pedig az ellenfél térélenek közepén. A két ütő egyszerre lehet mozgatni. Ha a labda falnak vagy ütőnek ütközik, akkor rugalmasan visszapattan, ha bekerül a egyik kapuba, akkor a legalsó sorban kiírt állás megváltozik: a gólt kapó fél pontja eggyel nő. Ha valamelyik játékos 10 gólt kapott, vége a játéknak, és a gép kiírja a győztest. Ha folytatódik a játék, tehát még nincs 10 gól, akkor a labda a gólt kapó játékos kapujától kerül játékba, mintha a kapus kirúgná.

Esetleg magyarázatra szorul a 390-es és az azt követő sorok szerepe. Miután a program kiszámolta az új koordinátákat (a labdát), megvizsgálja, nincs-e ott ütő vagy fal. Ha igen, akkor a régi koordinátákat számolja ki, a fal vagy ütő elé kirajzolja a labdát, majd egy kis idő múlva leoltja, és az elmentés irányban halad tovább a labda (ütőkész). Az 560-as sor a gól után a labda irányát ellentétesre állítja, így az újra játékba kerül.

A labda úgy mozog, hogy az X és Y koordinátáikhoz hozzáadjuk U és V változó értékét (2 vagy -2). Ha értékük ellentétes lesz, akkor megfordul a labda útja.

## A program magyarázata

- 10—170 Információ.
- 200—299 A változók értékének megadása és a pálya kirajzolása.
- 300—345 A billentyűzet figyelése, a labda és a játékosok kirajzolása.
- 350 A labda új helyének kiszámítása.
- 360—370 Megvizsgálja, nincs-e gól.
- 380—430 Megvizsgálja, hogy ütközik-e falnak vagy ütőnek a labda.
- 500—560 A gólokkal foglalkozó rész.
- 520—540 Kiírja az állást: ha van 10 kapott gól, akkor vége a játéknak.
- 560— A gólt kapó játékos kapuja felől kerül játékba a labda.

```

10 CLS
20 PRINT"FE'L KE'PERNYO !":PRINT
30 INPUT* KERSZ INFORMACIOD * ";A#
40 IF LEFT$(A#,1)<>"1",200 ELSE CLS
50 PRINT" * * * * * "
60 PRINT"*
70 PRINT"* T I D - J Á T É K *
80 PRINT"*
90 PRINT" * * * * * "
100 PRINT:PRINT" 1. JATEKOS: FEL ES LE A"
110 PRINT"          NYILAKKAL MOZOGHAT."
120 PRINT" 2. JATEKOS: O - FEL"
130 PRINT"          L - LE"
140 PRINT:PRINT"AZ VESZIT, AKINEK ELOBB LESZ"
150 PRINT"TIZ KAPOTT GOLJA."
160 PRINT:PRINT#MEHET A JATEK ?"
170 IF INKEY#="" ,170
200 CLS:DEFINT A-Z:A=15010
210 B=15037:X=32:Y=23
220 U=2:U=2:Z=14338:H=14400:C=0
230 FORI=0 TO 62:SET(I,2):SET(I,42):NEXTI
240 FORI=2 TO 9:SET(0,I):SET(62,I):NEXTI
250 FORI=35 TO 42:SET(0,I):SET(62,I):NEXTI
260 FORI=2 TO 42 STEP2:SET(31,I):NEXTI
270 PRINT@960,"HA MEHET, NYOMJON LE VALAMIT";
280 IF INKEY#="" ,280
290 PRINT@960,"AZ ALLAS:          ";
299 REM * FORPROGRAM *
300 SET(X,Y):K=PEEK(H):L=PEEK(Z)
310 IFK=8 AND A>15400,POKEA,32:POKEA+20,32:A=A-64:GOTO330
320 IFK=16 AND A<16100,POKEA,32:POKEA+20,32:A=A+64
330 IFL=128 ANDB>15400,POKEB,32:POKEB-20,32:B=B-64:GOTO345
340 IFL=16 ANDB<16200,POKEB,32:POKEB-20,32:B=B+64
345 POKEA,149:POKEB,149:POKEA+20,149:POKEB-20,149
350 RESET(X,Y):X=X+U:Y=Y+U
360 IF X=0 AND Y>9 AND Y<35,500
370 IF X=62 AND Y>9 AND Y<35,510
380 IF Y=3 OR Y=41,U=-U
390 IF POINT(X,Y),X=X-U:Y=Y-U ELSE 300
400 SET(X+U/2,Y+U/2):FORI=1 TO 20:NEXTI
420 RESET(X+U/2,Y+U/2):Y=Y+U:U=-U
430 GOTO 300
500 B1=B1+1:GOTO520
510 B2=B2+1
520 PRINT@970,B1;" - ";B2;
530 C=C+(B2=10)+(B1=10)*2
540 IF C,CLS:PRINT@512,-C;"-ES JATEKOS NYERT !":END
560 X=(SGN(-U))*6+X:U=-U:GOTO 300

```

### A változók szerepe

A	Az 1-es csapat kapusának a helye.	B1	Az 1-es csapat kapott góljainak száma.		
A+20	Az 1-es csapat játékosának a helye.	B2	A 2-es csapat kapott góljainak száma.	I	
B	A 2-es csapat kapusának helye.	X, Y	A labda jelenlegi X és Y koordinátája.	K, L	
B-20	A 2-es csapat játékosának helye.	Z, H	Értékük 14 338 és 14 400; a billentyűzet figyelését könnyítik meg.		
C	A győztes csapat (játékos) száma.	U, V	Minden alkalommal ezek értéké-		

vel növeljük meg X és Y értékét (értékük 2 vagy -2). Ellentettjük használatakor a labda ellentétes irányba mozog.  
Ciklusváltó.  
A billentyűzet figyelése alatt értékük PEEK(Z) és PEEK(H) (Z és H értékét lásd feljebb!).

BACZÓ TAMÁS

# Z80 programozási gyakorlatok 7.

Cikksorozatunk utolsó részében egy nagyon sokat vitatott problémáról, a rekurzióról szöveg, és ennek kapcsán egy kicsit túl is tekintünk az assembly programozás kerekein.

Sokszor hangoztatott vélemény, hogy a rekurzió mint programozási fogás felesleges, mert amit rekurzióval meg lehet írni, azt meg lehet anélkül is. Azt mondják, gépi kódban nem készíthető rekurzív program, és mégis minden megoldható ezen a nyelven. Nos, ez az érvelés nem egészen jó, mert van, amit nem érdemes megírni rekurzió nélkül, és mert gépi kódban sem lehetetlen rekurzív programok készítése, csak elég nehézkes.

Nézzünk meg egy rendkívül egyszerű, bináris fáth kiíró szubrutint! A felépített fára az *ábra* mutat példát. Ennek a fának minden egyes eleme tartalmaz valamilyen értéket, továbbá két mutatót, amelyek az elem bal, illetve jobb fáira mutatnak. A fa felépítése olyan, hogy a gyökér bal oldali részfájának minden eleme kisebb értéket hordoz, mint a gyökér, a jobb oldali pedig nagyobb. Ez igaz a részfák gyökereire és azok mindkét oldali részfáira, és így tovább. A kiírás úgy megy végbe, hogy az elemek értékei nagyság szerint következzenek. Világos, hogy mielőtt a gyökérhez tartozó értéket kiíratnánk, ki kell íratni annak bal oldali részfáját, hiszen ott minden elem értéke kisebb a gyökér értékénél. A jobb oldali részfa viszont csak a gyökér után következhet, hiszen ott minden érték nagyobb a gyökérénél. A kiírás tehát valahogy így megy:

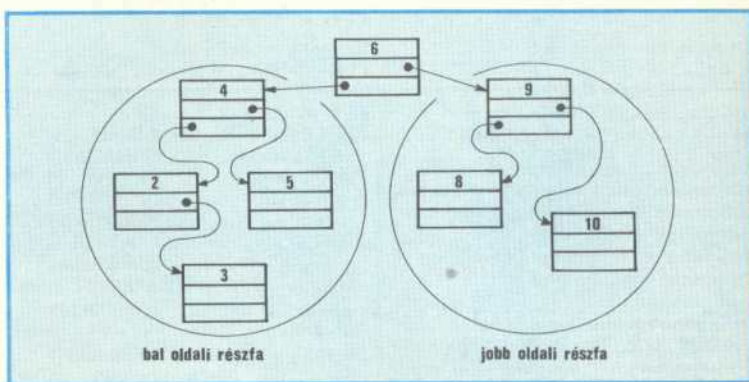
1. Irassuk ki a gyökér bal oldali részfáját.
2. Irassuk ki a gyökér értékét.
3. Irassuk ki a gyökér jobb oldali részfáját.

Ezzel meg is van a szubrutin. Csupán az 1. és 3. pontokkal van gond: hogyan irassuk ki a részfákat? Nem másként, mint a most megírt szubrutinnal. Ez a rekurzív hatás. Nagyon fontos, hogy valahol kilépjünk a rekurzióból, azaz ha nincs bal vagy jobb oldali részfa, akkor ne hívja meg a szubrutin önmagát a nem létező részfa kiírására.

## 1. program

```

AE13 20 PROGRAM IIRTREE1
AE13 30 TYPE TREE=RECORD
AE13 40   NUM:INTEGER1
AE13 50   LEFT:POINTER1;TREE
AE13 60   END1
AE13 70   POINTER#TREE1;
AE13 80 (*L*)
AE13 100 PROCEDURE WRITREE(P:POINTER);
AE1F 110 BEGIN
AE37 120
AE37 130 IF P^.LEFT<NIL THEN
AE32 140   WRITREE(P^.LEFT);
AE68 150
AE68 160 WRITELN(P^.NUM);
AE78 170 IF P^.RIGHT<NIL THEN
AE98 180   WRITREE(P^.RIGHT);
AEA7 190 END1;
AE37 200 (*L*)
End Address: AECC
    
```



Az eljárást PASCAL nyelven mutatja az 1. program; a nyelv támogatja a rekurzív hívást.

Nézzük meg, hogyan lehet megírni a szubrutint gépi nyelven! A PASCAL függvény argumentuma egy mutató. A Z80 szubrutin (2. program) ezt a mutatót az IX regiszterben várja, azaz mielőtt valamilyen program meghívja a WRITE szubrutint, előtte el kell helyezni IX-ben a gyökér címét. A szubrutin megvizsgálja, hogy van-e bal oldali részfa, és ha nincs, akkor a NOKIR címkenél — ahol a gyökér értéket írja ki a program — folytatódik a futás, majd megnézi, hogy van-e jobb oldali részfa. Ha nincs, akkor a szubrutin befejezte a működését, és visszatér.

Lássuk, mi történik akkor, ha van bal oldali részfa! Ebben az esetben a szubrutin meghívja önmagát, de előtte az IX regiszterbe helyezi a bal oldali részfa gyökérének címét. A visszatérés után ki kell íratni a gyökér értékét, de ehhez szükség van annak címére, ezért meg kell őrizni IX értékét a gyökér hívás előtt. Nem használhatunk valamilyen változót erre a műveletre, hiszen akkor azt is el kellene menteni a hívás előtt, mivel a szubrutinnak szüksége van rá. Olyan helyre kell IX-et elmenteni, ahol biztonságban van, azaz amelyik helyet a szubrutin nem érinti. Ez a hely a verem. Amikor a jobb oldali részfát íratjuk ki, akkor már nem rekurzív módon hívja meg önmagát a program. Természetesen felesleges lenne

## 2. program

```

FEED
ORG 100H
*****
!* 2. PROGRAM *
!*   Bináris fáth listazo *
!*   szubrutin *
!* IX=érték *
!* bal *
!* jobb *
!* *
!* V.P. 7C2.Nov. *
*****
0100 D06E02 WRITREE LD L,(IX*2) ;HL=
0103 D06E03 LD H,(IX+3) ; left
0106 7C LD A,H
0107 B5 OR
0108 280A JR 2,NOKIR ;ha left=NIL
010A D0E5 PUSH IX ;IX elmentese
010C E5 PUSH HL ;IX=
010D D0E1 POP IX ; HL
010F C0001 CALL WRITREE ;rekurzív hívás
0112 D0E1 POP IX ;IX vissza
0114 D06E00 NOKIR LD L,(IX+0) ;HL=
0117 D064D1 LD H,(IX+1) ; num
011A C0EEFE CALL WRITEL HL ;írás
011D D06E04 LD L,(IX+4) ;HL=
0120 D06605 LD H,(IX+5) ; right
0123 7C LD A,H
0124 B5 OR L
0125 08 RET Z ;ha right=NIL
0126 E5 PUSH HL ;IX=
0127 D0E1 POP IX ; HL
0129 1825 JR WRITREE ;csiklus!!!

Workarea - A3E5 to A526
ORG end - D12B
LOAD end - 0000
    
```

# BASIC és gépi kód

elementeni a mutató értékét, rá már nincs többé szükség. Ez a hívás megfelel egy egyszerű ciklusnak. Ciklus helyett nem szabad rekurziót alkalmazni!

Gyakori, hogy a rekurzió demonstrálására a faktoriális-számító programot hozzák fel példának, rendkívüli egyszerűsége miatt. Programozástechnikai szempontból azonban ez éppen a legtipikusabb ellenpélda arra, hogy mikor nem szabad rekurziót alkalmazni!

Térjünk vissza a bináris fa kiíratásához! Mennyivel egyszerűbb mindez PASCAL nyelven! Nem kell az előbbiekkal törődni, a nyelv mindezt megteszi helyettünk.

Vajon a változóit mind belenyomja a verembe? Természetes, hogy nem tárolhat mindent regiszterben; ha sok változó van a függvényben, akkor a többszörös memóriában kell tárolni. Ugyanúgy, ahogy hosszabb assembly program megírásakor tesszük: helyet foglalunk például DEFS direktívával. Az ilyen változók a program futása során mindig ugyanott vannak, tehát egy rekurzív hívás előtt el kell őket menteni. Mivel a fordító nem tudhatja egy hívásról, hogy rekurzív-e, hiszen lehet, hogy A függvény hívja B-t és B hívja A-t, ezért minden függvényhívás előtt el kell menteni az összes változót.

Ennél egyszerűbb, ha a függvény változóit dinamikusan helyezzük el, azaz amikor egy függvényt meghív a fő programrész vagy egy másik függvény, akkor a változók számára szükséges tárhelyet megcímizzük egy mutatóval, és az egyes változók címét a mutató plusz egy eltolási érték adja. Z80 assemblynél ez a mutató lehet az IX vagy IY regiszter, 6502 processzornál egy nullalapú tároló, 8086 processzornál a BP regiszter.

Amikor a függvény meghív egy másik függvényt, akkor a mutató régi értékét elmenti a verembe, majd beállítja az új értéket, és végrehajtja a hívást. A visszatérés után POP utasítással kerül vissza a mutatóba a régi érték, és folytatódik a függvény végrehajtása. A fő programrész változóit nem dinamikusan tárolják; ezek ugyanolyanok, mint assemblyben a DEFS direktívával lefoglalt területeken levő változók; tehát fix címük van a futás során, míg a függvények egyéb változóinak a címe más és más attól függően, hogy honnan és mikor lett meghívva a függvény.

Programunkban volt még egy lépés, amelyet — bár a feladat egyszerűsége miatt az előbb nagyon könnyen megoldottunk —, részletesebben meg kellene vizsgálnunk. Ez a paraméterátadás. Nagyon hasonló probléma, mint a helyfoglalás a függvény saját változó számára. A különbség csak annyi, hogy ezeknek a változóknak kezdőértékük van.

Maradjunk a PASCAL nyelvnél, és csak az értékkel hívott paraméterekkel foglalkozunk! A paraméterek helyét tehát ugyanúgy le kell foglalni, de még a függvény meghívása előtt ezekre a helyekre az aktuális paramétereknek megfelelő értéket el is kell helyezni. Szokás, hogy mind a paraméterek, mind a lokális változók számára a helyet a veremben biztosítják a fordítók. Ennek az az előnye, hogy a paraméterértékek elhelyezhetők a PUSH utasítással, a

Legutóbb a számláló utasításokról és a közvetett JMP utasításról volt szó. Most az utasítások egy újabb csoportjával ismerkedünk meg. Megértésük a különleges címzési mód miatt egy kicsit nehezebb az eddigieknél.

## Feltételes ugróutasítások

Az eddig megismert utasításokkal csak olyan gépi kódú rutinokat tudunk írni, amelyekben a végrehajtás sorrendje rögzített, és ettől a sorrendtől a futás során nem lehet eltérni. Az itt következő utasítások lehetővé teszik, hogy a végrehajtás sorrendje bizonyos feltételektől függően különböző legyen.

Nyolc utasítás tartozik ebbe a csoportba. Nevük az angol BRANCH (elágazás) szó kezdőbetűjéből és az ugrás feltételének két-betűs angol rövidítéséből áll. A feltétel az állapotbajt egyes biteinek pillanatnyi értékéből adódik, ez a segédletben is megtalálható. A feltételbitek értékét az ugróutasítások nem módosítják.

## A relatív címzési mód

A feltételes ugróutasítások mindig két-bájtosak, és a relatív címzési módot használják. Ez azt jelenti, hogy az ugrási feltétel teljesülése esetén az utasításszámláló regiszter (PC) tartalmához egybájtos előjeles számmint hozzáadják az utasítás operandusát. A program futása az így keletkezett új címen folytatódik. Az operandus az ugrási címnek és az ugróutasítást követő utasítás címének előjeles különbségét tartalmazza. Gépi kódban programozva ezt az értéket nekünk kell kiszámítani; vigyázni kell, mert könnyű eltéveszteni! Velem is többször előfordult, hogy az ugróutasítástól számoltam a távolságot, nem az azt követőtől.

Könnyebb dolga van annak, aki valamilyen assemblerrel dolgozik. Az operandust kétbájtos abszolút címként kell megadnia; az egybájtos gépi kódú operandust az assembler kiszámítja.

helyfoglalás pedig a többi változó számára a veremmutató csökkentésével érhető el. A mutató maga az SP vagy valamely más megfelelő regiszter lehet, amelybe az SP-vel egyező értéket töltünk. Ez célszerű is, hiszen például egy értékadás is szubrutinhívások sorozatává válhat a fordítás során.

Az eddig leírtak természetesen nem tényleg, csupán szokások, amelyeket az ésszerűség diktált. Biztos például, hogy egy 6502 processzoros gépen futó PASCAL program nem használja a vermet paraméterátadásra, hiszen az első lap 256 bajtja erre nem alkalmas, de minden bizonnyal valamilyen hasonló struktúrával dolgozik. A ZX-Spectrum futó PASCAL nyelv a leírt módon

A segédletben a relatív címzési mód leírásába híba csúszott. Az első bekezdés pontos, az azt követő példában a BEQ \$24 nem 34-gegy, hanem 36-tal viszi előre a PC-t, míg a BEQ \$FF nem 126-tal, hanem csak 1-gegy hátra. (Ami ugyancsak furcsa eredményt produkál, tekintve, hogy az ugrás a BEQ utasítás operandusára történik.) A következő bekezdésben is híba van, ugyanis hátra 128, előre 127 bajt nyit ugorhatunk.

## Az utasítások leírása

BPL — Branch on PLUS. Ugrás, ha az N-bit tartalma 0.

BMI — Branch on MINus. Ugrás, ha az N-bit 1-et tartalmaz.

BVC — Branch on oVerflow Clear. Ugrás, ha a V-bit 0-t tartalmaz.

BVS — Branch on oVerflow Set. Ugrás, ha a V-bit 1-et tartalmaz.

BCC — Branch on Carry Clear. Ugrás, ha a C-bit 0-t tartalmaz.

BCS — Branch on Carry Set. Ugrás, ha a C-bit tartalma 1.

BNE — Branch Not Equal. Ugrás, ha a Z-bit tartalma 0.

BEQ — Branch on Equal. Ugrás, ha a Z-bit 1-et tartalmaz.

A feltételes ugróutasítások alkalmazásait legközelebb egy gyakorlati példán mutatunk be.

## Néhány észrevétel

Székely Béla kollégám figyelmeztetett, hogy az 1986. szeptemberi számban közölt ROM-cím összehasonlító táblázatban híba van, a C16 oszlopban a negyedik cím helyesen \$9314. Utánanéztém, a gépelési hibát én követtem el, elnézést kérek. A júniusi számban megjelent BASIC betöltőben a hibátlan cím szerepel.

Kollégám néhány nappal később megemlítette, hogy a kiszámított GOTO alprogram kétser olyan hosszú, mint szokásos. Igaza van, de erről majd máskor írok. BARNALASZLÓ

oldja meg a problémát, és az említett mutató az IX regiszter. Ez a processzor sokkal alkalmasabb az ilyesmi megvalósítására. A 8086 processzor a tervezői úgy konstruálták meg, hogy utasításkészlete egyenesen sugallja, hogy a paramétereket a veremben adjuk át.

Gyakorlatként ajánlom az olvasónak, hogy próbálja meg megírni — először esetleg magas szintű nyelven — Hoare 1962-ben kitalált gyorsrendezési algoritmusát. Az algoritmus megtalálható például Aho—Hopcroft—Ullman: Számítógépalgoritmusok tervezése és analízise c. könyvben (Műszaki Könyvkiadó, 1982).

VERHÁS PÉTER

# A programozás Trabantja



Semmiképpen nem véletlen, hogy a programozással kapcsolatban az emberek egyre növekvő többsége a BASIC-re gondol. Ugyanakkor a hivatásos programozók gyakran óva intenek e nyelv használatától, mert rontja a stílust, nehezíti az „igazi” programozás mesterségének (művészetének) elsajátítását.

Mivel a BASIC-re sokan felnéznek (például a TV-BASIC-ből vizsgázott több ezer ember), és számosan lenézik azt, idősebbeknek látszik a nyelv tényleges helyének meghatározását, érdemeinek és hátrányainak felmérése, egybevetése. A téma aktualitására utal a hazai BASIC irodalom jelentős mértékű szaporodása, valamint az a tény is, hogy idén áprilisban az NJSZT megfelelő szakosztályának műsorán is szerepelt.

A hagyományoknak megfelelően először a BASIC előnyeit vizsgáljuk, és megpróbáljuk elejét venni egyes — elsősorban a korszerűségét kétségbe vonó — kritikáknak.

A BASIC kialakulásának dátuma alapján első generációs nyelvről számít. Születésének idejét 1964-re teszik, a szülők (J. G. Kemeny és T. E. Kurtz) egyike magyar. Ekor világszerte az első és második hardver-generációs gépek működtek, a tipikus tárméret néhány ezer bajt — bár ekkor még főleg egyszerűsített operatív táraikat használták —, az átlagos műveleti sebesség pár százötlet pár ezer művelet/sec-ig terjedt. A nyelvek közül a BASIC elődje — kortársa például a FORTRAN, a COBOL és az ALGOL-60.

Elfogadva, hogy a nyelv az elmúlt 20 évben korszerűsödött, új utasításokkal bővült, szabad-e a modernségét számon kérni, a jóval később „felfedezett” strukturált programozás támogatásának hiányát felróni egy olyan nyelvnek, amely túlélt az ALGOL-t, pedig ennek bölcsője körül ott bábáskodott a világ akkori számítástudományi elitje, P. Naurtól J. W. Backuson keresztül. A van Wijngaardenig? Nyilván nem.

A BASIC legfontosabb erénye az egyszerűség. Ez a tulajdonság kétarcú: könnyű a nyelvet akár előzetes ismeretek hiányában is elfogadható szinten elsajátítani (a TV-BASIC tanfolyam adásideje  $15 \times \frac{1}{2}$  óra) és viszonylag nem nehéz egy-egy konkrét gépen megvalósítani sem.

Az átlag felhasználót az elsajátíthatóság érinti közvetlenül, számára az egyszerű megvalósíthatóság abban nyilvánul meg, hogy már 4 kilobájtos, zsebméretű gépeken is használhatja a BASIC-et.

A nyelvet azért könnyű megtanulni, mert kevés, egyszerű fogalommal dolgozik, utasításkészlete szerény, az utasítások pedig köztérhető, tipikus tevékenységeket hajtának végre.

Az egyszerű megvalósíthatóság nagyrészt a nyelv szerkezetén alapszik, amely ahol csak lehet, segíti a gépi feldolgozást. Ennek érdekében írunk egy sorszámat a program utasításai elé (a modernebb változatokban két sorszám között már több utasítás is állhat), és ezért kell a numerikus és szöveges változók neveit egymástól megkülönböztetni (X és XS). Az általában nem nagyon hatékony értelmező rendszer munkáját segíti, hogy minden utasítás egy jellemző kulcsszóval kezdődik, és így azonnal felismerhető. Az értékadásban szereplő LET kulcsszó elhagyását újabbban sok értelmező eltűri.

A BASIC további előnye az átlagosnál jobb felhasználói kapcsolat. A rendszer általában egy szerkesztő alrendszert is tartalmaz, amely a már előbb említett sorszámkat jól használja fel a BASIC program sorainak rendezésére. Sok szerkesztő az igen barátságos, ún. „full-screen”, azaz a képernyő tetszőleges pontját értékelő módon működik. A BASIC tervezői tehát jól látják, hogy a programozók munkájuk során túlnyomórészt írják vagy módosítják programjukat, ezért ebben kell őket hatékonyan támogatni. A program írása inkrementálisan, utasításról utasításra haladva történik, és minden utasítás elfogadásáról vagy hibájáról a rendszer azonnal felvilágosítást ad. Kár, hogy ettől az igen hasznos megoldástól több népszerű BASIC elterít, például a C64 is.

Ha furcsán hangzik is, úgy tűnik, hogy a BASIC-nek nemcsak erényei, hanem bűnei is az egyszerűségéből fakadnak. A nyelv legszembetűnőbb hiányossága, amely a modern programozási elvek érvényesülését is akadályozza, az igazi eljárások hiánya. A GOSUB még a névvel rendelkező, paraméter nélküli eljárásokat sem igen pótolja,

pedig egy strukturált programterv lépésenkénti finomítással történő gépi megvalósítása jól használható paraméterátadási mechanizmust igényel. A paraméterek globális változók segítségével való átadása felesleges, a feladat lényegéhez nem tartozó tevékenységeket erőltet a programozóra. Ebből már következik, hogy az IF... THEN utasításban igen gyakran szerepel a GOTO, ami a programot áttekinthetetlenül teszi.

Másik jogos vád a BASIC ellen az, hogy az értelmezési technika, ami igen hasznos a rendszer barátságos viselkedése szempontjából, meglehetősen lassú programfutáshoz vezet. Ezzel kapcsolatban nem szabad figyelmen kívül hagyni a BASIC fordítók és azt a tényt, hogy nagyobb teljesítményű gépeknél a kis hatékonyságú értelmezés sem jelent problémát.

A BASIC egyszerűségéből, valamint a programozó szabadságának biztosításából eredően sok rendszerben a nyelv nem egyértelmű. Általában például nem kötelező szöveget hagyni egy utasítás alkotórészei között. Ha azonban ezt a FOR I=S TO P utasítással próbáljuk megtenni, több ismert rendszer (például a C64) csöddőt mond, mivel az S TO P egybeírva a STOP utasítást adja.

Az egyszerű utasítások kifejező ereje többféleképpen is növelhető. Az egyik mód például az általánosított VALUE függvény alkalmazása, amely nemcsak számokat, hanem egy szöveges változóban tárolt tetszőleges aritmetikai kifejezést is elfogad és kiértékel.

Ha mérlegelt akarunk vonni a BASIC előnye és hátrányai között, azt is figyelembe kell venni, hogy a nyelv milyen nagy mértékben járult és járul hozzá a számítástechnika terjesztéséhez. Az általam kiharozott egyenleg, amely természetesen nem teljesen objektív, okvetlenül pozitív. A BASIC a maga szintjén megfelelő eszköz. Olyan, mint a Trabant a magyar átlagpolgár számára.

Végül — a hasonlatra alapítva — egy jó tanács és egy észrevétel:

Trabanttal nagy hiba lenne Forma—1 futamon indulni.

A jelenlegi trabantosok jó része szívesen áttűnne egy jobb kocsihoz.

BAKOS TAMÁS



Egyre többben dolgoznak Commodore 64 számítógépen, és ezzel egy időben jelentősen gyarapszik a jogtalan másolással terjedő programok száma. Ezen szörötnékn segíteni. Hosszabb-rövidebb ideig meggátolhatjuk a programunk terjedését, ha hatásosan védjük a különböző másolóprogramok ellen.

Mielőtt rátérnénk az e célt szolgáló különféle technikákra, meg kell ismerkednünk a VC 1541-es meghajtó hardver és szoftver felépítésével. Igaz, erről már több könyv is megjelent a Data Becker kiadó gondozásában — például a „Das grosse Floppy—Buch” és a „Floppy VC 1541 Pflegen und Reparieren” —, de ezekből a kiadványokból pontosan a védelemre mutató részek hiányoznak. Most ezeket a hézagokat pótoljuk.

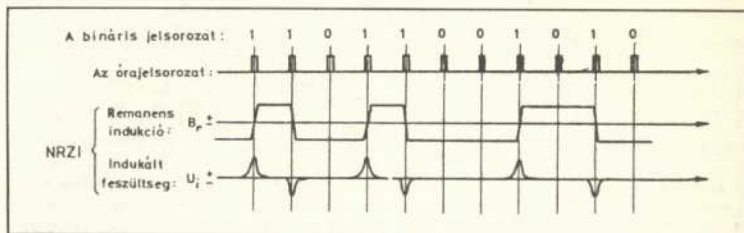
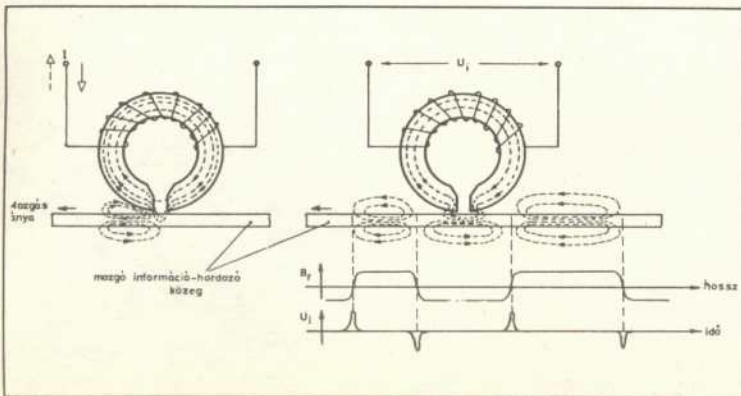
Nézzük először is, hogyan rögzíti a meghajtó a floppylemezen az adatokat. A lemez mágnesezhető bevonata általában vasoxid vagy Ni-Co ötvözetű fémréteg. Jelrögzítésre ennek a bevonaton picinyke felületdarabja mágnesesződik, és kis lokális mágnesként viselkedik; nagyságát mikrométerekben mérhetjük.

Az író-olvasó fejek sematikus szerkezetét az 1. ábra mutatja.

Beírásnál áramot vezetünk a tekercsbe. Mivel az információt hordozó közegnek a fejtől mért távolsága kisebb a mag légrésénél, a magban létrejövő mágneses erővonalak a légrés helyett többszörösen az információhordozó közegen keresztül záródnak. Ezáltal a közegnek azt a részét, amely az íróáram idején a fej alatt mozog, az áramiránynak megfelelő mágneses irányban telítésbe viszi. Miután ez a tartomány elhagyja a fej terét, megfelelő irányú indukció marad benne vissza. Ha az íróáram irányát megváltoztatjuk, az indukció iránya is megváltozik. Az azonos irányú indukcióval rendelkező összefüggő tartományok egy-egy kis mágneset alkotnak.

Kiolvasásnál ezek a kis mágnesek rendre a fej alá kerülnek, és erővonalaik egy része a fej magján keresztül záródik. Ha az erővonalak iránya hirtelen megváltozik, a tekercsben a változás irányának megfelelő polaritású feszültségimpulzus indukálódik.

1. ábra



2. ábra

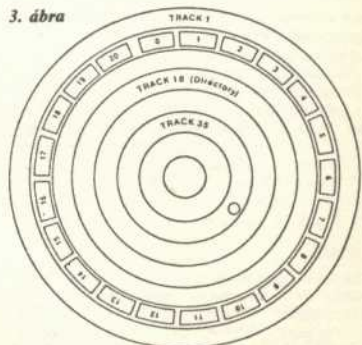
A digitális jelrögzítésre az egyik leggyakoribb módszert, a zérusra vissza nem térő invertáló (Not Return to Zero Inverting; NRZI) eljárás alkalmazták, amelynek jellegzetes képét láthatjuk a 2. ábrán egy adott bináris jelsorozatára.

Nézzük, ezekből a mágneses jelsorozatokból hogyan is épül fel a mágneslemez adatállománya.

A lemezegység motorja 300 fordulat/perc sebességgel forgatja a lemezt, ami a centrifugális erő hatására elveszti hajlékony jellegét. Így lehetővé válik az író-olvasó és a törölfej mozgatása egy léptetőmotor segítségével, amely sugárirányban mozgatja a fejeket. Ezzel a fejmozgatással a jelek koncentrikus körökben (sávokban) rögzíthetők. Két sáv távolsága megközelítően 1/40 inch, egy sáv szélessége pedig 1/80 inch. Ilyen módon a DOS (Disk Operating System) 35 sávot (track) használ, amelyeket kívülről és eggyel kezdődően számozik meg a rendszer. A lemezegység léptetőmotorja kétszer ennyi sávot is képes lenne behatárolni, de ez a lemez kapacitását jelentősen megnövelné, és ezt az író-olvasó fej nagysága nem teszi lehetővé. Ugyanis ha a léptetőmotor által megengedett 70 különböző fejállást felhasználnánk — vagyis 42 sáv közé pozícionálnánk a fejet —, akkor nagy valószínűséggel valamelyik sávon felülírnánk az ott lévő adatokat, tehát az megsemmisülne.

A rendszer a sávokat szektorokra osztja, amelyeket egy adatblokknyi (256 adatbajt) információ ír (lásd a 3. ábrát). Fontos, hogy 256 adatbajtról van szó, ugyanis a DOS az adatokat a később bemutatott GCR (Group Code Recording) kódformátumban rögzíti, amelyben egy adatblokk 325 bajtnyi.

3. ábra



Mivel a meghajtó — a legtöbb típustól eltérően — a külső sávokba sűrűbben ír, ott több szektor fér el; ilyen módon négy zónára osztja a lemezt, amelynek adatait az 1. táblázat mutatja.

1. táblázat

Zóna	Sávszám	Szektorszám	Szektorok száma	Bajt
1	1—17	0—20	21	5376
2	18—24	0—18	19	4864
3	25—30	0—17	18	4608
4	31—35	0—16	17	4352

Látható, hogy standard formattálás esetén 683 blokk található a lemezen.

A meghajtó a szinkronjel segítségével írja fel az adatokat a lemezre. A különböző írássűrűséget így éri el, hogy a 16 MHz-es órafrekvencia 13-as, 14-es, 15-ös és 16-os osztásával állítja elő az előbb említett szinkronjelet, így a négy zónában más-más időközönként kapja az impulzust. A 2. táblázat a pontos adatokat mutatja.

2. táblázat

Sáv	Az órajel leosztása	Két impulzus között eltelt idő
1—17	16 MHz/13 = = 1,230769 MHz	0,8125 μs
18—24	16 MHz/14 = = 1,142857 MHz	0,8750 μs
25—30	16 MHz/15 = = 1,066666 MHz	0,9375 μs
31—34	16 MHz/16 = 1 MHz	1 μs

3. táblázat

Sáv	1 bit	1 bájt	Bit/s	Bit/for- dulat
	felírás ideje			
1—17	3,25 µs	26 µs	307 692	61 538,4
18—24	3,50 µs	28 µs	285 714	57 142,8
25—30	3,75 µs	30 µs	266 667	53 333,4
31—34	4,00 µs	32 µs	250 000	50 000,0

A bitsűrűség 4000 bit/inch és 6000 bit/inch között mozog. Ha az egész lemezen a megengedett legnagyobb írássűrűséggel dolgoznánk, akkor a teljes lemez kapacitása 2 027 676 bit, vagyis 253 459 bájt lenne. A váltakozó írássűrűség miatt a lemez tényleges kapacitása csak 174 848 bájt, amiből a felhasználó 169 984-et (256 × 664) tölthet fel, ugyanis a többi a rendszer foglalja le a lemezkezelés számára.

Hogyan szabályozhatjuk az írássűrűséget? Erre a célra a VIA 6522 chip B kapujának 5. és 6. bitje (PB5, PB6) hivatott, amelyet lemeze írásnál vagy olvasásnál a Disk controller állít be folyamatosan.

Az egyes bitkombinációk jelentését a 4. táblázat tartalmazza.

4. táblázat

Sáv	1 bit írássebessége	PB5	PB6
1—17	3,25 µs	0	0
18—24	3,50 µs	0	1
25—30	3,75 µs	1	0
31—35	4,00 µs	1	1

Ezeknek az adatoknak a birtokában már két védelmi elvet is megérthetünk. Mindkét módszerben közös, hogy a lemez egy részét a standard eljárástól eltérően formátáljuk. Mivel a lemezformázás módja sokféle lehet, csekély a valószínűség, hogy a másolóprogram éppen az általunk használt módszer szerint dolgozik.

Elősként készíthetünk egy olyan programot, amellyel megváltoztatjuk a lemez egyes sávjain az írássűrűséget, és így biztosítjuk, hogy a DOS ne olvashassa el. Ahhoz viszont, hogy mi hozzáferjünk, nem elég az íróprogramot elkészíteni, hanem egy olvasóprogramot is kell írni, amellyel az adatokat majd visszaolvassuk. Hogy kinek melyik sáv szimpatikus, melyikben kívánja a formátumot megváltoztatni, mindenki maga dönti el; ezért ha a védelem módszerét ismerjük is, mégsem lehet a lemezt lemasolni, mert ehhez tudni kellene, hogy melyik sáv formátumát változtattuk meg, továbbá azt is, hogy ezen a sávon az adatok milyen sűrűn helyezkednek el.

A második eljárás is a lemez némely sávjának formátumát módosítja, de az adatsűrűség változatlan marad. Emellettük, hogy bár a léptetőmotor több mint 70 sáv pozicionálására is képes, az író-olvasó fej mérete nem teszi lehetővé ennek kihasználását, mert ha két sáv közé írunk, akkor a határsávokon megsérülhetnek az adatok. Ezt a két sávot viszont nyugodtan feloldozhatjuk a titkosítás érdekében: így olyan területet nyerünk, amelyet csak a saját dekódoló programunk olvashat, tehát más nem férhet hozzá.

Ahhoz, hogy a fejet beállítsunk akár két sáv közé is, vagy hogy nagyobb bitsűrűséggel tudjunk írni a lemeze, még sok mindennel meg kell ismerkednünk. Ezt sorozatunk további részei remélhetőleg elősegítik.

KOVÁCS P. ATTILA

gyakran kell számítógépekkel olyan feladatokat megoldanunk, amelyeknek eredményeire valamilyen rajzok elkészítéséhez van szükség. Kényelmes, ha a gép által kiszámított adatokból nem nekünk kell a pontokat, vonalakat rajzeszközökkel, például görbe vonalzókkal megrajzolni, megszerkeszteni — ez az út egyébként is fáradságos, lassú, nem mindig kielégítően pontos, és sok a hibalehetőség is —, hanem a rajzolást is géppel végeztethetjük el. Ilyenkor a számítógép kiszámítja a rajzolóhoz szükséges adatokat, és a rajzológép működését is vezérli.

A gépi rajzolás, a regisztráló berendezések alkalmazása jóval korábbi technikai vívmány a modern számítástechnika rajzológépeinél: igaz, az előbbieket túlnyomórészt analóg jellel alakítható mennyiségeket regisztráltak.

A fejlődés ellenére a rajzológép (nem rajzgép!) — angol nevén plotter — nem volt mindig olyan gyakran használt számítógép-perifériális egység, mint napjainkban. A rajzológép mai tömeges elterjedéséhez a vizuális megjelenítési igény fokozódása mellett a rajzológépek árának jelentős csökkenése is hozzájárult. Ma már olyankor is készíttetünk rajzokat, ha ez nem kötelező, pusztán azért, hogy numerikus eredményeinket szemléletes formában tanulmányozhassuk, hiszen az emberi agy — tudósok szerint! — 60 000-szer gyorsabban fogja fel a grafikusan megjelenített információt, mint ugyanazt számok, illetve szöveg formájában.

Ezt könnyen és gyorsan megtehetjük, mert már tömegével állnak rendelkezésünkre olcsó és elfogadható pontosságú, számítógéphez, sőt még otthoni számítógépekhez is csatlakoztatható rajzológépek.

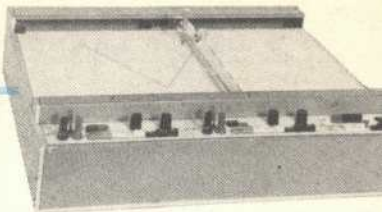
Igaz, hogy rajzolásra majdnem minden nyomtató alkalmas valamilyen mértékig. Számos rajzolósi feladatot nem is érdemes igazi rajzológépekkel elvégeztetni, hiszen (nem precíziós célokra!) megfelelő rajzokat kaphatunk egy közepes „pontosságú” és „felbontóképességű” tús, lézer vagy akár mágneses vagy tintaceppes nyomtató alkalmazása révén is.

Vannak azonban olyan pontosságra, vonalminőségre és rajzméretre vonatkozó követelmények, amelyeket csak rajzológépekkel tudunk kielégíteni. Foglalkozunk tehát röviden (a teljesség igénye nélkül) a rajzológépekkel kapcsolatos tudnivalókkal!

Jelenleg két rajzológép-konstrukció tömeges gyártása folyik. Az egyikben a rajzfelület (papír, pausz, film vagy fólialemez) rögzítve van a rajzolás kezdetétől annak befejezéséig. Ilyen például az EMG 79812 típusú vagy a RADELKIS OH—860 rajzológép.

A másik konstrukcióban a rajzfelület is mozgásba vészes a géphez képest.

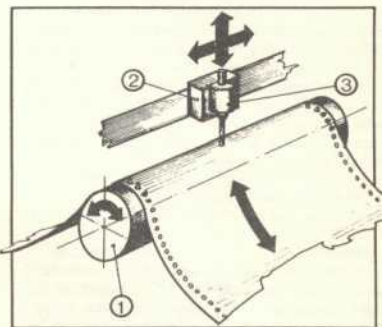
A rögzített rajzfelületű rajzológépeknek téglalap alakú síkfelületre (asztalra) rögzítik a rajzhordozót. A rögzítés rászívással vagy elektrosztatikus úton szokott történni. A rászívásnál az asztal meghatározott helyein apró (kb. 0,1 mm átmérőjű) furatok vannak, amelyeken keresztül állandóan



Az EMG 79812 típusú rajzológépe

szívják a levegőt. Az így létrejött vákuum rögzíti a papírt vagy más rajzhordozót.

Az elektrosztatikus rögzítés a különböző előjelű elektromossággal feltöltött testek közötti vonzóerőt használja fel a papír rögzítésére.

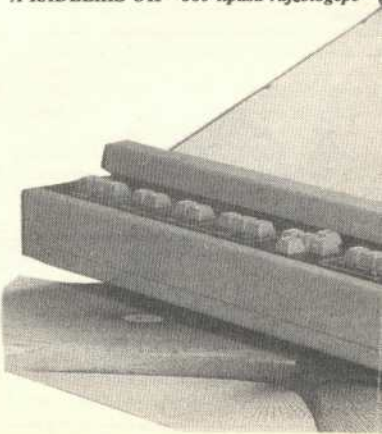


1. ábra. Az IMI EPA—207 rajzológép konstrukciója

zítésére. Az előbbieken említett RADELKIS OH—860 és az EMG 79812 típusú gépek ezt a rajzhordozó-rögzítési módot használják.

Ahhoz, hogy rajzológépünkkel rajzolhassunk, szükséges egy rajzolószerszemet is, amely a rajzeszköz befogó részéből, valamint magából a rajzeszköz(ök)ből áll.

A RADELKIS OH—860 típusú rajzológépe



# Rajzológépek

A több rajzeszközt tartalmazó rajzolószerveket befogónyílásaiban a rajzeszközök egyenes vagy körív mentén helyezkednek el.

A rajzeszközökkel szemben komoly követelmény az, hogy nagy sebességű rajzoláskor is folyamatos és egyenletes vastagságú vonalat húzzanak.

A különböző papír alapanyagú rajzfelületekre általában rosttollal, golyóstollal,

A rajzolószerveket mozgatószára különböző módszerek ismerünk.

A rögzített rajzfelületű gépéknél a téglalap alakú asztal egyik átellenes oldalpárján kialakított sínekre támaszkodik két végével egy mozgatható rúd (ún. „hid”), amely mindig csak az asztal másik oldalpárjával párhuzamos helyzetben lehet. E rúdön is van egy sín, amelyen e rúd mentén mozoghat az íróeszköz tartó kocsi. Az elmondot-

jedhetnek. A berajzolni kívánt rajz felület hátravonalaival általában programtechnikai úton jellemezhető ki oly módon, hogy megadjuk a bal alsó, illetve a jobb felső pontok koordinátáit. Erre rendszerint külön utasítások szolgálnak.

A rajzológépek másik nagy csoportját az ún. hengeres vagy dobos rajzológép — angol nevén drum plotter — képezi. Ez a típus kisebb pontosságú követelmények kielégítésére szolgál. Elnevezése onnan származik, hogy a rajzhorizontál alkalmazzott papírfelület egy henger alakú dob körül tekeredik le. Ilyen elven működik például az EPA—207 típusú rajzológép, amelyet az Ipari Műszergyártó Iklad (IMI) gyárt. A belső borítónkon szereplő ábra is ezzel a dobos rajzológéppel készült. Ezt a gépet a gyár vendégeként volt alkalmunk közelebből is tanulmányozni.

A konstrukció alapelveit az 1. ábra szemlélteti. A papír síkján belüli hosszirányú mozgást az  $\text{O}$  jelű papírmozgató dob végzi a papírtovábbító tuskék segítségével. Az oldalirányú (dob alkotó menti) tollmozgást a  $\text{O}$  jelű írófej végzi. A tollaknak a papírhoz való érintését és felemelését a  $\text{O}$  jelű tollmeghúzó elektromágnes végzi.

A működési alapelve az, hogy a léptetőmotorok megfelelő fogaskerék-átvitelen keresztül csatlakoznak a papírmozgató dobhoz, illetve a tollakat tartó írófejhez. Ha csak a dobot mozgattuk — és a toll a papírral érintkezik — „függőleges” egyenest kapunk. Fordított esetben — ha csak az írófejet mozgattuk — rá merőleges, „vízszintes” egyenest rajzol a toll. „Ferde” egyenes rajzolása mindkét motor működtetésével lehetséges, amikor is az egyenesek szöge a léptetőmotorok sebességarányától függ. Például  $45^\circ$ -os egyenes rajzolásakor mindkét motor egyforma sebességgel forog.

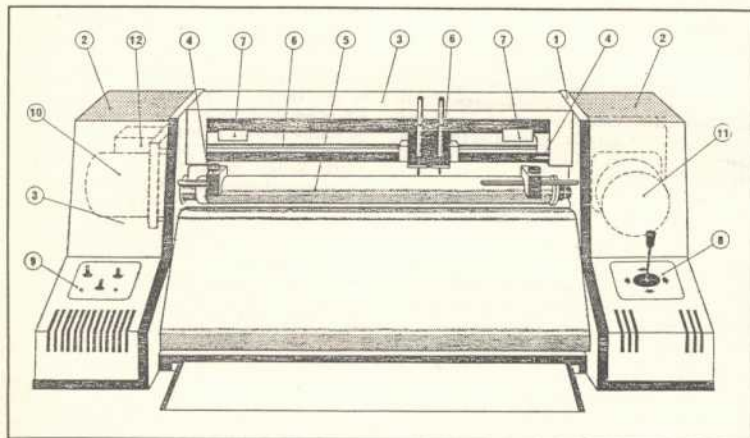
A 2. ábrán láthatjuk a rajzológép felépítését. Az  $\text{O}$  a víz, a  $\text{O}$  a műanyagból készült oldalborító lemez, a  $\text{O}$  fedőlemez védi az írófejet és az azt vezető sint, az ugyancsak  $\text{O}$  jelű papírréteg lemez a papír „megvezetéséről” gondoskodik. A  $\text{O}$  papírleszorító karok a papírtovábbító tuskékre szorítják a papírt.  $\text{O}$  a papírmozgató dob. A  $\text{O}$  rajzolófejbe két tollat foghatunk be.

A  $\text{O}$  végálláskapcsolók érzékelik a rajzolófejek szélső helyzetét, így behatárolhatják az oldalirányú rajzméretet. A  $\text{O}$  botkormányal (kézikapcsolóval) a gép off-line üzemmódjában módokan áll a toll helyzetét előzetesen beállítani, vagy segítségével akár „kézzel” is rajzolhatunk nyolc főirányban.

A  $\text{O}$  jelű rész a vezérlő előlap, ahol beállíthatjuk az üzemmódokat. A kezelő visszajelzéseket is kap a beépített világítódiodák (LED-ek) révén.

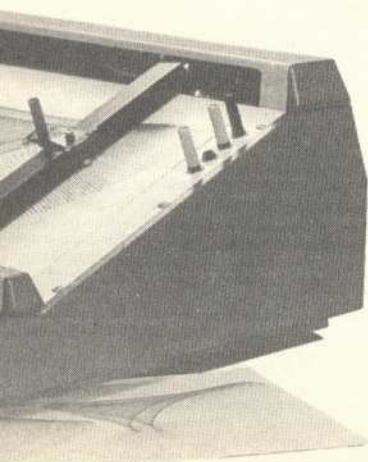
A dobmozgató egység a  $\text{O}$  jelű rész, el- lenpárja a  $\text{O}$  jelű léptetőmotoros rajzolófejmozgató egység. A rajzolófej mozgatása ennél a típusnál bordás szíjmeghajtással történik.  $\text{O}$  a szíjvezető ház.

A rajzológép hátoldali részén — ez ábránkon nem látszik — helyezkedik el a hálózati betáplálás a biztosítóházzal és a kapcsolóval, valamint a gép vezérléséhez szükséges illesztőcsatlakozók. Az illesztőcsatlakozók kétirányú CCITT V24-es és RS232 csatlakozást tesz lehetővé, max. 128 bájthosszúságú utasítások, illetve adatblokkok részére. Ezen keresztül illeszthető például PROPER 8—16, IBM PC, HT—1024 és Commodore gépekhez is.



2. ábra. Az EPA—207 rajzológép felépítése

igényesebb esetekben speciális csőtollal rajzolnak. A belső borítónkon látható zebra rajza golyóstollal készült. Elképzelhető az is, hogy nemcsak rajzeszközöket fognak be a rajzolószervezetbe, hanem különböző szerszámokat. Például üveglapok vagy fém- lapok gravírozásához karcoló tüket, műanyagok esetén különböző fejtőkéseket használnak.



takat jobban megérthetjük, ha az esztergápadok hossz- és keresztmetszeteire vagy a hídvaruk hidjára és futómascskájára gondolunk.

Vannak azonban más, például acélhűros mozgatószárral működő rajzológépek is. Ezeknél a típusoknál a hossz-szám, illetve a kocsimozgató végző keresztmetszete sinjei helyett acélhűrokat alkalmaznak. Az ilyen megoldású rajzológépek pontossága romlik az acélhűrok bizonyos idő után történő megnyúlása következtében. Az acélhűros mozgatószárral pontatlanságát küszöbölő ki a menetes orsóval vagy fogasléccel történő számmozgató. Ezek természetesen precíziós megmunkálást, valamint a hűtágulás okozta méretváltozásokat kiküszöbölő mechanikai rendszert igényelnek.

A rajzolószervezet és a papírmozgatószárra általában léptetőmotorokat használnak. A léptetőmotorok olyan tulajdonságúak, hogy az állórészükre adott vezérlőimpulzus hatására forgórészük mindig ugyanakkora „egységnyi” szögelfordulást végez. Így a forgórész mozgása szakaszos, nem akárhog állhat meg. A léptetőmotor forgásiránya is megváltoztatható. Az egy impulzus hatására bekövetkező szögelfordulás, az ún. lépésszög értéke általában  $1^\circ$  és  $30^\circ$  közé esik.

Ahhoz, hogy megfelelő pontossággal mozgathassuk a rajzolószervezetet, bonyolult elektronikával kell vezérelnünk a léptetőmotorokat, és precízen megmunkált mechanikára van szükség.

A rögzített rajzfelületű gépek rajzméretei kb. A4-es mérettől szabványi területű ter-

## és lesz!

A párizsi SICOB — mely 1949. évi alapítása óta a francia és az európai informatika elsőrangú bemutatójává fejlődött — 1986 őszén mármár szegényesedni látszott. A kiállítás száma 850-ről 600-ra csökkent, és a hagyományosan 10 napos kiállítás tavaly szeptemberben csak egy hétig tartott nyitva. A jelenség oka azonban nem más volt, mint visszahúzódás a nagy „ugrás” — terjeszkedés — előtt.

A SICOB kinötte a CNIT (Centre National des Industries et des Techniques) Defense negyedbeli területét, és — saját hirdetésének szavaival idézve — „elröpül” Villepinte-be. Villepinte-ben, Párizstól északra ugyanis 1986 októberében, az 5-ös számú csarnok átadásával befejeződött Európa legmodernebb kiállítási központjának, a Parc International D'Expositions Paris-Nordnak a felépítése. Ezzel lehetővé vált, hogy 1987 tavaszán a SICOB 117 100 m<sup>2</sup>-en fogadja látogatóit.

A megnövekedett kiállítási terület, az új környezet, valamint az egyidejűleg megrendezőző és tematikájukkal az kiegészítő események kitűnő alkalmat kínálnak a SICOB-nak: bővítené a bemutatott termékek körét, a világ egyik legnagyobb vásárvár- és üzleti központjává léphet elő.

1987. április 6—11. között Párizsban az új Nemzetközi Vásárház (Pont de la Porte Maillot-nál lévő Kongresszusi Palota) hat esemény színhelye lesz. Ezek a következők:

### 1. Maga a hagyományos SICOB

Nemzetközi kiállítás, melyen több mint 120 ország kiállítói mutatják be termékeiket négy ipari főcsoportban: irodaügynyitési eszközök; számítástechnika (hardver és szoftver, mikro is!); távközlési eszközök; irodaszervezési eszközök.

### 2. Irodaszervezési konferencia

1987-ben először hívják össze Franciaországban. Célja, hogy új metodikákat, technológiákat mutasson be, melyekkel kényelmesebbé tehetik környezetüket az ügyviteli rendszerek felhasználói.

### 3. Alkalmazástechnikai konferencia

Előadásai bővebb ismertetést nyújtanak a SICOB-on kiállított termékekről, azok gyártási technológiájáról, a gyártmányok főbb felhasználási területeiről.

### 4. Informatikai konferencia

Már szintén hagyományos: Európa legnagyobb szoftverkonferenciája.

### 5. Automatizálási konferencia

Az 1987. évi SICOB alkalmával először rendezte meg a GIPIRA (a francia vállalatok szövetsége). Az automatizálási szakemberekhez kíván szólni arról, hogyan élhetnek a számítógép adta lehetőségekkel speciális problémáik megoldásában.

### 6. SSII 87 — SICOB szolgáltatók és mérnöki informatika

Ez a szintén az idén bemutatkozó kiállítás főleg a szolgáltató és tanácsadó vállalatok munkájához nyújt segítséget azzal, hogy közepes és nagy rendszerekhez ajánl alap- és felhasználói szoftvert, szoftvercsomagokat.

A SICOB átköltözése Villepinte-be az eddigi gyakorlat megfordítását jelenti. Eddig ugyanis volt egy SICOB-nak májusban, amit speciális ünnepeknek neveztek, és volt egy „nagy” SICOB ősszel. 1987-ben pontosan fordítva lesz. Az 1987. áprilisi után, 1987. szeptember 15—20-ig a CNIT-ben lesz megtartva a SICOB Special Micro, egy időben a Kongresszusi Palotában rendezett Infodial-Videotex kiállítással. Így az idei év különösen gazdag programot kínál az informatikai szakembereknek. A SICOB-bal közel egy időben ugyanis Lyonban az INFORA, Hannoverben pedig a CEBIT is csábító lehetőség a tájékozódásra.

A 37. SICOB, mely utolsó volt a maga nemében, az alábbi fő területeken mutatott újdonságokat:

- a szupermikró megjelenése,
- az AT kompatibilis rendszerek fejlődése,
- helyi (lokális) hálózatok fejlesztése,
- grafikai munkahelyek (munkaállomások) tökéletesedése,
- nyomtatók.

A szupermikrók — külső megjelenésükre olyan „dobozok”, melyekben már nemcsak a VC kártyákat és az ellenőrző egységeket építették be, hanem a Winchester-lemezeket és a védőegységet is — egy vagy több multiprocesszor tartalmaznak, gyorsak (innen származik a nevük), el látják a Unix operációs rendszert vagy annak valamely változata vezérlését, és képesek a felhasználók tucatjait több funkcióban kiszolgálni. Ilyen kategóriájú gépeket a kisebb francia cégek közül a Sistral, az Internechnique, a Mercure és a Plessey France ajánlott az ősi SICOB-on. Annak ellenére, hogy az USA-ban ezek a típusok már a legnagyobb sikert aratták, itt csak kisebb érdeklődést váltottak ki. Az idei SICOB fogja eldönteni, hogy meg tudják-e vetni „lábakat” az európai piacon is.

Ugyancsak nagy választékot jelentek meg a mikro családhoz tartozó PC, XT, AT kompatibilis, hordozható Mac típusok. Ezek között az ADDX-System kínálta azt, amely a piacon az egyik legjelentősebb termékek mutatkozik; az AT kompatibilis ADD—X 286 rendszer 3. modellje 80286-os, 8-10 MHz-es mikroprocesszorral épül, kiegészíthető 1—9 Mbájtos RAM-mal, a lemez kapacitása 20—71 Mbájttal, a védőegysége 20 Mbájttal.

A hordozható típusok nagy száma egyébként jellemző volt a mikrók más kategóriáiban is. E téren az úttörő szerepet a Tandy, a HP és a Data General játszotta. A Data General az 1986-os SICOB-on mutatta be a DG/One új változatát, a 2. számú modell, amely tetszés szerint működtethető folyadékkristályos vagy normál képernyővel.

Másik érdekes vetülete volt az 1986-os kiállításnak a nagyszámú grafikus munkaállomás megjelenése, amelyek jól tükrözik a mikroprocesszorok fejlődését. Legtöbbjük 68020-as mikroprocesszorral épül, nemcsak a kielégítő sebességi paraméterek miatt, hanem főleg azért, mert ezek speciális áramkörrel alkalmazhatók a legjobban a grafikaiban és a CAD-ban. Az élenjáró kiállított a Gixi, a Sun Microsystems és a Control Data volt

tak ebben a kategóriában. Például a Gixi cég a Radiance 3020/3030 jelű munkaállomását mutatta be. Ez 68000-es 10 MHz-es áramkörrel és 68881-es kalkuláló processzorral, 3 D-s gyorsítóval, 19 hüvelykes 1024×800 pontos színes képernyővel, 1—4 Mbájtos memóriával van felszerelve. Ára 160 000 FF körül volt az alapváltozatban.

A távközlésben szenzációknak számított az x25 és x32 kártya, a Transpac vevő, az ATLAS 400-as üzenetrögzítő megjelenése, valamint a helyi hálózatok nagy száma. A helyi hálózatok fejlesztésének imponáló példáját mutatta be az ADDX-System. Az ADD X-Link egy helyi hálózat buszra csatlakoztatva, Novell logikai egységgel. Az RLX Link lehetővé teszi nyolc RLX rendszer egymás közötti kapcsolatát.

A technológiai érdekeségekről szólva, nagy figyelmet érdemeltek az optikai információhordozók (optikai lemez, videolemez, CD—ROM) között a CD—ROM első kereskedelmi prototípusai, illetőleg azok első ipari alkalmazásai a Philips és Sony cégeknél. A holnap fő memóriájának, a CD—ROM-nak további gyártói között 1987-ben várhatóan már francia cég is jelentkezik.

Az 1986-ban kiállított nyomtatók közül két modell újszerűsége keltett érdeklődést. Az AST France, amely a világon először állított elő személyi számítógéphez is használható kártyákat és perifériákat, most a Turbo lézerező jelent meg a piacon. Ez az összes PC-vel kompatibilis, olyan asztali lézernyomtató, amely gazdaságos komplett egységként működött, és újság minőségben képes nyomtatni, továbbá hátrahozott vonalú, egész oldalas grafikkákat rajzol. Memóriája 1,5 Mbájttal, amely a kontroll kártyára könnyedén disponálható. Saját kategóriájában az AST Turbo lézer különösen gyors, percnként 8 oldalt nyomtat. Ezt a teljesítőképességet az AST a lézerezés nélküli (LPC) köszönheti. A Turbo lézer az összes PC logikával kompatibilis, beleértve a táblázatkezelőket, a szövegszerkesztőket, az alapszámítógépek manipuláló kereskedelmi grafikkáknak építő, mint a CAO és DAO grafikkákat. Ára 12 000 FF, egyéves ingyenes karbantartással.

Egyik konkurens a Siemens, amely egy egészen más — a PT—85-ös hőátadásos — elven működő nyomtatójával jelentkezik. Nyomtatvány minőségű terméket ad, biztosítja a szöveg magas szintű variálhatóságát, és grafikkák megörökítésére is képes. A hőátadásos technológia mellett a nyomtatás tetszőleges papírra — a normál papírtól az átlátszó fóliáig — kerülhet. Könnyedén cserélhető a betűk formája is. A PT—85 kompatibilis az IBM PC-vel, ára 9170 FF volt. Jóval olcsóbb a lézernyomtatónál, viszont nem olyan gyors.

Az 1986-os ősz legjellegzetesebb vonása mégsem a fent említett új termékek, technológiák megjelenése, hanem az IBM kompatibilis gépek árainak zuhanása volt, ami 40 százalékra tehető a múlt év folyamán. A SICOB reagált a krízisre, és a Villepinte-be való átköltözéssel együtt felkészült arra, hogy meghatározó szerepet töltsön be a világ számítógépiparára jellemző főbb tendenciák bemutatásában.

SÁGHINÉ dr. HARTAI VERONIKA

# A DIGITÁL rejtvénypályázat eredményhirdetése

Kedves Fistolok!

Rejtvénypályázatunk lezárultával közöljük az egyes fordulók feladatainak megoldásait.

Mielőtt ezt megtennénk, engedjétek meg, hogy néhány mondatban összefoglaljuk a jó tanácsa jelleggel tapasztalatainkat.

A beküldött megoldásokból leszűrhető, hogy nagy lelkesedéssel láttatok hozzá a programíráshoz. Talán ebből a tényből adódtak a jellemző hibák is. Sokan nem ismerik a végleges - vagy csak nem alkalmazták a programkilusokot.

Többen nem vettétek figyelembe a feladatkiírást /futtathatóság C16-on és HT-n a második forduló feladatánál, az adatok sztringekben történő megadása a harmadik forduló feladatánál/.

Aki programot ír, annak nem szabad megfelekedni néhány fontos szemponttól sem:

- a program legyen jól tervezett,
- hatékony
- áttekinthető /más számára is érthető
- működőképes

**Az értékelés módja:**

A feladatokat az első kivételével azonos súlyúnak tekintettük, függetlenül a nehézségi fokban mutatkozó kisebb különbségektől. A megoldásokat pontoztuk. Az 1. feladatra 5, a többire - hibátlan megoldás esetén - 10 pontot adtunk.

A különleges vagy "elegáns" megoldásokat 1-2-3 további ponttal jutalmaztuk. Értékelési szempontnak tekintettük azt is, hogy mind a három forduló feladatait beküldték-e.

Az értékelés eredményeként a nyertes:

**TURCSÁN ZSOLT /2941 Ács, Kertész u. 3./**  
lett

34 ponttal, aki a DIGITÁL Számítástechnikai Szakuzletben /Jo26 Bp. Szilágyi Erzsébet fasor 35./ választhatja ki a kiíráásban felsoroltak közül a neki legmegfelelőbb nyeresémet. A nyertesnek gratulálunk!

Dicséret illeti Nyéki Pétert /Kemenesszentpéter, Jókai u.48./, aki 28 pontot, valamint Adamát Lajost /Kecskemét, Jókai u.1./, aki 26 pontot ért el.

További munkáikhoz sok sikert kíván a Szerkesztőség!

**A feladatok helyes megoldása:**

**Első forduló 1. feladat:**

A program a fokban megadott szög szinusztát számítja ki a sin x hatványsorával történő közelítéssel, 10<sup>-12</sup> pontossággal.

```
10 FORI=0:025:PRINT:NEXTI:REM ERNYOTORLES
20 PRINT "EMLÉKEZES":RESTORE
30 INPUT "AZ ÖSSZEJ":PI:IFPI<0:GOTO30
40 FORI=1:0.5
50 READA:K=INT (P/A):I=P-P-K*A
60 IFK<>0:THENPRINT: "DARAB":A:"FORINTOS"
70 IFPI<>0:THENNEXTI
80 INPUT "UDRA (I/N) Y/N"
90 IFAS="I" THEN10
100 IFAS="N" THEN120
110 DOTOBO
120 END
130 DATA1000,500,100,50,20,10,5,2,1
```

2. feladat: A címlétező program

```
10 REM ATVITEL A USER PORTON
20 REM AZ ADÓ PROGRAMJA
30 OUT31,7:OUT30,0
40 PRINT "A KARAKTER"
50 X=INKEY:IFX="" THENDOELSEX=ASC(X)
60 X=ASC(X)
70 OUT31,1:OUT30,1
80 OUT31,15:OUT30,X
90 OUT31,1:OUT30,0
100 OUT31,15:OUT30,0
110 DOTOBO
MÁSODIK FORDULÓ: Az adó programja
```

```
10 REM ATVITEL A USER PORTON
20 REM A VÉVŐ PROGRAMJA
30 OUT31,7:OUT30,0
40 OUT31,14:INP(31)
50 IFA(I) THEN40
60 OUT31,15:BINP(31)
70 PRINT "AZ ADAT:";CHR$(B)
80 DOTOBO
A vévő programja
```

Harmadik forduló:  
A digitális órá  
szimuláló program

```
10 DEFSTR:J=0:DATA31,28,31,30,31,30,31,31,30,31,30,31,31
20 RESTORE:INPUT "HO.NAP":A:IFLEN(A)<4:THEN20
30 HN=VAL(LEFT$(A,2)):NP=VAL(RIGHT$(A,2))
40 IFHN<10:RN=12:THEN20
50 FORI=1:TOHN:READB:NEXTI:IFB<N:THEN20
60 INPUT "ORA,PERC,NASODPERC":A:IFLEN(A)<6:THEN60
70 H=VAL(LEFT$(A,2)):M=VAL(MID$(A,3,2)):S=VAL(RIGHT$(A,2))
80 IFH<0:ORH<0:ORH<59:ORS<0:ORS<59:THEN60
90 IFINKEY$(C)="" THENJ=NOTJ:A=""
100 IFJ=-1:THENCLS:PRINTUSING "###":HN:NP:GOTO120
110 CLS:PRINTUSING "###":H:M:S
120 FORI=0:TOSSO:NEAT
130 B=S+1:IFB=60:THENB=0:ELSE90
140 M=M+1:IFM=60:THENM=0:ELSE90
150 H=H+1:IFH=24:THENH=0:ELSE90
160 NP=NP+1:IFNP=60:THENNP=1:READEL,SE90
170 HN=HN+1:IFHN=12:THENHN=1:RESTORE
180 GOTO90
```

# ADOK – VESZEK – CSERÉLEK

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk. A díjszabás: közletemek gépeelt soronként (60 karakter) 100,— Ft, magánzemélyeknek az első sor 50,— Ft, minden további sor 20,— Ft. Az NJSZT tagjainak az első három sor ingyenes. Hirdetéseiket a szerkesztőség címére várjuk.

■ 48 k-s Spectrumhoz játékprogramokat cserélek. A válaszokat listával kérem. Cseh Attila, Debrecen, Kapitány u. 16. 4033.

■ C64 programdokumentációkat keresek, cserélek. Cserébe 1985/86. évi fejlesztési szuper játékprogramokat, grafikai programokat (dokumentációval), zené- és sakkprogramokat adok. Programcsere is lehetséges — kizárólag lemezen. Honti József, Csákvár, Május 1. út 11. 8083.

■ Vállalatok, figyelem! Programozást, rendszertervezést, számítástechnikai szaktanácsadást vállalom. Egyedi programrendszerek készítése magas színvonalon C64, IBM PC kompatibilis gépekre. Telefon: 154-352.

■ Commodore 16-os játékprogramokat cserélnék. Sutus Péter, Bp. Kelli u. 9. fszt. 1. 1135.

■ C64 programokat cserélek kezten. Laczko Attila, Süllyás, Dózsa György út 1/c 2241.

■ Commodore 64 magnó, joystick, könyvekkel és programokkal eladó. Buzinszki Attila, Miskolc, Vasverő út 4. 3535.

■ Commodore VIC-20 magnóval, assembler, disassembler, monitor és játékprogramokkal, német, magyar nyelvű dokumentációval 10 000 forintért eladó. Oláh István, Szolnok, Kassai u. 56. 1/4. Telefon: 56-15-496.

■ EPROM-programozó készülékhez 28 lábas IC-foglalatók eladók (390,— Ft/db). Csányi Gábor, Mezőtúr, Ifjúsági lkt., 13. III. 16.

■ C64-es programokat cserélek. Cím: ifj. Sitkei Zoltán, 7200 Dombóvár, Gunaras kemping. Tel.: 74/65-523.

■ Floppy disk (ketoldalas, dupla sűrűségű) eladó. Tel.: 425-242.

■ ZX81 6 k-s bővítővel, szakkönyvekkel, programokkal 8000 forintért eladó. Tel.: 896-227. Cím: Krajcsovits László, 1043 Budapest, Virág u. 23. II. 16.

■ Commodore 64-es játékprogramokat cserélek. Bárdos Péter, Borsodbóta, Rákóczi út 162. 3658.

A haldoklóknak hitt Sinclair QL többek csodálkozására feléledt és táncra perdült.

Armstrad, aki szőröstül-bőröstül felvásárolta a Sinclair céget, 400 nyugatnémet márka körüli áron kiárusítást tart. Ez az alacsony ár természetesen sok új rajongót szerez a gépnek. A lelkes tábor és a néhány mecnás, akik a gép továbbfejlesztett változatának gyártására is vállalkoztak, elősegítik a QL reneszánszát.

Nálunk is felkérték táncra a QL-t: a tavalyi 300-zal szemben ugyanis napjainkban már 600 QL van hazánkban, és megalakult a QLub-juk is. A Csokonai Művelődési Ház (Bp. XV., Eötvös u. 64-66.) adott ott-hon a QLub-nak, amely még csak 33 fős létszámmal, de igazi lelkesedéssel indult. Itt a csatlakozni szándékozóknak is módjukban áll, hogy társakkal találkozzanak, egyelőre minden szerdán délután 5 és 9 óra között. (Faragó—Pacher)

## Ki ad magyarázatot?

A C16 egy közepes hosszúságú, kb. 8 kb-ajos BASIC program újrásorozásosakor különös dolgot produkált. A RENUMBER parancs szintaktikai hibát jelzett, és a program fele „deformálódott”.

A programot csak úgy tudtam helyreállítani, hogy Monitor üzemmódban átírtam a veremmutatót (SP regiszter) \$ F8-ra, így ismét kilistázhattam a programot, de a GOTO-k paraméterei átszámóztak, és csak hosszas munkával kaptam vissza.

IFJ. MONFERA RÓBERT

## Helyreigazítás

Sajnálatos módon az 1986/10. számunkban közölt Programtitkosítás című cikkbe hiba csúszott. A P—3 helyfoglaló program leírása említi a  $\theta$ -ás utasítássor szerepét. A program listájáról azonban ez a sor lemaradt:

$\theta$  GOTO 2

Az utasítás szerepe kettős: egyrészt átlépi az 1-es utasítássort, ahol a program lefutása után már nem szabályos BASIC-kódok lesznek, másrészt a 150-es sor csak így tudja az 1-es sor REM-je utáni területre tölni a gépi kódu rutint.

A szerző elnézést kér mindazoktól, akik sikertelenül próbálkoztak a programmal. Kéri, ne adják fel; ezzel a kis kiegészítéssel a várt eredmény nem fog elmaradni.

A lap 1986. októberi számában megjelent „Ki ad magyarázatot?” kérdésre válaszolok. A felvetett jelenségek magyarázata a következő.

1. A jelenséget a Primo BASIC interpreterének programhibája: egy állapotjelző helytelen kezelése okozza. Ez a hibát a Primo—B típusnál már kijavították. Az INPUT, INPUT\* és a READ utasítások ugyanis csak a bemenő adatok forrásában (billentyűzet, magnó, DATA sor) különböznek egymástól. Ezért célszerű az utasításokat egy olyan rutinnal kezelni, amely ezt a különbséget figyelembe tudja venni. Ehhez viszont valahogy jelezni kell, hogy melyik utasítás hajtódik végre. Erre két bájttal szolgál. Az egyikből kiderül, hogy READ vagy INPUT utasítás zajlik-e: tartalma INPUT esetén zérus, READ utasításnál nem zérus. A bájttal a 16606 címen található. A másik bájttal (címe 16553) azt jelzi, hogy az INPUT végrehajtásakor a billentyűzetről vagy a magnóról — pontosabban a magnópufferből — kell-e olvasni. Magnós értéke zérus, a billentyűzetről pedig nem zérus.

A READ utasítás egy DATA mutatót használ, amely megadja, hogy a következő adat melyik címen található. Ettől a címtől kezdve beolvassa a szükséges adatokat, majd a mutató új értékét tárolja a következő beolvasáshoz. Nos, a hiba itt következik be!

Az új érték tárolása ugyanis attól függ, hogy a 16553 cím tartalma 0-e. Ha igen, akkor nem tárolja az új értéket. Ezért tűnik úgy, mintha a READ után RESTORE lenne, mert minden READ utasítás ugyanarról a címről olvassa az adatot, hisz a mutató nem kapja meg az új értéket. Az eddigiekből következik, hogy a hiba nemcsak szöveges változóknál, hanem mindig jelentkezik.

Ezt a 16553 című bájtot a READ utasítás nem állítja be magának zérustól különböző értékre. INPUT\* utasításnál viszont értéke 0, és az INPUT\* sem változtatja meg az utasítás kezelése után. A hiba tehát nem a felhasználó BASIC programjában van, de szerencsére könnyen kivédhető. Amikor INPUT\* után READ-et kérünk a programban, akkor a READ előtt adjunk ki egy POKE 16553,1: utasítást. Ezután a cikk a—c pontjában írtakra nincs szükség.

2. A problémát az adatok felvétele és beolvasása közötti időzítéskülönbség okozza. A folyamatot példán mutatom be. Van mondjuk egy „A” nevű, 5000 bájtos numerikus tömbünk. Tipusa közbömbös. Ha ezt a tömböt a fájl megnyitása után a FOR I=1 TO 5000: PRINT# A(I): NEXT: CLOSE: ciklussal vesszük fel, akkor a következő zajlik. A PRINT# utasítás egy 256 bájtos pufferbe írja az adatokat addig, míg a pufferbe nem telik. Ez gyorsan bekövetkezik. Ekkor a puffer tartalmát egy blokkban felveszi a magnóra, és folytatja a most már üres pufferbe a kiírást. Addig ismétlődik a dolog, míg az adatok el nem fogynak. A fájl lezáró utasítás a még pufferben levő részt is felveszi.

Ezáltal lesz egy valahány blokkból álló felvételnél, ahol a blokkok között alig észrevehető szünetek vannak. Hiba akkor keletkezik, ha a visszaolvasás nem ugyanolyan ciklussal halad, mint a felvétel, mivel a Primo nem tudja vezérelni a magnót. Ha beolvasáskor például a beolvasott adattal valamilyen műveletet is végzünk, akkor egy adat feldolgozása hosszabb ideig tart, mint a felvétele. Ezáltal a program nem őríti ki a puffert arra az időre, mikor a magnóról már a következő blokk adatai jönnek.

Így ez a blokk elvész, hiszen a gép nem akarja beolvasni. Nem biztos, hogy az elvesztés rögtön hibaüzenetet okoz. Az adatfájl végénél viszont hiányozni fognak adatok, és ez már FD ERROR jelzést eredményez.

Tapasztalatom szerint még ugyanolyan ciklussal visszaolvasásnál is ajánlatos a biztonság kedvéért egy rövid, üres FOR ciklust tenni a PRINT# utasítás után a kiírásnál. Ha a visszaolvasást más módon végezzük, mint a felvételt, akkor a FOR ciklust olyan hosszúra válasszuk, hogy a felvételen az egyes blokkok között 2-3 másodperces szünetek legyenek. Ezekben a szünetekben kényelmesen megállíthatjuk a magnót.

E módszer szépséghibája, hogy nem tudhatjuk, a magnót mikor kell újraindítani. Én ez úgy oldottam meg, hogy a kapcsolási rajz X4 mezőjében levő T2 tranzisztort a szükséges ellenállásokkal beépítettem a gépbe, és egy LED-et tettem rá. Ez mindig világít, ha a gép olvasni akar.

Enyhén szólva nem szép a gyártótól, hogy az ilyen hibákat nem publikálja!

A „Prima ötlet”-hez is van egy megjegyzésem. Mint a szerzők is írják: „elég kezdetleges”, bár több a semminél. A RESET gomb megnyomása azonban több problémát okoz, mint amit a cikkben említenek. Hatására ugyanis a régi képernyőterületen elhelyezkedő adatok — kellően nagy program esetén maga a program vagy az adatai — is kitörölődnek. Ez 6 kb-ajtnyi információ elvesztését jelenti, ami igen nagy bosszúságot és sok plusz munkát okozhat. Ezért javaslom, hogy aki elvégzi ezt a bővítést, az mindenképpen végezze el a szükséges EPROM-cserét is.

Ezzel és az előbbi LED-beépítéssel kapcsolatban is szívesen segíték mindenkinek.

SZABADOS ISTVÁN  
Bp., Szabadság u. 48/a 1028

Az alábbiak azoknak az érdeklődésre tarthatnak számot, akik ZX81 számítógépüket nem akarják más típusra kicserélni, de szeretnék a gépükön Spectrum BA-SIC-ben dolgozni, esetleg a Spectrum gazdag programkínálatából egyes gépi kódú programokat is — lehetőség szerint minimális munkaráfordítással — gépükön futtatni.

# A Spectrum operációs rendszere ZX81-en

Mint látni fogjuk, erre van lehetőség. Természetesen a gép nem fog tudni színes képet szolgáltatni, hangja sem lesz, de ettől eltekintve, meglehetősen hasonlóságot fog mutatni a Spectrumhoz, beleértve a finomgrafikát is. Mindezt azonban nem kapjuk ingyen: elég sokat kell dolgozni, annál többet, minél nagyobb hasonlóságot akarunk megvalósítani a két gép között. A munkához segédesszközök is kellenek: valamilyen monitor-disassembler a ZX81-re, esetleg assembler is; jó, ha van egy ZX81 ROM-listánk, magyarázatokkal (12); elengedhetetlen a Spectrum ROM-lista és egyéb, a Spectrumra vonatkozó segédkönyvek (7, 8, 9, 10).

Némi költség szükséges a hardvermódosítások elvégzéséhez. A ZX81 kapcsolási vázlatát például (11) adja meg. Az ismertetendő legegyszerűbb változathoz is 64 k-s memóriabővítő kell, melyet ráadásul a finomgrafika megvalósításához át kell alakítani. Akinek Memopak 64 k-s bővítője van, az az átalakítást carcolóttá, huzal és forrasztópáka segítségével el tudja végezni. Az átalakításhoz (1) szakirodalmat ad segítséget. A nálunk nagy számban található, az Audio Computers 32 k-s memóriájáról mással bővítő is átalakítható 64 k-s-ra.

## Memóriabővítés

A memóriabővítővel szemben két követelmény van. Egyrészt a finomgrafika megvalósításához az RFSH jel alatt a RAM-ból olvasni kell tudnunk; másrészt a 0000H—3FFFH, eredetileg ROM-tartomány helyére a RAM ugyanezen tartományát kell tudni kapcsolni. Ebben a tartományban kell ugyanis majd a Spectrum átalakított, illetve kiegészített ROM programjának elhelyezkednie. Ha ráadásul ezt a részt írásvédetté tesszük a monitorprogram bekerülése után, akkor az esetleges „elszállások” után nem kell mindig megismételni a programbetöltést.

Mivel a két gép közötti alapvető különbség a képalkotásban van, tekintsük át röviden, hogyan történik ez az egyes esetekben. A ZX81-et igen olcsó gépnek szánták, ezért minél több feladatot igyekeztek szoftver úton megoldani, hogy a gép szerkezete minél egyszerűbb lehessen. Így a képalkotásban is jelentős szerepe van a processzornak. Ezt a feladatot azonban csak egyéb feladatok rovására tudja elvégezni; ennek következtében a „hasznos” programok lassabban futnak, mint akkor futnának, ha minden idejét rájuk tudná fordítani. A se-

bességcsökkenés jelentős. Az ún. SLOW üzemmódban, amikor állandóan van kép, csak kb. negyed sebességgel dolgozik a gép a FAST üzemmóddhoz képest, amikor viszont csak akkor van kép, ha a gép feladatát már elvégezte, illetve amikor billentyű megnyomására vár. Ezenfelül egyéb korlátozásokat is tudomásul kell vennünk. Gépi kódú program csak a 0000H—7FFFH tartományban futhat, bizonyos regisztereket (IX, AF) állandóan a képalkotás céljaira kell rendelkezésre bocsátanunk, a maszkolható és nem maszkolható megszakítást is ez használja.

Másként áll a dolog a Spectrumnál. Itt a képet harder úton állítjuk elő, a fenti korlátozások jórészt elesnek. A monitorprogramnak mindössze arról kell gondoskodnia, hogy a képinformációt tartalmazó RAM-rész — az ún. képernyőfájl — folyamatosan fel legyen töltve a megjeleníteni kívánt új információval. A képalkotás bizonyos feltételek mellett itt is sebességcsökkenést okoz, ez azonban nem túlzottan jelentős.

Feladatunkat most már úgy oldjuk meg, hogy a Spectrum monitorprogramot kiegészítjük olyan képalkotó programmal, mely a ZX81 adottságait felhasználva, a Spectrum képernyőfájl tartalmát meg tudja jeleníteni a képernyőn; a monitorprogram azon részeit, melyek megszakításokat és beviteli-kiviteli műveleteket tartalmaznak, szükség esetén átalakítjuk; végül az így módosított programot a RAM említett helyére juttatjuk, és átadjuk neki a vezérlést.

A „képernyőrutint” (1) alapján lehetett létrehozni. A cikkben majdnem minden lényeges gondolat megtalálható, mely célunk eléréséhez szükséges volt.

A járulékos programokat a Spectrum monitorprogram fel nem használt 386EH—3CFFFH területén helyezük el, itt csupa FFH található. A módosításokat úgy végezzük, hogy lehetőleg olyan cím, amelyet valamely, a Spectrumhoz kidolgozott program hívhat, ne változzék.

A gyakorlati használathoz igen lényeges, hogy a magnetofonról a programokat gyorsan tudjuk betölteni; a ZX81 eredeti SAVE—LOAD eljárása erre a célra alkalmatlan, mivel lassú és megbízhatatlan. A választás a c't folyóiratban közölt Supertape programra esett, mely a ZX81-re (2)-ben található. Ezzel a programmal 3600, illetve 7200 baud sebességgel rögzíthetünk programot szalagra.

Néhány tanács a Supertape program használatához. Célzerű lehet a szalagra

való felvétel szintjét növelni. Ez úgy történhet, hogy a ZX81-ben R29-et 1 Mohm-ról 10 kohm-ra változtatjuk, C12-t áthidaljuk, C11-et és R27-et eltávolítjuk. Az „Ear” bemenet érzékenyítése R34 eltávolításával történik (3). Ugyanekkor C10 értékét meg kellett növelni 0,33 µF-ra: ezt kis kapcsolóval célszerű kiiktathatónak tenni, mert az eredeti ZX81 impulzusalakhoz ez sok lehet. Bár az idézett cikk szerint a Supertape a közönséges magnetofonokon is jó eredményt ad, jobb eredményt lehet elérni, ha magnetofonunkat a digitális üzem követelményeinek megfelelően átalakítjuk (4). Fontos a mechanika, különösen a szalagotvábbító túske rezgésmentessége; jelentős javulást észleltünk, amikor ennek műanyag perselyét erősen viszkózus, hőálló zsírral kentük. Nagyon jó eredményt ad a Polimer C60-as szalag.

A Supertape a Spectrum is megvalósítható. Ha tehát mód van rá, készítsünk a Spectrum ROM-ról szalagra Supertape formátumú felvételt, s ez a ZX81-be bejuttatható. A másik lehetőség, hogy olyan programot állítsunk össze, mellyel az eredeti Spectrum-felvétel a ZX81-be tölthető. A következőkben ezt ismertetjük.

Előjáróban bemutatok egy segédprogramot (1. program), mellyel a szalagon levő program impulzusainak vagy szünetekinek hosszát tudjuk megállapítani. Jó szolgálatot tesz akkor is, amikor a felvétel-lejtszás minőségére vagyunk kíváncsiak.

A programmal 8000 H—BFFFH-ig folyamatosan feltöltjük a memóriahelyeket a szünet- vagy az impulzushosszra jellemző számmal. Rövid program esetén azt többször betöltjük egy a betöltési tartomány felső címét csökkentjük. A program indítása: FAST után RAND USR 16514.

Ha a szünetek helyett az impulzushosszokra vagyunk kíváncsiak, a 4088H és 4090H helyen levő utasításokat ellenkező értelműre változtatjuk. Ha katalógusból kikeressük az egyes utasításokhoz tartozó végrehajtási időket, akkor a tényleges szünet-, illetve impulzusidőket is kiszámíthatjuk. A ZX81 órajel-frekvenciája 3,25 MHz.

Egy JVC gyártmányú rádiómagnón egy eredeti, digitális magnón felvett Spectrum-program a szünethosszokra a következő eredményeket adta:

Bevezető fűty: 3A—3F  
Szinkronjel: 10  
Program 0 szint 15—18  
1 szint 2E—35

Az impulzus-, illetve szünethosszak tehát szórnak. A szórás nagysága az adatrögzítő

100 REM	4082	210080	LD HL,8000	;a betoltes kezdeti cim
110 REM	4085	0BFE	IN A,(FE)	;lekerdezzuk az EAR bemenetet
120 REM	4087	17	RLA	;a 7.bit-a bemszint-Carrybe
130 REM	4088	30FB	JR C,4085	;ha a szint magas,vissza
140 REM	408A	0E00	LD C,00	;szunet kezd.: hossztarolo regiszter 0
152 REM	408C	0C	INC C	;hossztarolo regiszter inkrem
154 REM				
160 REM	408D	0BFE	IN A,(FE)	;EAR bemenet?
170 REM	408F	17	RLA	;allapot C-be
180 REM	4090	30FA	JR NC,408C	;ha alacsony a szint,vissza
190 REM	4091	79	LD A,C	;A-ba a hosszjellemzot
200 REM	4092	77	LD(HL),A	; (HL)-be
210 REM	4093	23	INC HL	;kovetkezo tarolohely
220 REM	4094	3EBF	LD A,BF	;BF00-ig taroljuk a hosszakat
230 REM	4096	BC	CP H	;elertuk mar?
240 REM	4097	CS	RET Z	;ha igen,vege
250 REM	4098	18EA	JR 4085	;ha nem,vissza az elejere

**1. program**

100 REM	1	REM	(IDE JON A 3.PROGRAM 182 BAJTJA)
110 REM	2	PRINT	"A BETOLTES KEZDETI CIM:"
120 REM	3	INPUT	A
130 REM	4	PRINT	A
140 REM	5	LET	X=INT(A/256)
150 REM	6	LET	Y=A-256*X
160 REM	7	POKE	16515,X
170 REM	8	POKE	16514,Y
180 REM	9	POKE	16517,X
190 REM	10	POKE	16516,Y
200 REM	11	LET	A=A+18
210 REM	12	LET	X=INT(A/256)
220 REM	13	LET	Y=A-256*X
230 REM	14	POKE	16519,X
240 REM	15	POKE	16518,Y
250 REM	16	RAND	USR 16520

**2. program**

100 REM	4088	CD230F	CALL 0F23	;FAST uzemmod
110 REM	408B	1E00	LD E,0	;szamlatlo reg.
120 REM	408D	CD460F	CALL 0F46	;BREAK-proba
130 REM	4090	D2A603	JP NC,03A6	;ha BREAK,Dhiba
140 REM	4093	1C	INC E	;szaml.novelese
150 REM	4094	03FF	OUT(FF),A	;magas szint a TV kimeneten
160 REM	4096	7B	LD A,E	;A-ba a szaml.
170 REM	4097	FEFF	CP FF	;elertuk mar FF-t?
180 REM	4099	300E	JR NC,40A9	;ha elertuk,va-ironjunk a szinkronjelre
190 REM	409B	1600	LD D,00	;a D szaml.=0
200 REM	409D	01004A	LD BC,4A00	;a futtyaszunet
210 REM	40A0	CD0541	CALL 4105	;A futtyaszunet hossza
220 REM	40A3	FE01	CP 01	; <4A?
230 REM	40A5	204	JR NZ,40B8	;ha nem,vissza
240 REM	40A7	13E4	JR 40B0	;ha nem,vissza
250 REM	40A9	03FF	OUT(FF),A	;fecho a kepen
260 REM	40AB	1600	LD D,00	;D=0
270 REM	40AD	010022	LD BC,2200	;a szinkronjel-hossz =22
280 REM	4088	CD0541	CALL 4105	;a szinkronszunetet keressuk
290 REM	4083	FE01	CP 01	;elertuk mar?
300 REM	4085	20F2	JR NZ,40A9	;ha nem,tovabb keresunk
310 REM	40B7	2A8240	LD HL(4082)	;adatterakas itt kezd
320 REM	40BA	1E00	LD E,00	;iparitasellenor
330 REM	40BC	03FF	OUT(FF),A	;zo gyujtoreg
340 REM	40BE	010025	LD BC,2500	;fecho a kepen
350 REM	40C1	CD1941	CALL 4119	;hosszu es rov. szunet kozt
360 REM	40C4	7A	LD A,D	;bajtkomplemens
370 REM	40C5	2F	CPL	;A-ba D
380 REM	40C6	77	LD(HL),A	;a bajt meghat.
				;a bajt betolt.

390 REM	40C7	AB	XOR E	;paritaskepzes
400 REM	40C8	5F	LD E,A	;fezt a gyujtoregiszterbe
410 REM	40C9	05	PUSH DE	;DE stackre
420 REM	40CA	EB	EX DE,HL	;DE-be a betolites helyet
430 REM	40CB	2A8640	LD HL(4086)	;HL-be a file neve
440 REM	40CE	A7	AND A	;vege
450 REM	40CF	ED52	SBC HL,DE	;Carry=0
460 REM	40D1	EB	EX DE,HL	;HL-DE
470 REM	40D2	D1	POP DE	;HL-be a betolites helyet
480 REM	40D3	2804	JR Z,40D9	;DE vissza
490 REM	40D5	23	INC HL	;ha vege,kontroll
500 REM	40D6	C3BC40	JP 00BC	;ha nincs vege
510 REM	40D9	7B	LD A,E	;kov.bajjt vissza
520 REM	40DA	A7	AND A	;A-ba a parit.
530 REM	40DB	C2A603	JP NZ,03A6	;flagallitas
540 REM	40DE	E5	PUSH HL	;ha nem zerus.
550 REM	40DF	2A8240	LD HL(4082)	;D hiba
560 REM	40E2	ED584640	LD DE(4084)	;HL stackre
570 REM	40E6	A7	AND A	;HL=(4082)
580 REM	40E7	ED52	SBC HL,DE	;DE=(4084)
590 REM	40E9	E1	POP HL	;Carry=0
600 REM	40EA	2048	JR NZ,4134	;HL-DE
602 REM	40EC	23	INC HL	;HL vissza
604 REM	40ED	228240	LD(4082),HL	;ha (4082)<
606 REM	40F0	110700	LD DE,0007	; (4084),kesz
608 REM	40F3	A7	AND A	;kov.bajjt helyet
610 REM	40F4	ED52	SBC HL,DE	;most (4082)-be
612 REM	40F6	5E	LD E,(HL)	;felt.leszok
614 REM	40F7	23	INC HL	;a filehossz
616 REM	40F8	56	LD D,(HL)	;helgenek meg-
618 REM	40F9	2A8240	LD HL(4082)	;hatarozasa
620 REM	40FC	ED5A	ADC HL,DE	;flagallitas
622 REM	40FE	23	INC HL	;HL a low bajt-
623 REM	40FF	228640	LD(4086),HL	;ira mutat
624 REM	4102	C38840	JP 4088	;E-be a file-
626 REM	4105	0BFE	IN A,(FE)	;hossz l.bajjt
628 REM	4107	17	RLA	;HL a h.bajtra
630 REM	4108	30FB	JR C,4105	;mutat
632 REM	410A	0E00	LD C,00	;D-be a h.bajjt
634 REM	410C	0C	INC C	;HL-be a file-kezdet
636 REM	410D	0BFE	IN A,(FE)	;CY=0,HL a file
638 REM	410F	17	RLA	;vege
640 REM	4110	D20C41	JP NC,410C	;a paritasbajjt
642 REM	4113	79	LD A,C	;HL helyet
644 REM	4114	90	SUB B	;vissza
646 REM	4115	CB12	RL D	;EAR=?
648 REM	4117	7A	LD A,D	;EAR=?
650 REM	4118	C9	RET	;ha szint CY-be
652 REM	4119	1600	LD D,00	;a meg ala-
654 REM	411B	CD0541	CALL 4105	;acsony,vissza
656 REM	411E	CD0541	CALL 4105	;ha valtott,
658 REM	4121	CD0541	CALL 4105	;A-ba C-t
660 REM	4124	CD0541	CALL 4105	;levonjuk B-t
662 REM	4127	CD0541	CALL 4105	;ha ACB,CY=1,
664 REM	412A	CD0541	CALL 4105	;int. D-be
666 REM	412D	CD0541	CALL 4105	;A-ba D-t
668 REM	4130	CD0541	CALL 4105	;visszaszok
670 REM	4133	C9	RET	;a bajt gyujto-
672 REM	4134	CD200F	CALL 0F20	;regisztere
674 REM	4137	C9	RET	

**3. program**

100 REM	4082	CDE702	CALL 02E7	;FAST
102 REM	4085	3E7F	LD A,7F	;
104 REM	4087	0BFE	IN A,(FE)	;BREAK?
106 REM	4089	1F	RRR	;eredmeny CY-be
108 REM	408A	03FF	OUT(FF),A	;fecho a kepen
110 REM	408C	30F7	JR C,4085	;icig meg nem
112 REM				;nyuotuk BREAK-
114 REM				;tet,vissza
116 REM	408E	210080	LD HL,8000	;formas kezdet



és visszadó lánc jóságának függvénye. Az adatokat statisztikailag értékelve bizonyára érdekes tapasztalatokat lehetne szerezni a magnókra, szalagminőségekre vonatkozóan.

A fenti példát tekintve látható, hogy a fűtő-, a szinkron- és a programszintek jól elkülöníthetők. Csakis ilyen esetben remélhetjük, hogy felvételi problémánk nem lesz.

A következők program a Spectrum ROM-program ZX81-be töltésére szolgál. A gépi kódú részt az 1. REM sorban helyeztük el. A BASIC részt a 2. program, a gépi kódú részt a 3. program tartalmazza.

Például a Spectrum ROM-program átvételénél, ha a 0000H címnek a 8000H címet akarjuk megfeleltetni, a következőképpen járunk el:

```
RUN /Newline
32748 /Newline
```

Megindítjuk a magnót. A képernyőn vékony fekete vízszintes vonalak láthatók. A beolvasás 0/16 üzenettel áll le. 32748-tól a Spectrum-program fejlcét láthatjuk, a tényleges ROM-tartalom 8000H—BFFFH-ig tart.

## A Spectrum ROM módosítása

Most már gépünkben van a Spectrum ROM-tartalma, megkezdhetjük annak módosítását. Előbb azonban azt a programot ismertethetjük, amelynek segítségével a 8000H—BFFFH között elhelyezkedő, módosított Spectrum ROM-programot a 0000H—3FFFH közötti ROM-terület helyett beiktatott RAM-területre másolhatjuk.

A következőkben — feltéve, hogy a Spectrum-programunknál minden szükséges változtatást végrehajtottunk —, a Spectrum rendszerre való áttérés az említett program segítségével a következő lépésekben megy végbe:

— Betöltjük a gyorstöltőprogramot ZX81 módszerrel.

— Betöltjük az átmásolóprogramot, most és a továbbiakban már gyorstöltéssel dolgozva.

— Betöltjük a módosított Spectrum ROM programot.

— A RAND USR 16514/Newline-nál megindítjuk az átmásolóprogramot: ez először csak egy zárt programhurokban kering, a képernyőre vízszintes mintát rajzol. A programszámláló tartalma mindig a 4000—7FFF közötti részen van, tehát a 0000H—3FFFH szegmenset átkapcsolhatjuk: ROM helyett RAM.

— A BREAK gomb megnyomásával megtörténik az átmásolás (rövid megnyomás legyen!).

— Ismét zárt hurokba kerülünk, a képernyőn az előbbi minta; ekkor azt az átkapcsolást lehet végrehajtani, mellyel lehetővé tesszük, hogy az egész RAM-tartományban futhassanak gépi programok (ld. később).

— Újabb BREAK: a jól ismert © 1982 Sinclair Research Ltd. üzenet jelenik meg, gépünkön a Spectrum BASIC-ben dolgozhatunk.

Az egész áttérési művelet mindenképp kb. két és fél percet vesz igénybe. Az átmásoló programot (4. program) egy REM sorban helyezhetjük el.

Most már ráterhetünk a Spectrum ROM program módosítására. Természetesen a képképződés kezdjük. Két lehetőség között választhatunk: vagy a ZX81 SLOW vagy a FAST mintájára építhetjük föl a képképződést. Az előbbi választottuk, annak ellenére, hogy így a gép lassabb, és az NMI-t is igénybe vesszük. Viszont a kép nem ugrik a billentyűk megnyomásakor. A programfutást az NMI generátor kikapcsolásával gyorsíthatjuk. Ilyenkor természetesen nincs kép [lásd (5)-öt].

A képképző program megértéséhez ismerni kell bizonyos mértékig a ZX81 lelkivilágát. Részletes leírása (6)-ban található. A legfontosabb tudnivalókat röviden áttekintjük.

Mint ismeretes, a televíziós kép sorokból áll, és az új sort sorszinkronimpulzus indítja. Ezeket a ZX81-ben frekvenciaszótt állítja elő az órajelből: minden 207 órajel után ad ki egy impulzust. Ez kicsit nagyobb frekvenciát is ki, mint a szabványos 15625 Hz, mely 208-as osztó esetén állna elő.

A sorszinkronimpulzusokat egy OUT(FE),A utasítással rá lehet kapcsolni a processzor NMI bemenetére és

OUT(FD),A utasítással önnan lekapcsolni. Lekapcsolt NMI esetén egy IN A,(FE) utasítás a tv-kimeneten 0 szintet állít elő, az OUT(FF),A pedig 1-et. Ezek segítségével állítjuk elő a képszinkronjelet.

A videokimeneten levő képinformáció a hasznos kép alatti és feletti üres sorokból, a hasznos kép sorából, képként képszinkron-, soronként sorszinkronjelből áll. A hasznos sornak is van baloldalt és jobboldalt egy üres része.

Miközben a számítógép valamely programot dolgoz fel, minden egyes NMI beérkezésekor megvizsgálja — a ZX81 ROM 0066H-nál kezdődő rutinja segítségével —, hogy elérkezett-e már a képiadás vagy a képszinkron ideje. Erre a célja az AF regiszter, mint számláló használja fel. Ha az idő elérkezett, végrehajtja vagy a képiadást, vagy a képszinkronadást, mely utóbbi alatt a tasztatúrát is lekerdezi.

A képiadás a részleteket nem tekintve, a következőképpen zajlik le. Végrehajtunk egy ugrást egy, a képernyőfájl első helyénél 8000H-re nagyobb memóriahelyre, vagyis az A15 értéke 1. A nem teljes memóriáidőoldás miatt azonban ez a képernyőfájl első helyére történik. Két dologra emlékeztetünk: a képernyőfájl az ábrázolandó karakterek kódját tartalmazza, másrészt „ágras” utasításlelívást jelent az ugrási helyről, és ekkor az MJ jel aktív. 64 k-s bővítőknél a nem teljes címdeklóolás így — a C000H—FFFFH tartományban — csak MI aktív állapotában kell hogy megvalósuljon. Így is készülnek.

Ott tartottunk, hogy a képernyőfájl első memóriahelyének tartalma megjelenik az adatbuszon, miközben A15 értéke 1. Ez utóbbi hatására az ULA a következőt teszi. Ha az adatbusz D6 biteje 0, akkor a processzornak NOP utasítást továbbít. Minden „ábrázolható” karakter kódjának D6 biteje 0. Így a processzor NOP utasítást hajt végre, az ULA pedig megjegyzi a képernyőfájlból kiolvasott karakterkódot. Amikor a NOP-on belül a frissítés ciklushoz érünk, a processzor a címre az I és R regiszter tartalmát teszi rá. Az I regiszter tartalma a karaktergenerátor táblázat kezdeti címének magas bájtja: I EH.

```
118 REM 4091 11000B LD DE,0000 ;cel kezdet
120 REM 4094 010040 LD BC,4000 ;blokkhossz
122 REM 4097 ED0B LDIR ;masolas
124 REM 4099 3E7F LD A,7F ;
126 REM 409B DBFE IN A,(FE) ;BREAK?
128 REM 409D 1F RRA ;az eredmény
130 REM ;CY-be
132 REM 409E D3FF OUT(FF),A ;echo a kepen
134 REM 40A0 38F7 JR C,4099 ;ha nincs BREAK
136 REM ;visza
138 REM 40A2 C3000B JP 0000 ;atadjuk a ve-
140 REM ;zerlest a SPEK
142 REM ;TRUM monitor-
144 REM ;programnak
```

4. program

```
118 REM 0038 C5 PUSH BC ;a fo regiszte-
120 REM 003C D5 PUSH DE ;istackra
122 REM 003D E5 PUSH HL ;
124 REM 003E 2A05C LD HL(5C00) ;HL-be a kep-
126 REM ;rva képszink-
128 REM ;ron rutin kez-
130 REM ;doctme
132 REM ;
134 REM 0041 76 HALT ;megszakitas-
136 REM ;ra varunk
138 REM 0042 D3FD OUT(FD),A ;az NMI genera-
140 REM ;tor lekapcs.
142 REM 0044 E9 JP (HL) ;ugras a kezdo
144 REM ;cimre
```

5. program

```
100 REM 0066 FS PUSH AF ;AF stackra
102 REM 0067 FD3447 INC(I+47) ;a szamlalot
104 REM ;mozditjuk
106 REM 006A FA0F0B JP M,006F ;tovabb szamol
108 REM 006D 28CC JR Z,003B ;kep vagy kep-
110 REM ;szinkron kia-
112 REM ;das következik
114 REM 006F F1 POP AF ;stackrol AF
116 REM 0070 C9 RET ;visza
```

```
102 REM 3800 0000000000000000 ;32 db NOP uta-
104 REM 3808 0000000000000000 ;sitas, a pszeu-
106 REM 3810 0000000000000000 ;do videoros
108 REM 3818 0000000000000000 ;
110 REM 38A0 C30339 JP 3903 ;viszsa a prog-
112 REM ;ramba
114 REM 38C0 0000000000000000 ;
116 REM 38C8 0000000000000000 ;32 db NOP uta-
```

118 REM	38C8	0000000000000000	;sitás a 2.pszé
120 REM	38D8	0000000000000000	;udosor
122 REM	38D8	0000000000000000	;
124 REM	38E8	C31539 JP 3915	;vissza a prog-
126 REM			;ramba
128 REM	38E3	00 NOP	;
130 REM	38E4	ED73805CLD(5C80),SP	;iatmenetileg ta-
132 REM			;roljuk SP-t
134 REM	38E8	31843A LD SP(3A84);	;az IR reg.tab-
136 REM			;lazarata mutat
138 REM	38E8	0603 LD B,03	;
140 REM	38ED	10FE DJNZ 38ED	;idokitolto
142 REM	38EF	010707 LD 8C,0707	;B=07,C=07
144 REM	38F2	210018 LD HL,1800	;H-ba a karak-
146 REM			;tersorok szama
148 REM	38F5	00 NOP	;idokitolto
150 REM	38F6	D1 POP DE	;IR aktualis
152 REM			;teke
154 REM	38F7	00 NOP	;idokitolto
156 REM	38F8	7A LD A,D	;A-ba D
158 REM	38F9	ED47 LD I,A	;I-be az aktu-
160 REM			;alis erteke
162 REM	38F8	3E08 LD A,08	;idokitolto
164 REM	38FD	78 LD A,E	;A-ba E
166 REM	38FE	ED4F LD R,A	;IR akt.ert.
168 REM	3900	C38088 JP 8880	;a pontsor kia-
170 REM			;idasa (allitott
172 REM			;A15)
174 REM	3983	14 INC D	;a kov.pontsor
176 REM	3984	7A LD A,D	;A-ba D
178 REM	3985	05 DEC B	;a 7.sort kiad-
180 REM			;tuk mar?
182 REM	3986	C2F738 JP NZ,38F7	;ha nem,vissza
184 REM	3989	41 LD B,C	;B kezdoteke
186 REM			;=07
188 REM	398A	25 DEC H	;a sorszamalo
190 REM			;csokkentese
192 REM	3988	ED47 LD I,A	;I-be A
194 REM	398D	3E08 LD A,08	;idokitolto
196 REM	398F	78 LD A,E	;A-ba E
198 REM	3910	ED4F LD R,A	;R-be az akt.
200 REM			;erteke
202 REM	3912	C3C888 JP 8880	;a 8.sor kidasa
204 REM			;allitott A15
206 REM	3915	20DF JR NZ,38F5	;ha meg nem ad-
208 REM			;tuk ki a kepet
210 REM			;vissza
212 REM	3917	ED78805CLD SP(5C80)	;SP eredeti ert-
214 REM			;teke
216 REM	3918	FD3647C9LD(IY+47)C9	;az ures sorok
218 REM			;szamaloja
220 REM	391F	213339 LD HL,3933	;az uj vissza-
222 REM			;steresi cim
224 REM	3922	22805C LD(5C80),HL	;rect(5C80)-ba
226 REM	3925	088000	;
228 REM	3928	D023 INC IX	;idokitolto
230 REM	392A	D028 DEC IX	;
232 REM	392C	D3FE OUT(IE),A	;ibe kapcsoljuk
234 REM			;az NMI gen.-t
236 REM	392E	E1 POP HL	;a fo regisztr-
238 REM	392F	D1 POP DE	;rek vissza
240 REM	3930	C1 POP BC	;
242 REM	3931	F1 POP AF	;
244 REM	3932	C9 RET	;a kep kesz,
246 REM			;vissza
248 REM	3933	2A785C LD HL(5C75);	;kepvaltasok
250 REM			;szama
252 REM	3936	23 INC HL	;inoveljuk
254 REM	3937	22785C LD(5C78),HL	;uj erteke

256 REM	393A	7C LD A,H	;ertek a 0000
258 REM	3938	85 OR L	;ertekel?
260 REM	393C	2006 JR NZ,3944	;ha nem,ugrunk
262 REM	393E	FD3440 INC(5C7A),	;FRAMES 3 nov.
264 REM	3941	C34839 JP 3948	;ugras
266 REM	3944	00000000000000	;7 db NDP(kesl)
268 REM	3948	0000	;keslettetes
270 REM	3948	E5 PUSH HL	;
272 REM	394E	E1 POP HL	;
274 REM	394F	23 INC HL	;
276 REM	3950	00 NOP	;keslettetes
278 REM	3951	0BFE IN A,(FE)	;kepszinkron
280 REM			;kezd
282 REM	3953	00 NOP	;idokitolto
284 REM	3954	0660 LD B,60	;
286 REM	3956	10FE DJNZ 3956	;keslettetes
288 REM	3958	D3FF OUT(FF),A	;kepszinkron
290 REM			;vege
292 REM	395A	0603 LD B,03	;
294 REM	395C	10FE DJNZ 395C	;keslettetes
296 REM	395E	FD3647C9LD(IY+47)C9	;az ures sorok
298 REM			;szamaloja
300 REM	3962	21E438 LD HL,38E4	;az uj vissza-
302 REM			;steresi cim
304 REM	3965	22805C LD(5C80),HL	;taroljuk
306 REM	3968	D3FE OUT(IE),A	;ibe kapcsoljuk
308 REM			;a tszaturu
310 REM	396A	C8FB02 CALL 02BF	;lekerdezes
312 REM			;
314 REM	396D	E1 POP HL	;
316 REM	396E	D1 POP DE	;a fo registre-
318 REM	396F	C1 POP BC	;rek vissza
320 REM	3970	F1 POP AF	;
322 REM	3971	C9 RET	;vissza a fo-
324 REM			;iprogramba
326 REM	3AB4	7F401F403F405F40	;az IR reg.
328 REM	3ABC	FF409F408F40DF40	;kezdote
330 REM	3AC4	7F481F483F485F48	;keinek tab-
332 REM	3ACC	FF489F488F48DF48	;lazata
334 REM	3AD4	7F501F503F505F50	;
336 REM	3ADC	FF509F508F50DF50	;

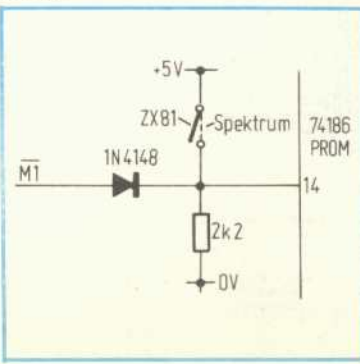
6. program

7. program

100 REM	0000	D3FD OUT(IE),A	;initializalas
102 REM	0002	AF XOR A	;897F folott
104 REM	0003	117FB8 LD DE,887F	;nem toroljuk
106 REM	0006	1808 JR 0013	;a tarat, mert
108 REM			;a pseudosorok
110 REM			;ott vannak
112 REM	0013	C3C811 JP 11CB	;
114 REM	0030	C9 RET	;az INT-ot nem
116 REM			;hasznaljuk
118 REM	03E1	0000 NOP,NOP	;Beeper-ben
120 REM	03FC	00 NOP	;a magnetofonke
122 REM	04C6	21C00F LD HL,0FC8	;zelo rutinok
124 REM	04CD	214C86 LD HL,064C	;modositasa
126 REM	04D4	76 HALT	;
128 REM	04D5	D3FD OUT(IE),A	;
130 REM	04D7	00 NOP	;
132 REM	04D8	10FE DJNZ 04D8	;
134 REM	04DA	D3FF OUT(FF),A	;

Közben az ULA a megjegyzett karakterkódból és egy három bites számláló tartalmából kiindulva a ROM címbusz alsó 9 vonalát módosítja a processzor által képezett címhez képest, úgy, hogy az az ábrázolandó karakter megfelelő sorának „bitmintáját” tartalmazza. A ROM számára az ULA által kiadott cím a döntő; az alsó 9 címvonal a processzorétól ellenállásokkal el van választva. Vagyis az ULA a „bitmintát” kapja meg az adatvonalakon, beteszi egy shiftregiszterbe, és a következő NOP alatt a képernyőre kiadja. És ez így megy tovább, míg a képernyőfájlban egy olyan utasítást nem találunk, melynek D6 biteje 1 (például a HALT, melynek kódja 76H). Az ilyen utasításokat az ULA eredeti alakban engedi a processzorhoz. A HALT után az R regiszter nemsokára nullázódik; legalábbis a

1. ábra



6-bites bitje, melynek adatvonala a processzor INT bemenetével van összekötve. A bekövetkező maszkolható megszakítás a következő sor kiadását készíti elő. Az INT-ot Sinclair bizonyára csak azért használta fel, mert az 1 k-s gépben a képernyőfájl „komprimálva” van. Mi az INT-ot nem program használni, ezért ennek részleteibe nem is megyünk bele.

Az előzőekből a mi képalakító programunk megértéséhez a következőket szűrjük le:

- az NMI jelet OUT(IE),A utasítás kapcsolja le a processzoról,
- az NMI jelet OUT(IE),A utasítás kapcsolja rá a processzorra,
- lekapszint NMI esetén az IN A,(FE) a videokimenetet 0-ra, OUT(FF),A pedig 1-re állítja,

```

146 REM      04DC 00      NOP
148 REM      04DD 0699   LD B,99
150 REM      04DF 10FE   DJNZ 04DF
152 REM      04E1 3E7F   LD A,7F
154 REM      04E3 DBFE   IN A,(FE)
156 REM      04E5 C3D839  JP 390B
158 REM      04E8 062A   LD B,2A
160 REM      04EA 10FE   DJNZ 04EA
162 REM      04EC D3FF   OUT(FF),A
164 REM      04EE 3E7F   LD A,7F
166 REM      04F0 0633   LD B,33
168 REM      04F2 10FF   DJNZ 04F2
170 REM      04F4 DBFE   IN A,(FE)
172 REM      04F6 C37239  JP 3972
174 REM      04F9 00000000
176 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
178 REM      0508 0000
180 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
182 REM      050F C30505  JP 0505
184 REM      0512 0000
186 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
188 REM      0518 063D   LD B,3D
190 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
192 REM      051C D3FF   OUT(FF),A
194 REM      051E 0639   LD B,39
196 REM      0520 00      NOP
198 REM      0521 C8      RET Z
200 REM      0522 C3F239  JP 39F2
202 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
204 REM      052D 062E   LD B,2E
206 REM      052F 062E   LD B,2E
208 REM      0531 7A     LD A,D
210 REM      0532 3C     INC A
212 REM      0533 C2FE04  JR NZ,04FE
214 REM      0536 0636   LD B,36
216 REM      0538 10FE   DJNZ 0538
218 REM      053A C9     RET
220 REM      053B 00000000
222 REM      053F F5     PUSH AF
224 REM      0540 3E7F   LD A,7F
226 REM      0542 DBFE   IN A,(FE)
228 REM      0544 1F     RRA
230 REM      0545 D3FF   OUT(FF),A
232 REM      0547 D3FE   OUT(FE),A
234 REM      0549 00000000000000
236 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
238 REM      0558 0000
240 REM      055A 15     DEC D
242 REM      055B 76     HALT
244 REM      055C D3FD   OUT(FD),A
246 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
248 REM      0565 E640   AND 40
250 REM      0567 0000
252 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
254 REM      059F 00000000
256 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
258 REM      05F6 E640   AND 40
260 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
262 REM      05FD 00000000
264 REM      0601 D3FF   OUT(FF),A
266 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
268 REM      091E 00
270 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
272 REM      098F C0A3A3  CALL 3A0A
274 REM      0992 0000
276 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
278 REM      1187 C0BF39  CALL 39BF ia NEM-ban
280 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

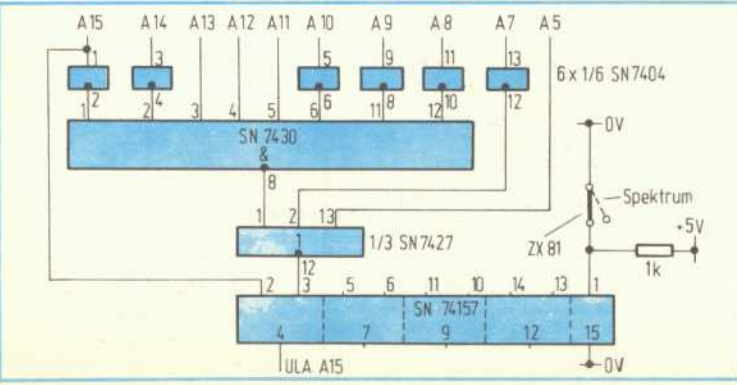
```

```

282 REM      11CC 000000000000
284 REM      1102 F3     DI
286 REM      1103 ED56   IM 1
288 REM      1105 3E3F   LD A,3F
290 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
292 REM      122E FD213A5CLD IY,(5C3A)
294 REM      1232 C0A638  CALL 386E
296 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
298 REM      18D2 3684   LD (HL),04 ivillogo karak
300 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
302 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
304 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
306 REM      229B 0000      iBORDER-ben
308 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
310 REM      33FB 117738  LD DE,3077 ia konstansok
312 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
314 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
316 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
318 REM      386E 21E438  LD HL,38E4
320 REM      3871 22B05C  LD(5C00),HL
322 REM      3874 D3FE   OUT(FE),A
324 REM      3876 C9     RET
326 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
328 REM      3972 08      EX AF,AF'
330 REM      3973 068A   LD B,0A
332 REM      3975 10FE   DJNZ 3975
334 REM      3977 062C   LD B,2C
336 REM      3979 6F   LD L,A
338 REM      397A 00      NOP
340 REM      397B C30785  JP 0507
342 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
344 REM      398F 3E7F   LD A,7F
346 REM      39C1 76     HALT
348 REM      39C2 D3FD   OUT(FD),A
350 REM      39C4 C9     RET
352 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
354 REM      390B 0698   LD B,98
356 REM      390D 20   DEC L
358 REM      390E C2D804  JP NZ,0408
360 REM      39E1 06A2   LD B,A2
362 REM      39E3 C0     RET NZ
364 REM      39E4 25     DEC H
366 REM      39E5 F2D804  JP P,0408
368 REM      39E8 C3E804  JP 04E8
370 REM      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
372 REM      39F2 10FE   DJNZ 39F2
374 REM      39F4 D2FE39  JP NC,39FE
376 REM      39F7 063B   LD B,3B
378 REM      39F9 0638   LD B,38
380 REM      39FB 00      NOP
382 REM      39FC 10FE   DJNZ 39FC
384 REM      39FE 3E7F   LD A,7F
386 REM      3A00 DBFE   IN A,(FE)
388 REM      3A02 1F     RRA
390 REM      3A03 D0     RET NC
392 REM      3A04 0638   LD B,38
394 REM      3A06 AF     XOR A
396 REM      3A07 C32505  JP 0525
398 REM      3A0A 0100A3  LD BC,A300
400 REM      3A0D F5     PUSH AF
402 REM      3A0E 08     DEC BC
404 REM      3A0F E3     EX(SP),HL
406 REM      3A10 E3     EX(SP),HL
408 REM      3A11 78   LD A,B
410 REM      3A12 B1     OR C
412 REM      3A13 20F9  JR NZ,3A0E
414 REM      3A15 F1     POP AF
416 REM      3A16 C9     RET

```

2. ábra



- a ZX81 és a használatos memóriabővítők a C000H–FFFFH tartományt legalsóbb MI aktív állapotában a 4000H–7FFFH tartományra képezik le,
- a 8000H–FFFFH tartományból lehívott utasítás aktiválja az ULA videokontroller áramkörét, azaz
  - a) ha az utasítás 6-os bite 0, a processzor NOP utasítást kényszerít;
  - b) az RFSH jel alatt a ROM címcsinjén az I regiszterből, a lehívott karakterkódból és egy belső számláló tartalmából összeállított cím jelenik meg. (A RAM címcsinjén nem! Itt az I és R regiszter tartalma jelenik meg!);
  - c) ezalatt az ULA az éppen megcímezett memóriarekeszből olvasást hajt végre; ha ez ROM-rekesz, az ismertetett „kompozita” címről, ha RAM-rekesz, az I és

adta címről. A tartalmát sftregiszterbe teszi, és a következő NOP alatt min. 8 képpontot a képernyőre kiadja; d) ha az ULA által kiolvasott utasítás 6-os biteje 1, ezt az utasítást eredeti alakban engedi a processzorhoz.

Programunk most már úgy fog működni, hogy képkijáráskor a processzor egy „becsapós” képernyősorra irányítjuk, mely 32 db NOP-ot tartalmaz, a képsor 32 karakterének megfelelően. Közben az I és R regiszter tartalmát úgy manipuláljuk, hogy a Spectrum képernyőfájl tartalma jelenjen meg a képernyőn. A becsapós sor 33. utasítása egy JP... utasítás, a következők sor kiadásának előkészítésére... és a folyamat így megy tovább, végig az összes soron.

A program az előzőek szerint egy NMI rutinból (5. program) és egy tulajdonképpeni képernyő-billentyűzet rutinból (6. program) áll.

Az NMI rutin hasonlóan működik a ZX81-éhez, de az AF és IX regisztert megakartjuk menteni (ezeket a Spectrum használja), ezért a rendszerváltozó terület három nem használt helyét használtuk föl. Ez sajnos bizonyos idővesztéssel jár. Az NMI rutin a következők. Mivel az INT-ot nem használjuk, az NMI rutin egy részét az itt levő Spectrum program egy része helyére írjuk.

Ha katalógusból kikeressük az egyes utasítások végrehajtásához szükséges időt, megállapíthatjuk, hogy minden sor kiadás összesen 207 ütemet vesz igénybe, ahogy ennek az előzőek értelmében lenni kell. Megjegyezzük, hogy az adódó kép a ZX81 képhez képest kissé jobbra helyezkedik el; akit ez zavar, az a 38EBH—38EDH késletét csökkentheti (LD B, 02), ekkor a kép a ZX81-hez pontosan kissé balra lesz. Természetesen pontos egyezés is megvalósítható.

Mint látni fogjuk, a két „becsapós” sornak (a magyarázattól eltérően, ahol egyről beszéltünk, az idők „kisakkozása” miatt kétfőre volt szükség) a kezdetét nem véletlenül választottuk 3880H-nak és 38C0H-nak. Így ugyanis könnyű olyan címértékű áramkört készíteni, mellyel elérhető, hogy csak akkor kapcsoljuk be a videokontrollert, amikor ténylegesen képkijárási van. Ezzel az áramkörrel, valamint a RAM-nak MT aktiv állapotától független címekódolásával az egész RAM-ban futtathatunk gépi programokat.

Most már áttérhetünk a Spectrum ROM egyéb részeinek átalakítására. Ezek kevés kivételtől eltekintve a megszakításkezelés és a be-, kiviteli különbözőségek figyelembevételét jelentik.

Megjegyezzük, hogy az egyes programrészek eléggé összevissza helyezkednek el. Ennek oka, hogy az ismertett megoldás számos próbálkozás eredménye; nem is célja a véglegesség, hanem az olvasó továbbfejlesztésre készítése.

Ezek után lássuk a változtatásokat. Az érthetőség kedvéért helyenként változatlan részeket is megismételtünk (lásd a 7. programot).

Miután az átalakításokat végrehajtottuk és a programot működésbe hoztuk, kipró-

bálhatjuk a Spectrum utasítások működését. Nyomatott hiányában nem foglalkoztunk a nyomtatót vezérlő utasításokkal. Nem módosítottuk a PAUSE utasítást sem. Ha mégis használni akarjuk, akkor értelemszerűen átalakítjuk, esetleg nagyobb számokat adunk meg. A gombok gyors megtalálásához a Spectrum-taszatúrát a ZX81 billentyűzet fölötti üres helyre rajzoljuk. Célszerű a tápegység 5 V-os stabilizátorát is „kihelyezni”.

## Hardverkiegészítés

Ezután ismertetjük azt a hardverkiegészítést, mellyel elérhetjük, hogy az egész 0000H—FFFFH tartományban futhassanak gépi programok. A kiegészítés egy 4 IC-s kapcsolós hozzáadásából áll, melyet például a tasztúrára alatt helyezhetünk el. A kapcsolás az 1. ábrán látható.

Bekötéséhez az ULA 15-ös lábához menő fölött át kell vágni, és a 157-es adatszellektor 4. lábát az ULA-hoz kötni. A működés azon alapszik, hogy ZX81-es üzemban az A15-öt az ULA A15 bemenetével kötik össze, Spectrum üzemban pedig a 3880—389F és 38C0—38DF címtartományok megszólításakor — ez a két pszeudoképernyősor — az ULA A15 bemenetére H szintet kapcsolunk.

Most még a memóriakiegészítő kell úgy átalakítanunk, hogy átkapcsolással a C000—FFFF tartomány átkapcsolását MT aktiv állapotában megszüntessük, az MT-et ne engedjük érvényesülni. Ez a Memopak 64-nél is valószínűleg ugyanolyan jól fog működni, mint a szerző 32 k-sból 64 k-sra átalakított Audio Computers tárolójánál (2. ábra).

Spectrum üzemban a kapcsolót zárjuk, az MT nem tud érvényesülni. A hardvermódosításokkal együtt a szoftvert is módosítani kell, 0003H LD DE, B87F helyett LD DE,FFFF mert a Ramtopnál nem kell tekintettel lenni a pszeudoképernyősorokra, azonfelül

3900 JP Pszeudo—1 + 8000H helyett JP Pszeudo—1

3912 JP Pszeudo—2 + 8000H helyett JP Pszeudo—2

mert a 15. címbit állítására nincs többé szükség a videokontrollert működtetéséhez.

Megjegyezzük, hogy természetesen alkalmas helyre téve a „becsapós” képernyősorokat, az előző esetben is kiterjeszthetjük a memória felső korlátját majdnem FFFFH-ig.

## Programmódosítás

A dolog izgalmasabb része most következik. Hogyan futnak a kész Spectrum programok az átalakított ZX81-en? Csak BASIC programokkal nincs különösebb baj. A gépi kódú programokhoz általában hozzá kell nyúlni. Egyszerűen sikerült adaptálni például az ismert GEN3M2—MON3M2 programokat. Igaz, hogy csak a ZX81 FAST-hoz hasonlóan, azaz a kép

gombnyomások megrándul, programváltás-rehajtás közben eltűnik. Sokat ígérő megoldás lehet az, hogy az NMI helyett az INT-ot használjuk. Ez további IC-k beépítésével és az üzemi rendszer hasonló elveken alapuló átalakításával megoldható. Így a gépi kódú programokba még kevésbé kell belelyúlnunk. Ennek végrehajtását és egyéb programok futásképesse tételét az érdeklődő — és kitartó — olvasóra bizzuk.

## A hivatkozott szakirodalom

- (1) Memopak-Riesen wachgerüttelt. c't 1985/4.
- (2) Für alle Mikrocomputer: Kassettenaufzeichnung mit 3600/7200 Baud. Super-Tape mit dem Z80 am Beispiel des ZX81—16K. c't, 1984/4.
- (3) Super-Tape Komfort für den ZX81. c't, 1984/5.
- (4) Digitaler Kassettenrekorder mit ZX81. Elektor, 1984. június
- (5) Digitaler Kassettenrekorder. Elektor, 1984. január
- (6) Tempo durch Nachbrenner. Funkschau, 1984/4.
- (7) Die Trickreiche ZX81 Videoausgabe. c't, 1985/4.
- (8) ZX-Spectrum ROM programja. 1985. Ipari Informatikai Központ
- (9) ZX-Spectrum. Bevezető és BASIC programozási kézikönyv. IIK, 1985.
- (10) ZX-Spectrum. BASIC és gépi kódú programozás. IIK, 1985.
- (11) Spectrum Hardware Manual. Melbourne House Publishers.
- (12) ZX81 Kochbuch. Funkschau Sonderheft Nr.8. 1984.
- (13) Logan—O'Hara: The Complete Timex TS1000/Sinclair ZX81 ROM Disassembly. Melbourne House Publishers.

BEKE GYULA

## Közületek, figyelem!

**Mikroszámítógépet  
akarnak  
vásárolni?**

**Tájékoztódnak  
előtte  
a naprakész  
piaci  
helyzetről!**

**Díjtalan  
ismertető:**

**MESZ  
Számítástechnika  
1368 Budapest  
Pf. 193.**

# MUNIFORM

Kedves diákok és tisztelt deákok! A mit, mivel, mikor, hol és hogyan kezdtétek a kérdésekre a mikroszámítógépes szakirodalom kapcsán sem könnyű felelni...E kérdések megválaszolása attól is függ, hogy ki és mennyire ismeri a nyelveket, amelyek az elsődleges forrásokból való merítéshez csobolyóul szolgálnak. Pedig bizony dús hozamúak e források: a külföldről könyvtárainkba rendszeresen beáramló mikroszámítógépes lapok.

E szaklapok között még olyanok is akadnak, amelyekből - magáncéla - ki-másolhatók a közölt programlisták, és amelyekben bővítő/illesztőkártyák vagy más egységek kapcsolási rajzai vannak. Ezek hasznosításához még nyelvtudás sem kell! (Nyelvet tanulni, kedves diákok, azért - kell !)

A nyelveket ismerő tisztelt olvasóink közül bizonyára többennek a "mikor" a legnagyobb gondjuk. Nekik is segíthetünk talán azzal, hogy e források feltárássában az első fázisokat készen kínáljuk. Így - már megverve az előválogatás idejét - több lehr-tőségük marad a számukra valóban érdekes programozási fogások, üzemeltetési, karbantartási tanácsok és a - még a sovány magyar túristapénztárcával is megszerezhető - perifériaújdonosságok megismerésére, továbbá olyan cikksorozatok figyelemmel kísérésére, amelyek igazán remekül vannak szerkesztve és célzottan informatívak.

Új szolgáltatást vezetünk be, ami által elsősorban Apple, Atari, Commodore és Sinclair modellek tulajdonosainak a mindennapi munkáját/szórakozását/tanulását reméljük megkönnyíteni. Minden számunkban tartalomeírás teszünk közzé ezután a következő folyóiratok aktuális közleményeiről..

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

Első lépésként a lista szerinti folyóiratok 1986.évi első számaikat dolgoztuk fel C-64/1541/802 rendszerrel oly módon, hogy a tartalomeíró szövegeket permutáltuk, a szövegvariációkat pedig alfabetikusan rendeztük. Így lehetővéadjuk nyílik arra, hogy olvasóink kívánsága szerint válaszuk ki a témákat: pl. "ATARI" vagy "barkácsolás", vagy akár "cikksorozat", "termékmértetés", stb... Most, bemutatásképpen, az adatbázis "programlista" kezdetű rekordjainak listáját mellékeljük.

A közölt listából az érdeklődők készen kapják tehát a legfrissebb információkat, vagyis hogy kedvenc gépükhöz vagy érdeklődési körükhöz kapcsolódóan mi jelent meg ezekben az újságokban. Így már csak a "hogyan szerezzem meg?" kérdés marad: másolatot lehet kérni a könyvtáraktól a kiválasztott anyagról.

( A folyóiratok például a SZÁMALK szak-könyvtárban fellelhetők. A másolás feltételéről bővebb felvilágosítást a 853-111/251 telefonszámon kaphatnak olvasóink.)

A tartalomeírás egy szövegből áll, majd a listában ezt követi a forrás

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

A forráshely karakterozatát nyíly vezeti be, melyet a / jelleg a folyóiratok azonosítója, a két / jel között az évszám, folyóiratszám és kötőjellel a kezdő oldalszám követi, a végén pedig a közlemény teljes oldalterjedelme áll.

Reméljük, hogy olvasóink - a diákok és deákok egyaránt - az első listákat tanulmányozva máris találnak érdekes-ségeket, amelyek hasznosításához sok sikert kívánunk.

Van még egy javaslatunk : Európa legjobb Commodore újságjáról, a 64'er Magazinról másféle információszolgáltatás is kérhető. Szolnoki Béla vállalja, hogy havi 20 Ft-ért (+ a postaköltség) elküldi a folyóirat tartalomjegyzékeinek meggyezéseikkel is ellátott fordítását - a lap 1986. 07. havi számját kezdődően bár melyikről, megbízás esetén folyamatosan. Az eligazító tartalomjegyzék alapján a megrendelő kiválaszthatja, hogy számára melyik cikkek fontosak, és ezek fordítását is megkaphatja - oldalanként 5 Ft-ért (+ a postaköltség). Különös aktualitását adhatja a megrendelésnek, hogy a 64'er Magazin 1986. októbertől sorozatot indított kezdő C64-eseknek!

Összehasonlítva az árakat bármely hasonló szolgáltatással, kedvező az eredmény, ezért ajánljuk olvasóinknak a "Számítástechnikai információszolgáltatás"-t Szolnoki Béla, 1446 Budapest, Pf 400. címen. A Szerkesztőség

## MUVELETGYORSÍTÁS

programlista||commodore 64||lemezegyseg (1541)||<hypra-load>||<ultraload plus>||összehasonlító értékes  
->happ/86.01-48/5

## MUVELETGYORSÍTÁS

termékmértetés||commodore 64||lemezegyseg (1541)||<turbotrans>||<turbo-accs> ->chip/86.01-66/1

## OKTATOPROGRAM KESZÍTÉS

programozási tanácsok||commodore 64;128||data-programok  
->comi/86.01-58/2

## PASCAL PROGRAMOZÁS

programlista||hexload program||<turbo-pascal> ->pers/86.01-42/2

## PASCAL PROGRAMOZÁS

termékmértetés||compiler kinalat||összehasonlító értékes  
->pers/86.02-100/2

## PL/1 PROGRAMOZÁS

programlista||cikksorozat||assembler r. utin behivas ->chip/86.01-152/3

## PL/1 PROGRAMOZÁS

programlista||linearis optimalas||modszertan||absztrakt pelda  
->pers/86.01-50/8

## PROGRAMLISTA

adatalomány Kezeles||commodore 64||basic bovités||<hypra-load>||<datawork-basic> ->64er/86.01-54/5

```

PROGRAMLISTA
  adatalomany vedelem commodore 64 re
  jtjelezes vedoprogram
  ->comi/86.01-54/3
PROGRAMLISTA
  adatavitel apple ii cp/m lemezforma
  tum illesztés ->mc/86.01-62/5
PROGRAMLISTA
  apple ii atari 400/800 xl xe commodo
  re 16;64;128;plus/4 jatekprogram (pa
  sziansz) ->cute/86.01-48/7
PROGRAMLISTA
  apple ii grafikus programozas 3d-gra
  fika ->hc/86.01-67/3
PROGRAMLISTA
  apple macintosh memoriatartalom kiir
  atas (ramedit) ->pers/86.01-46/2
PROGRAMLISTA
  apple billentyuzet modositás
  ->cute/86.01-102/2
PROGRAMLISTA
  apple ucsc-pascal karaktertipus-valt
  as ->mc/86.01-86/1
PROGRAMLISTA
  assembler programozas commodore 64;1
  28 (es-ae 64) alKalmazasi utmutato
  ->happ/86.01-56/7
PROGRAMLISTA
  atari 130xe (copy 130) egymenetes le
  mezmasolo ->happ/86.01-86/3
PROGRAMLISTA
  atari 400/800 xl xe adatvedo rutin r
  eset/break kiiktatas
  ->cute/86.01-109/2
PROGRAMLISTA
  atari 400/800 xl xe basic programoza
  s sortorlo rutin ->cute/86.01-112/1
PROGRAMLISTA
  atari 400/800 xl xe verifikalas (kaze
  tta) ->cute/86.01-100/1
PROGRAMLISTA
  atari 520st (scopy 520 st) filemasol
  as ->happ/86.01-83/2
PROGRAMLISTA
  atari 600/800 xl jatekprogram (dunge
  ons of xotha) ->hc/86.01-70/5
PROGRAMLISTA
  atari 800xl listazas-lasitas (slowl
  ist) ->hc/86.01-94/2
PROGRAMLISTA
  barkacsolas commodore 64 illesztodi
  afilmvetito vezeres
  ->run/86.01-35/4
PROGRAMLISTA
  barkacsolas commodore 64 lemezegyseg
  (1541) muveletgyorsitas busz (parhuza
  mos) ieee-dos generalas
  ->64er/86.01-144/3
PROGRAMLISTA
  basic programozas commodore 64 lista
  -formatalas ->run/86.01-74/3
PROGRAMLISTA
  basic programozas commodore 64;128 <
  error 64> hibas programsor kijelzese
  ->happ/86.01-74/1
PROGRAMLISTA
  cikksorozat commodore 64 muveletgyor
  sitas gepikod generalas (ascompiler-
  64) ->64er/86.01-58/7
PROGRAMLISTA
  cikksorozat commodore 64;128 grafiku
  s programozas koordinata transzforma
  cio hires-3 ->64er/86.01-131/5
PROGRAMLISTA
  cikksorozat pl/1 programozas assembl
  er rutin behivas ->chip/86.01-152/3
PROGRAMLISTA
  commodore 128 sprite-kezeles
  ->comi/86.01-90/3
PROGRAMLISTA
  commodore 64 (disassembler 64)
  ->cute/86.01-89/2
PROGRAMLISTA
  commodore 64 (scroll-64) programlist
  a-gorgetes ->happ/86.01-74/2
PROGRAMLISTA
  commodore 64 (speedscript) modositás
  ->cute/86.01-10/1
PROGRAMLISTA
  commodore 64 (speedscript) karaktert
  ipus modositás ->cute/86.01-107/3
PROGRAMLISTA
  commodore 64 grafikus programozas an
  imacios program (pulsing pictures)
  ->run/86.01-48/5
PROGRAMLISTA
  commodore 64 hangjelzes (vesszo/retur
  n eseten) ->pers/86.02-46/1
PROGRAMLISTA
  commodore 64 hex-billentyuzet
  ->64er/86.01-78/1
PROGRAMLISTA
  commodore 64 jatekprogram (magikus m
  ezo) ->hc/86.01-43/3
PROGRAMLISTA
  commodore 64 jatekprogram (word jumb
  ler) ->run/86.01-104/3
PROGRAMLISTA
  commodore 64 kurzorvezerles fuzerbei
  ras (control-q) ->comi/86.01-52/1
PROGRAMLISTA
  commodore 64 lemezegyseg (1541) lemez
  tar karbantartas masoloprogram
  ->coho/86.01-49/2
PROGRAMLISTA
  commodore 64 lemezegyseg (1541) muvel
  etgyorsitas (hypra-load) (ultraload
  plus) osszehasonlito ertekeles
  ->happ/86.01-48/5
PROGRAMLISTA
  commodore 64 list-kiiktatas
  ->hc/86.01-85/4

```

## PROGRAMLISTA

helyesbites (cute/85.11-92)  
->cute/86.01-87/1

## PROGRAMLISTA

helyesbites (hc/85.09-44)  
->hc/86.01-22/1

## PROGRAMLISTA

Kereskedelemmbasic programozas Kalk  
ulacios programok ->pers/86.01-44/3

## PROGRAMLISTA

linearis optimalas pl/1 programozas  
modszertan absztraKt pelda  
->pers/86.01-50/8

## PROGRAMLISTA

Nyomdo 85.10.13  
basic programozas Kepernyoszoveg Ki  
iratas ->pers/86.02-44/1

## PROGRAMLISTA

pascal programozas hexload program  
(turbo-pascal) ->pers/86.01-42/2

## PROGRAMLISTA

sinclair ql lemezformatalas filemaso  
las ->pers/86.02-46/2

## PROGRAMLISTA

sinclair spectrum (bunte pause) Kere  
tszinezes ->happ/86.01-89/1

## PROGRAMLISTA

sinclair spectrum ablaKkijeloles  
->inc/86.01-62/3

## PROGRAMLISTA

sinclair spectrum jateKprogram (adve  
nturer) ->inc/86.01-107/3

## PROGRAMLISTA

sinclair spectrum jateKprogram (chop  
per mission) ->ysin/86.01-58/4

## PROGRAMLISTA

sinclair spectrum jateKprogram (droi  
d wars) ->ysin/86.01-101/4

## PROGRAMLISTA

sinclair spectrum jateKprogram (KraK  
atoa) ->inc/86.01-104/3

## PROGRAMLISTA

sinclair spectrum jateKprogram (towe  
r of doors) ->inc/86.01-101/2

## PROGRAMLISTA

sinclair spectrum jateKprogram (wash  
ing line) ->inc/86.01-103/1

## PROGRAMLISTA

sinclair spectrum jateKprogram (worm  
) ->ysin/86.01-62/2

## PROGRAMLISTA

sinclair spectrum jateKprogram jateK  
automata szimulator  
->hc/86.01-54/10

## PROGRAMLISTA

sinclair spectrum monitorKep tarolas  
->inc/86.01-71/2

## PROGRAMLISTA

sinclair spectrum morse-Kod generala  
s ->ysin/86.01-61/2

## PROGRAMLISTA

Nyomdo 85.10.13  
comodore 64 nem-garantalt lemezható  
ldal vizsgalat ->hc/86.01-85/1

## PROGRAMLISTA

comodore 64 nyomtato rutin Kepernyo  
szoveg Kiiratas ->64er/86.01-64/5

## PROGRAMLISTA

comodore 64 oraido megjelenites  
->64er/86.01-78/2

## PROGRAMLISTA

comodore 64 uzenetmegjelenites szov  
egmozgato rutin ->cute/86.01-111/1

## PROGRAMLISTA

comodore 64;128 speedcalc alkalma  
zasi utmutato ->cute/86.01-66/12

## PROGRAMLISTA

comodore 64;128 grafikus programoza  
s sprite-vezeres ->happ/86.01-73/1

## PROGRAMLISTA

comodore 64;128 jateKprogram (taxi)  
felhasznalasi utmutato  
->happ/86.01-68/5

## PROGRAMLISTA

comodore 64;128 tarKapacitas Kijelz  
es (free 64) ->happ/86.01-73/1

## PROGRAMLISTA

comodore 64;vc20 list-kiiKtatas (au  
to-run) ->run/86.01-100/2

## PROGRAMLISTA

comodore 16;128;plus/4 gepiKod viss  
zaalakitas (datastater) basic progra  
m gepiKodbol ->comi/86.01-61/3

## PROGRAMLISTA

comodore 16;64;plus/4;vc20 nyomtato  
program listazas ->cute/86.01-99/2

## PROGRAMLISTA

cp/m filekezeles (unera) fileregener  
alas ->pers/86.01-49/1

## PROGRAMLISTA

cp/m mbasic programozas filekezeles  
(random) ->pers/86.02-48/3

## PROGRAMLISTA

dbase ii nyomtato alprogram  
->pers/86.02-52/5

## PROGRAMLISTA

dbase ii ragaszthato cimKek nyomtata  
sa ->pers/86.02-45/1

## PROGRAMLISTA

hc-main home-computer beviteli utmut  
ato atari comodore ->hc/86.01-75/1

## PROGRAMLISTA

helyesbites (chip/85.09-132)  
->chip/86.01-21/1

## PROGRAMLISTA

helyesbites (chip/85.10-188)  
->chip/86.01-21/1

## PROGRAMLISTA

helyesbites (cute/85.11-125)  
->cute/86.01-87/1

## A szótárprogram magyar ábécés változata

A program a magyar ékezetes betűket grafikus üzemmódban írja ki. Emiatt a Beta basicet KEYWORDS  $\theta$  üzemmódba kellett állítani (G kurzor 8). Ebben az üzemmódban a Beta zavartalanul működik, kivéve az ON ERROR utasítást. Grafikus üzemben az új kulcsszavak helyett a felhasználói grafikus karakterek írónak ki a képernyőre. Ez a jelenség a program listázásánál is megfigyelhető.

Például az 55-ös sor:

```
KEYW.  $\theta$ :
55 PRINT #  $\theta$ ; AT 1,1; „Törléshez nyomd meg az ENTER-t!”
```

```
KEYW. 1:
55 ; „T DPOKE rI ROLL shez nyomd meg az ENTER-t!”
```

Vagy a 45-ös sor:

```
KEYW.  $\theta$ :
45 GO TO  $\theta$ a
```

```
KEYW. 1:
45 GO TO ON a
```

Lehet, hogy ez a programlista kicsit furcsán hat, de a program futása ettől még zavartalan. Mivel az ON ERROR így nem

használható, az egész programból ki kell törölni (15, 215, 315, 41 $\theta$ , 73 $\theta$ , 81 $\theta$ -es sorok). Ugyanez vonatkozik a POP utasításokra is (22 $\theta$ , 815-ös sorok). Mivel az ON ERROR-t elhagytuk, nincs szükség a program futásának megállítására sem. Ezért a lista eleje a következő lesz:

```
1 GO TO 3
2 PRINT AT 17,0: LOAD "" CODE
3 INK  $\theta$ : DELETE 1 TO 3
  Természetesen SAVE "pr." LINE 2.
```

Ekkor a program megszakitás nélkül fut, de az első, például magnóöltési hibánál hibáüzenet helyett csak villogó K kurzort látnunk. A 4-5-ös sor elmarad. A 2 $\theta$ -as sorban beállítjuk a KEYW.  $\theta$ -t.  
2 $\theta$  KEYWORDS  $\theta$  : PRINT AT 1,1 $\theta$ ; "SZÓTÁR"

Már csak azt kell megoldanunk, hogy a scrollozás zavartalan legyen. Ez nekem a következőképpen sikerült. Felvettem egy scr nevű változót, amit a kereső eljárás meghívása előtt alaphelyzetbe állítottok:  
212 LET scr=1

215 PROC kereses

```
valamint
81 $\theta$  PRINT INVERSE 1;k$.LET scr=1.PROC sztorles
```

Ezután a kereső eljárást kell módosítani.  
1157 LET scr=scr+INT (LEN i\$/32)+1; vagyis számoljuk a sorokat.

En egy képernyőre 19 sort irattam, mivel az előző változatnál a felső két sor mindig elveszett a 2 $\theta$  INPUT sor miatt. Vagyis:  
1158 IF scr >= 2 $\theta$  THEN POKE 23692,255: PRINT # $\theta$ ,AT 2, $\theta$ ; "scroll?":  
GET a\$: IF a\$="n" OR a\$="N"  
OR a\$= CHR\$ 32 THEN LET h= $\theta$ :  
GO TO 1175: ELSE LET scr=1:  
PRINT # $\theta$ ,AT 2, $\theta$ ;"  
GO TO 116 $\theta$

Tehát ha a sorok száma elérné a 20-at, a 23692-es rendszerváltozóba 255-öt írva kapcsolom a gép saját scrollozó rendszerét, és az enyém lép működésbe. A szokott módon felteszi a "scroll?" kérdést, és "n" vagy SPACE válaszra az eljárás végére ugrik, be-

### PROGRAMLISTA

```
sinclair spectrum tarKapacitas Kiira
tas ->hc/86.01-94/1
```

### PROGRAMLISTA

```
sinclair user beviteli utmutato
->sinc/86.01-101/1
```

### PROGRAMLISTA

```
termeKvalaszteK a/d atalakito commo
ore 64 alkalmazasi pelda
->comi/86.01-66/6
```

### PROGRAMLISTA

```
terminal uzemmod cikksorozat commo
re 64 runterm plus >
->run/86.01-84/7
```

### PROGRAMLISTA

```
uzemeltetesi tanacsok commodore 64
imezegyseg (154) reset-szoftver
->cute/86.01-20/1
```

### PROGRAMLISTA

```
veletlenszam generalas gauss fele ve
letlenszam ->happ/86.01-94/2
```

### PROGRAMLISTA-BEVITELI MOD

```
commodore microcomputers
->comi/86.01-123/3
```

### PROGRAMOZASI TANACSOK

```
commodore 64 felhasznaloi javaslatok
->64er/86.01-74/4
```

### PROGRAMOZASI TANACSOK

```
commodore 64 128 oktatoprogram Keszi
tes data-program sorok
->comi/86.01-58/2
```

### PROGRAMOZASI TANACSOK

```
dbase ii alkalmazasi utmutato
->pers/86.01-41/1
```

### PROGRAMOZASI TANACSOK

```
sinclair spectrum felhasznaloi javas
latok ->ysin/86.01-29/1
```

### PROLOG NYELV

```
adatbazis lekerdezes cikksorozat log
ikai programozas ->pers/86.01-58/2
```

### PROLOG NYELV

```
cikksorozat logikai programozas reku
rziv hivas ->pers/86.02-64/3
```

### PROLOG PROGRAMOZAS

```
termeKismertetes commodore 64 micro
-prolog ->coho/86.01-41/1
```

### ROBOTRON (NDK)

```
termeKismertetes matrixnyomtato (1q)
teszteles illesztési opciók
->pers/86.01-21/3
```

### ROM-RUTINOK

```
cikksorozat commodore 64
->64er/86.01-92/2
```

### SINCLAIR QL

```
programlista lemezformatalas filemas
olas ->pers/86.02-46/2
```

### SINCLAIR SPECTRUM

```
jatekprogram orokelet poke-lista
->ysin/86.01-28/2
```



ajánlat a kirás. Ha itt nem állítanám 0-ra a h változót, akkor szótórlésnél esetleg olyan szót is kitörölhetnék, ami még nem is

jelent meg a képernyőre, hiszen a kirás ezután következik a 1160-as sorban. Más billentyűt lenyomva folytatódik a kirás.

A sok feltételes elágazás miatt a program futása kicsit lelassult, ezért az 1160-as sort módosítottam: 1160 PRINT i\$: PAUSE 10

```

3 CLEAR 28699: GO TO 4
4 PLOT 25,140: DRAW 200,0: DR
AU 0,-110: DRAW -200,0: DRAW 0,1
10 PRINT AT 5,7: "Litauzskij Gyo
rgy": AT 10,7: "S Z O T A R **
" AT 10,12: "© 1986": RETURN
4 GO SUB 3: INK 7: PRINT AT 1
7,0: LOAD ""CODE: RANDOMIZE USR
58419
5 CLS: INK 0: GO SUB 3: INK
7: PRINT AT 17,0: POKE 23609,5
6 LOAD ""

```

```

Litauzskij Gorygy
** S Z O T A R **
© 1986

```

```

A loader:
3 CLEAR 28699: GO TO 4
4 PLOT 25,140: DRAW 200,0: DR
AU 0,-110: DRAW -200,0: DRAW 0,1
10 PRINT AT 5,7: "Litauzskij Gyo
rgy": AT 10,7: "S Z O T A R **
" AT 10,12: "© 1986": RETURN
4 GO SUB 3: INK 7: PRINT AT 1
7,0: LOAD ""CODE: RANDOMIZE USR
58419
5 CLS: INK 0: GO SUB 3: INK
7: PRINT AT 17,0: POKE 23609,5
6 LOAD ""

```

- A menu
- | S Z O T A R    |
|----------------|
| 1. Beiras      |
| 2. Kereses     |
| 3. Kimentes    |
| 4. Betoltes    |
| 5. Szó tortese |
| 6. File tortes |
| 7. Informacio  |
- © 1986

```

Kereses:
310 air=air.rugy
air@mail=legiposta
air=levego
aircraft=repulogep
air travel=legi utazas
air liner=utasszallito
air port=repuloter
air plane=repulogep
air crew=legcsavar
air frame=szkanyu(rep.gepvaz)
air field=repuloter
air borne=legi uton szallitott
air ly=majdnew.szepen,megetheto-
sen
par=par.parosodik

```

```

KERESEK: Visszateres:0
Ha tenyleg csak az 310-kkeressuk
air@mail=legiposta
air=levego

```

```

aircraft=repulogep
air travel=legi utazas
air liner=utasszallito
air port=repuloter
air plane=repulogep
air crew=legcsavar
air frame=szkanyu(rep.gepvaz)
air field=repuloter
air borne=legi uton szallitott

```

```

Visszateres:0
Informacio
INFORMACIO
A tar merete: 27060 byte
Felhasznalva: 16952 byte
Szabad: 10108 byte
Beirva: 789 szopar

```

```

26700 55760
Nyomj meg egy billentyut!

```

```

A program:
1) GO TO 5
2) INK 0: PRINT AT 17,0: LOAD ""CODE
" INK 0: PRINT BRIGHT 1,AT 10
4) Inditsd el a programot.: AT
11,10: (RUN-ENTER)
4) STOP
5) DELETE 1 TO 5
10 CLEAR
15 ON ERROR 15
55 PRINT AT 0,1: "S Z O T A R":
35 PRINT AT 2,2: "
30 PRINT AT 5,7: "1. Beiras":
AT 7,7: "2. Kereses": AT 9,7: "3.
Kimentes": AT 11,7: "4. Betol-
tes": AT 13,7: "5. Szó tortese":
AT 15,7: "6. File tortes": AT 17,
7: "7. Informacio": AT 19,7:
35 PRINT AT 20,2: "
40 PRINT #0,AT 0,11: "© 1986"
45 CLS: GO TO 0 ON A,100,200,30
0,400,800,55,700
50 GO TO 10
55 PRINT #0,AT 1,1: "Torteskez-
nyomj meg az ENTER-t!": PAUSE 0
60 LET r$=INKEY$: IF r$( )<>CHR$
13 THEN GO TO 70
45 CLEAR
45 PRINT AT 0,0: "KERESEK": TAB
15, " Visszateres:0": AT 2,0: LIN
E 15
105 IF b$="0" THEN GO TO 10
110 PROC beiras
115 PRINT b$ TO 800
120 GO TO 100
200 INPUT AT 1,0: "KERESEK": AT
1,15: " Visszateres:0": AT 3,0: L
INE 15
205 IF k$="0" THEN GO TO 10
210 PRINT INVERSE 1,k$
215 ON ERROR 200 PROC kereses
220 POP b$ TO 800
300 INPUT AT 0,0: "KIMENTES": T
AB 15: "Fitenev.": AT 2,0: m$
305 GO SUB 600
310 SAVE #CODE 28700,ci$-28700
315 ON ERROR 300
320 INPUT "MELYIK (i/n) :": i$
325 IF i$<>0 THEN GO TO 10
330 VERIFY #CODE
335 GO TO 10
400 GO SUB 600: IF ci$<>28700 T
HEN PRINT AT 10,1: "Betoltes elo-
tt a szallitandot leesehez ha-
sznald a menu 6.": TAB 12: "opci-
ojat.": PRINT #0,AT 1,3: "Nyomj m-
eg egy billentyut!": PAUSE 0: GO
TO 10
405 INPUT AT 0,0: "BETOLTES":
TAB 15: "Fitenev.": AT 2,0: m$
410 ON ERROR 400
415 LOAD #CODE

```

```

420 GO TO 10
500 RANDOMIZE USR 55783
505 RETURN
600 LET ci$=INSTRING(28700,MEMO
RY$(1),CHR$(0))
605 RETURN
700 LET b$=600: PRINT AT 1,5: "I
N F O R M A C I O": AT 5,1: "A l-
ar merete": TAB 20: "27060 byte":
AT 7,1: "Felhasznalva": TAB 20: c
i$=2700: byte": AT 9,1: "Szabad:
TAB 20: 55760-ci$": byte
705 PLOT 0,50: DRAW 255,0: DRAW
0,5: DRAW -255,0: DRAW 0,-5
710 PLOT 0,50: DRAW 0,-4: PLOT.
128,50: DRAW 0,-3: PLOT 255,50:
DRAW 0,-4
715 PLOT 0,52: DRAW (ci$-28700)
/105: DRAW 0,1: DRAW -(ci$-287
00)/105,0
720 PLOT 0,40: "28700": PLOT 215
,40: "55760"
11. "Beirva": TAB 20,a: "szopar":
730 ON ERROR 200
735 PRINT #0,AT 1,3: "Nyomj meg
egy billentyut!": PAUSE 0: GO TO
10
800 INPUT AT 1,0: "KERESEK":
805 INPUT AT 1,0: "BETOLTES":
TAB 15: "Visszateres:0": AT 3,0: L
INE 15
805 IF k$="0" THEN GO TO 10
810 PRINT INVERSE 1,k$: ON ERRO
R 800 PROC szotortes
815 POP b$ TO 800
1000 DEF PROC beiras
1005 GO SUB 600
1010 IF 55760-ci$<LEN b$+1 THEN
PRINT "Szabad hely:": GO T
O 200
1015 POKE ci$,42
1020 POKE ci$+1,b$
1025 END PROC
1100 DEF PROC kereses
1105 LET c$=INSTRING(28700,MEMORY
$(1) TO 55760),k$)
1110 IF c$<>0 THEN PRINT "New isse-
ret ilyen szot.": LET h$=GO TO
1175
1115 LET h=h+1
1120 GO PRINT AT 17,0: LOAD ""CODE
1125 EXIT IF PEEK c=42
1130 LET c=c-1
1135 LOOP
1140 LET i=c
1145 LET c2=INSTRING(i+1,MEMORY$(
) TO 55760),CHR$(42))
1150 IF c2=0 THEN GO SUB 600: LE
T i=ci$
1155 LET i$=MEMORY$(i)(ci+1 TO c2
-1)
1160 PRINT i$: PAUSE 20
1165 LET c2=INSTRING(c2,MEMORY$(
) TO 55760),i$)
1170 IF c2=0 THEN GO TO 1120
1175 END PROC
1200 DEF PROC szotortes
1205 PROC kereses: IF h=0 THEN G
O TO 1245
1210 PRINT #0,AT 1,3: "Az utolso
szó tortese": GET a
1215 GO TO 0 ON a,1225
1220 GO TO 1245
1225 GO SUB 600: POKE c1,STRING$(
LEN i$,CHR$(0))
1230 POKE c1,MEMORY$(i)(c2 TO ci$
)
1235 GO SUB 600: POKE ci$+1,STRIN
G$(LEN i$,CHR$(0))
1240 PRINT "Az utolso szó töröl-
ve!":
1245 END PROC

```

```

Meg egy kis kereses:
wing=szarny
wing flap=fekszarny
wing spain=feszty
wing area=szarnyfelulet
f=wing
wing=szarny
wing flap=fekszarny
wing area=szarnyfelulet
novel
increase=novel,szaporit,fokoz,e-
mel
raise=emel,novel,ebreszt,okoz
raise=emel,novel,ebreszt,okoz
KERESEK: Visszateres:0

```



I. ábra

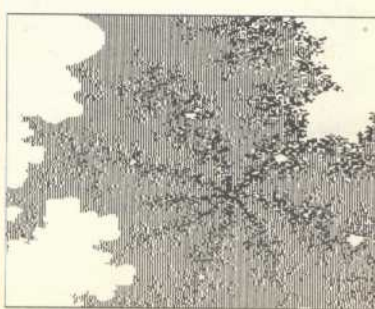
Előző számunkban a PTA—4000-rel készíthető Mandelbrot-képek programjait mutattuk be. Ezúttal a Commodore-hívőknek kedvezünk.

A program alkalmas a Mandelbrot-kép megalkotására, a kép tárolására lemezen, utólagos behívására és átszínzésére is. Lehetőség van arra is, hogy a kép készítését felfüggesztjük, majd máskor folytassuk. A kép készítéséhez szükséges adatok: a kiválasztott számhalmaz szélső értékei (valós és képzetes határok), valamint a különböző színekhez rendelt iterációs lépésszám, röviden színhatárok. Ezeket a kép készítése előtt kell megadni; a képelőállítás felüggesztésekor az adatok szekenciális fájlban tárolódnak, s a folytatáskor a program behívja őket.

A programot szubrutin blokkokból építettem fel. A 20-69-es sorok között van a menü. A kép előállításához szükséges adatok bekérése a 100-160-as sorok között történik. A vizsgált terület képzetes és valós szám határait a  $cv()$  valós, illetve  $ck()$  képzetes tömbben tárolom,  $\theta$  index esetében a kezdeti érték, 1 indexnél a pillanatnyi érték, 2 indexnél a végérték szerepel. A 200-280-as sorban beállítjuk a nagyfelbontású grafikát (a képernyőt \$6000-re), töröljük a képernyőt, illetve a színmémoirea területét, majd a 3030-as szubrutin egy vonalkezelő rajzol. Ezután 350-től indul a két főciklus, ahol x és y a nagyfelbontású képernyő pontjainak koordinátái is.

A 880-as szubrutin végzi magát az iterá-

2. ábra



Szükség volt még arra is, hogy a kírítást bármikor meg lehessen állítani. Erre a SPACE billentyűt választottam.

1162 IF INKEY\$=CHR\$ 32 THEN GO TO 1175 ; elj. végére ugrik.

Ezután a Spectrum-kézikönyv utasításai szerint elkészíthetjük a magyar ékezetes betűket, és hozzáfűzhetjük a Beta gépi kódú programrészéhez. Mivel szótörlésnél „az utolsó szó törölve!” felirat nem elég egyértelmű, az 1240-es sort így módosítottam: 1240 PAPER 5: PRINT IS;TAB 31;” ”: PAPER 7: PRINT ” ”:TAB 11;”TÖRÖLVE”; TAB 31;” ”

A PAPER 7 utáni két ” ” és a ”TÖRÖLVE” felirat két PAPER színvezérléssel van beírva.

Természetesen még sok más dolog megoldható lenne a programmal, de minél tovább bővítenék, annál feljebb kellene helyezni a RAMTOP-ot, ez pedig a beírható szavak számát csökkenti.

Például ha átjavítjuk az 1210-es sort a következőképpen: 1210 PRINT #0,AT 0,0; IS; TAB 31;” ”: PRINT 0;” ”: TAB 10;”TÖRLÉSE:” ”;TAB 31;” ”: GET a akkor szótörlésnél törölés előtt is kírja a gép a törölendő szót. Ekkor azonban ha a RAMTOP-ot nem helyezük feljebb, akkor már csak kb. 5 sor terjedelmű bejegyzést írhatunk be, mert ha ki akarunk törölni, akkor a „4 Out of memory” hibáüzenetet kapjuk. Mindenestre a program szótárnak még így is használható. Én szavak beírásánál egyszer sem használtam fel két sornál többet.

## Még egyszer a Beta basicről

Ha valakinek megvan a Beta programja, de nem ismeri, írja be a következő rövid programot a gépbe:

```
10 FOR n= 255 TO 0 STEP -1
20 PLOT n,88;”Fenyujag”
30 NEXT n
40 GO TO 10
```

Ez a lényegében még csak Spectrum basic kulcsszavakat tartalmazó, de már Beta basicben írt programrész egy balra scrollozó gépi kódú rutint helyettesít. A szöveg jobbról balra átvész a képernyőn, meghozza úgy, hogy ami eltűnik a bal oldalon, az újra megjelenik jobbról.

Már ebből is látható, hogy milyen új lehetőségeket nyújt a Beta. Az új ROLL és SCROLL kulcsszavak pedig ennél sokkal többre képesek.

Sokkal könnyebb a programírás az új programszerkesztő parancsokkal. Ezek a következők:

EDIT (shift nélküli  $\theta$ , de csak ENTER után)

Például: EDIT 210 ENTER.

Az ENTER-t lenyomva megjelenik a 210-es sor az alsó képernyőn, és máris javítható. A javítandó sorban mind a négy(!) kurzorgomb értelemszerűen használható. Ez még nem fullscreen editor, de már nem sok hiya.

RENUM (G kurz. 4)

Például: RENUM + ENT : Az egész programot átsorszámozza, az első új sorszám a 10, a növekmény 10.

Az átsorszámozó rutin a képernyő-memóriát használja, ezért átsorszámozás közben csíkokat láthatunk a képen, de ezek az ENT lenyomása után eltűnnek. A rutin átjavítja a GO TO és GO SUB utasításokat is, de NEM javítja át a GO TO ON utasítás után álló értéket.

DELETE (G kurz. 7)

Erről már volt szó.

AUTO (G kurz. 6)

Például AUTO ENT: automata sorszámadás a következő sortól, tizesével növekedve.

AUTO 25 ENT: automata sorszámadás 25-től, tizesével növekedve.

AUTO 30,4 ENT: automata sorszámadás 30-tól négyesével növekedve.

Hiba: NEW után AUTO parancsra a sorszámadás nem 10-től, hanem az előző (kitörölt) program aktuális sorától kezdődik.

JOIN (G kurz. SYM SH + 6)

Az aktuális sort (> jel) összeolvasztja az utána következővel. Az új sorszám az aktuális soré lesz, a két előző programsort értelemszerűen kettőspont választja el egymástól, és a magasabb sorszámmal megszünik.

SPLIT

Ez nem kulcsszó, hanem normál, tehát nem grafikus kurzorral beírt „nem egyenlő” jel (SYM SH+W).

Ha egy az alsó képernyőn lévő sor bármelyik, de nem első utasítása elé beírjuk a <> jelet, az ENTER-t lenyomva a <> jel előtti rész felkerül a felső képernyőre, a maradék azonos sorszámmal a szerkesztőmezőben marad. Így ha ennek sorszámat átjavítjuk, két külön sort kapunk.

Helyhiány miatt már csak a DEF KEY utasítást említem meg (G. kurz. SYM + 1). Ezzel bármelyik szám- vagy betűbillentyűhöz tetszőleges feladatot rendelhetünk, és így 36 db(!) funkcióbillentyűhöz jutunk.

Írjuk be például: DEF KEY "m": PRINT FN m//;” bytes free” + ENT

A válasz ”0 OK.0:1”. Ha most megnyomjuk a DEF KEY shifet (SYM SH+SPACE), egy villogó szorzásjel látunk (DEF KEY kurzor). A definiált ”m” billentyűt lenyomva a gép ki is írja: ”31315 bytes free”.

Az FN m vagyis MEM függvény Beta nélkül is használható, mivel ROM rutinra támaszkodik. Bárki odairhatja programja elejére a következőt:

```
1 DEF FN m// = 65535 - USR 7962 . Így programírás közben bármikor megtudhatjuk, mennyi helyünk van még PRINT FN m//.
```

A billentyűkhöz rendelt definíciók a RAMTOP fölé kerülnek, így NEW ellen védettek. Szalagra mentésük a Beta kitöltő rutinjával lehetséges, vagy simán is, ha megkeressük az új RAMTOP-ot (23730-as rendszerváltozó).

LITAUSZKY GYÖRGY

# Mandelbrot-program

ciós műveletet, majd a színbeállítás következik, a bitmintázat alapján. A 660-as szubrutin állítja be a megfelelő bitet a színes grafikus képernyőn. A kép előállítás után a program gondoskodik a kép tárolásáról, és a menüre tér vissza.

A tárolt és újra behívott képeket át is színeztük, erre a 830-867-es sorok között van mód. 1500-tól 1670-ig terjed a felbontott kép folytatásához szükséges adatok beolvasása, majd a kép továbbrajzolása a fájlkulnál folytatódik.

Ha a rajzolást abbahagyjuk, az 1700-as sortól elmentődik a felkészítő, illetve fontosabb adatok. Az utóbbiak mindig az „abbahagy” nevű szekvenciális fájlban találha-

tók, és mivel ez minden újabb abbahagyáskor felülíródik, egy lemezoldalon egyszerre csak egy képen dolgozunk, hogy ne legyen a folytatáskor keveredés.

A 2000-es szubrutin állítja be a nagyfelbontású színes üzem módot, a 3000-es törli a grafikus területet. A 3030-as szubrutin rajzolja a keretet, a 4000-es szubrutin állítja vissza a normál képernyőt. Az 5000, 5010, 5020-as szubrutinok egy-egy szint állítanak be a 6000-as színbeállító alaprogram szubrutinjáiként. A 8000-es szubrutin menti el a képet.

A BASIC változatot ne is indítsuk, mert nagyon lassú. Fordítsuk el először egy fordítóprogrammal; én az Austro-speed-et használom. A lefordított program behívást után RUN-nal indítható, ekkor mindjárt a menü jelentkezik:

F-1 (grafikus üzemmód)

F-3 (kép előállítás)

F-5 (kép folytatása)

F-7 (kép behívása)

F-8 (vége)

F-1 (grafikus üzemmód): láthatóvá válik a színes nagyfelbontású képernyőn, ami persze üres állapotban csak „csillagos ég”. Ha előzőleg előállítottunk vagy behívtünk egy képet, akkor az láthatóvá válik. Az F-1 billentyű ismételt lenyomásával újra a menüre léphetünk vissza, a többi funkcióbillentyűvel pedig a színeket állíthatjuk.

F-3 (kép előállítása): először meg kell adni a kép nevét, majd a valós és képzetes számhatárokat. Az így meghatározott négy-szög területét osztja fel a gép 160 x 200-as színes pontthalmazra. Egyre közelebb eső határok megadásával fokozhatjuk a nagyságot. Tapasztalataim szerint néhány milliószoros nagyságot könnyedén el lehet érni. A felbontóképesség határán vízszintes vagy függőleges oszlopszerű képződmények tűnnek fel, attól függően, hogy melyik irányban értük el a maximális nagyságot. Érdekes egyébként lépésszámról lépésre nagytítani, mert könnyen megeshet, hogy egy fekete területre tévedünk.

A területhatárok kijelölése után a színhatárokat kell megadni (c1, c2, c3, c4). Annak alapján dől el a pont színe, hogy hány iterációs lépést kellett végrehajtani ahhoz, hogy a komplex szám értéke 2 fölé nőjön.

iterációs lépésszám < c1 → bitpár: 00 → szín: fekete

iterációs lépésszám < c2 → bitpár: 01 →

1. szín

iterációs lépésszám < c3 → bitpár: 10 →

2. szín

iterációs lépésszám < c4 → bitpár: 11 →

3. szín

Tulajdonképpen a negyedik színhatár (c4) az iterációs felső határa, ezért ez 360-400-nál nem legyen kevesebb.



3. ábra

```

1 REM *****
2 REM *****
3 REM *****
4 REM *****
5 REM *****
6 REM *****
7 REM *****
8
9
10 POKE55,0:POKE56,92:CLR
11 POKE53201,0:POKE53200,0:POKE53272,21
12 BRSE<=64499E
13
14 *****
15 *****
16 *****
17 *****
18 *****
19 *****
20 *****
21 *****
22 *****
23 *****
24 *****
25 *****
26 *****
27 *****
28 *****
29 *****
30 *****
31 *****
32 *****
33 *****
34 *****
35 *****
36 *****
37 *****
38 *****
39 *****
40 *****
41 *****
42 *****
43 *****
44 *****
45 *****
46 *****
47 *****
48 *****
49 *****
50 *****
51 *****
52 *****
53 *****
54 *****
55 *****
56 *****
57 *****
58 *****
59 *****
60 *****
61 *****
62 *****
63 *****
64 *****
65 *****
66 *****
67 *****
68 *****
69 *****
70 *****
71 *****
72 *****
73 *****
74 *****
75 *****
76 *****
77 *****
78 *****
79 *****
80 *****
81 *****
82 *****
83 *****
84 *****
85 *****
86 *****
87 *****
88 *****
89 *****
90 *****
91 *****
92 *****
93 *****
94 *****
95 *****
96 *****
97 *****
98 *****
99 *****
100 *****
101 *****
102 *****
103 *****
104 *****
105 *****
106 *****
107 *****
108 *****
109 *****
110 *****
111 *****
112 *****
113 *****
114 *****
115 *****
116 *****
117 *****
118 *****
119 *****
120 *****
121 *****
122 *****
123 *****
124 *****
125 *****
126 *****
127 *****
128 *****
129 *****
130 *****
131 *****
132 *****
133 *****
134 *****
135 *****
136 *****
137 *****
138 *****
139 *****
140 *****
141 *****
142 *****
143 *****
144 *****
145 *****
146 *****
147 *****
148 *****
149 *****
150 *****
151 *****
152 *****
153 *****
154 *****
155 *****
156 *****
157 *****
158 *****
159 *****
160 *****
161 *****
162 *****
163 *****
164 *****
165 *****
166 *****
167 *****
168 *****
169 *****
170 *****
171 *****
172 *****
173 *****
174 *****
175 *****
176 *****
177 *****
178 *****
179 *****
180 *****
181 *****
182 *****
183 *****
184 *****
185 *****
186 *****
187 *****
188 *****
189 *****
190 *****
191 *****
192 *****
193 *****
194 *****
195 *****
196 *****
197 *****
198 *****
199 *****
200 *****
201 *****
202 *****
203 *****
204 *****
205 *****
206 *****
207 *****
208 *****
209 *****
210 *****
211 *****
212 *****
213 *****
214 *****
215 *****
216 *****
217 *****
218 *****
219 *****
220 *****
221 *****
222 *****
223 *****
224 *****
225 *****
226 *****
227 *****
228 *****
229 *****
230 *****
231 *****
232 *****
233 *****
234 *****
235 *****
236 *****
237 *****
238 *****
239 *****
240 *****
241 *****
242 *****
243 *****
244 *****
245 *****
246 *****
247 *****
248 *****
249 *****
250 *****
251 *****
252 *****
253 *****
254 *****
255 *****
256 *****
257 *****
258 *****
259 *****
260 *****
261 *****
262 *****
263 *****
264 *****
265 *****
266 *****
267 *****
268 *****
269 *****
270 *****
271 *****
272 *****
273 *****
274 *****
275 *****
276 *****
277 *****
278 *****
279 *****
280 *****
281 *****
282 *****
283 *****
284 *****
285 *****
286 *****
287 *****
288 *****
289 *****
290 *****
291 *****
292 *****
293 *****
294 *****
295 *****
296 *****
297 *****
298 *****
299 *****
300 *****
301 *****
302 *****
303 *****
304 *****
305 *****
306 *****
307 *****
308 *****
309 *****
310 *****
311 *****
312 *****
313 *****
314 *****
315 *****
316 *****
317 *****
318 *****
319 *****
320 *****
321 *****
322 *****
323 *****
324 *****
325 *****
326 *****
327 *****
328 *****
329 *****
330 *****
331 *****
332 *****
333 *****
334 *****
335 *****
336 *****
337 *****
338 *****
339 *****
340 *****
341 *****
342 *****
343 *****
344 *****
345 *****
346 *****
347 *****
348 *****
349 *****
350 *****
351 *****
352 *****
353 *****
354 *****
355 *****
356 *****
357 *****
358 *****
359 *****
360 *****
361 *****
362 *****
363 *****
364 *****
365 *****
366 *****
367 *****
368 *****
369 *****
370 *****
371 *****
372 *****
373 *****
374 *****
375 *****
376 *****
377 *****
378 *****
379 *****
380 *****
381 *****
382 *****
383 *****
384 *****
385 *****
386 *****
387 *****
388 *****
389 *****
390 *****
391 *****
392 *****
393 *****
394 *****
395 *****
396 *****
397 *****
398 *****
399 *****
400 *****
401 *****
402 *****
403 *****
404 *****
405 *****
406 *****
407 *****
408 *****
409 *****
410 *****
411 *****
412 *****
413 *****
414 *****
415 *****
416 *****
417 *****
418 *****
419 *****
420 *****
421 *****
422 *****
423 *****
424 *****
425 *****
426 *****
427 *****
428 *****
429 *****
430 *****
431 *****
432 *****
433 *****
434 *****
435 *****
436 *****
437 *****
438 *****
439 *****
440 *****
441 *****
442 *****
443 *****
444 *****
445 *****
446 *****
447 *****
448 *****
449 *****
450 *****
451 *****
452 *****
453 *****
454 *****
455 *****
456 *****
457 *****
458 *****
459 *****
460 *****
461 *****
462 *****
463 *****
464 *****
465 *****
466 *****
467 *****
468 *****
469 *****
470 *****
471 *****
472 *****
473 *****
474 *****
475 *****
476 *****
477 *****
478 *****
479 *****
480 *****
481 *****
482 *****
483 *****
484 *****
485 *****
486 *****
487 *****
488 *****
489 *****
490 *****
491 *****
492 *****
493 *****
494 *****
495 *****
496 *****
497 *****
498 *****
499 *****
500 *****
501 *****
502 *****
503 *****
504 *****
505 *****
506 *****
507 *****
508 *****
509 *****
510 *****
511 *****
512 *****
513 *****
514 *****
515 *****
516 *****
517 *****
518 *****
519 *****
520 *****
521 *****
522 *****
523 *****
524 *****
525 *****
526 *****
527 *****
528 *****
529 *****
530 *****
531 *****
532 *****
533 *****
534 *****
535 *****
536 *****
537 *****
538 *****
539 *****
540 *****
541 *****
542 *****
543 *****
544 *****
545 *****
546 *****
547 *****
548 *****
549 *****
550 *****
551 *****
552 *****
553 *****
554 *****
555 *****
556 *****
557 *****
558 *****
559 *****
560 *****
561 *****
562 *****
563 *****
564 *****
565 *****
566 *****
567 *****
568 *****
569 *****
570 *****
571 *****
572 *****
573 *****
574 *****
575 *****
576 *****
577 *****
578 *****
579 *****
580 *****
581 *****
582 *****
583 *****
584 *****
585 *****
586 *****
587 *****
588 *****
589 *****
590 *****
591 *****
592 *****
593 *****
594 *****
595 *****
596 *****
597 *****
598 *****
599 *****
600 *****
601 *****
602 *****
603 *****
604 *****
605 *****
606 *****
607 *****
608 *****
609 *****
610 *****
611 *****
612 *****
613 *****
614 *****
615 *****
616 *****
617 *****
618 *****
619 *****
620 *****
621 *****
622 *****
623 *****
624 *****
625 *****
626 *****
627 *****
628 *****
629 *****
630 *****
631 *****
632 *****
633 *****
634 *****
635 *****
636 *****
637 *****
638 *****
639 *****
640 *****
641 *****
642 *****
643 *****
644 *****
645 *****
646 *****
647 *****
648 *****
649 *****
650 *****
651 *****
652 *****
653 *****
654 *****
655 *****
656 *****
657 *****
658 *****
659 *****
660 *****
661 *****
662 *****
663 *****
664 *****
665 *****
666 *****
667 *****
668 *****
669 *****
670 *****
671 *****
672 *****
673 *****
674 *****
675 *****
676 *****
677 *****
678 *****
679 *****
680 *****
681 *****
682 *****
683 *****
684 *****
685 *****
686 *****
687 *****
688 *****
689 *****
690 *****
691 *****
692 *****
693 *****
694 *****
695 *****
696 *****
697 *****
698 *****
699 *****
700 *****
701 *****
702 *****
703 *****
704 *****
705 *****
706 *****
707 *****
708 *****
709 *****
710 *****
711 *****
712 *****
713 *****
714 *****
715 *****
716 *****
717 *****
718 *****
719 *****
720 *****
721 *****
722 *****
723 *****
724 *****
725 *****
726 *****
727 *****
728 *****
729 *****
730 *****
731 *****
732 *****
733 *****
734 *****
735 *****
736 *****
737 *****
738 *****
739 *****
740 *****
741 *****
742 *****
743 *****
744 *****
745 *****
746 *****
747 *****
748 *****
749 *****
750 *****
751 *****
752 *****
753 *****
754 *****
755 *****
756 *****
757 *****
758 *****
759 *****
760 *****
761 *****
762 *****
763 *****
764 *****
765 *****
766 *****
767 *****
768 *****
769 *****
770 *****
771 *****
772 *****
773 *****
774 *****
775 *****
776 *****
777 *****
778 *****
779 *****
780 *****
781 *****
782 *****
783 *****
784 *****
785 *****
786 *****
787 *****
788 *****
789 *****
790 *****
791 *****
792 *****
793 *****
794 *****
795 *****
796 *****
797 *****
798 *****
799 *****
800 *****
801 *****
802 *****
803 *****
804 *****
805 *****
806 *****
807 *****
808 *****
809 *****
810 *****
811 *****
812 *****
813 *****
814 *****
815 *****
816 *****
817 *****
818 *****
819 *****
820 *****
821 *****
822 *****
823 *****
824 *****
825 *****
826 *****
827 *****
828 *****
829 *****
830 *****
831 *****
832 *****
833 *****
834 *****
835 *****
836 *****
837 *****
838 *****
839 *****
840 *****
841 *****
842 *****
843 *****
844 *****
845 *****
846 *****
847 *****
848 *****
849 *****
850 *****
851 *****
852 *****
853 *****
854 *****
855 *****
856 *****
857 *****
858 *****
859 *****
860 *****
861 *****
862 *****
863 *****
864 *****
865 *****
866 *****
867 *****
868 *****
869 *****
870 *****
871 *****
872 *****
873 *****
874 *****
875 *****
876 *****
877 *****
878 *****
879 *****
880 *****
881 *****
882 *****
883 *****
884 *****
885 *****
886 *****
887 *****
888 *****
889 *****
890 *****
891 *****
892 *****
893 *****
894 *****
895 *****
896 *****
897 *****
898 *****
899 *****
900 *****
901 *****
902 *****
903 *****
904 *****
905 *****
906 *****
907 *****
908 *****
909 *****
910 *****
911 *****
912 *****
913 *****
914 *****
915 *****
916 *****
917 *****
918 *****
919 *****
920 *****
921 *****
922 *****
923 *****
924 *****
925 *****
926 *****
927 *****
928 *****
929 *****
930 *****
931 *****
932 *****
933 *****
934 *****
935 *****
936 *****
937 *****
938 *****
939 *****
940 *****
941 *****
942 *****
943 *****
944 *****
945 *****
946 *****
947 *****
948 *****
949 *****
950 *****
951 *****
952 *****
953 *****
954 *****
955 *****
956 *****
957 *****
958 *****
959 *****
960 *****
961 *****
962 *****
963 *****
964 *****
965 *****
966 *****
967 *****
968 *****
969 *****
970 *****
971 *****
972 *****
973 *****
974 *****
975 *****
976 *****
977 *****
978 *****
979 *****
980 *****
981 *****
982 *****
983 *****
984 *****
985 *****
986 *****
987 *****
988 *****
989 *****
990 *****
991 *****
992 *****
993 *****
994 *****
995 *****
996 *****
997 *****
998 *****
999 *****
1000 *****

```

## VC20 programok beolvasása

A VC20 számítógép óráuteme nagyobb a C64 üteménél, ezért a magnetofonzalagra eltérő frekvenciával rögzítik az adatokat. Tapasztalataim szerint a VC20-szal be lehet olvasni a C64 programokat, de fordítva nem. Az 1. *listán* köztölt gépi kódú program átmásolja az interpretert és az operációs rendszert a RAM-ba, majd módosítja a szalagolvasás és a szalagra írás időállandóit. A SYS 52000 parancs kiadása után a C64 be tudja olvasni a VC20 által szalagra mentett programokat vagy adatokat. A POKE 1,53 parancs után újra a saját operációs rendszerével dolgozik a számítógép.

Az átmásolást a 2. *lista* BASIC programja több mint egy perc alatt végzi el.

NÉMETH BÉLA

C027 A0 E0	150	LDN #0	
C028 A2 00	160	LDN #0	
C029 20 A0 CB	170	JBR SUB	:OPERACIOS RENDEZER
C030 A0 00	180	LDN #11	:IDOLLANDOR
C030 A0 1E	190	LDY #22	:CSOKKENTESE
C032 09 60 CB	200	READ	LDN C1M,Y
C033 00 42 CB	210	STA POKE+1	
C038 03 61 CB	220	LDN C1M+1,Y	
C039 00 42 CB	230	STA POKE+8	
C03E 00 70 CB	240	LDN ERTEK,X	
C041 00 7F FF	250	POKE	STA #000
C044 00	260	DEY	
C045 00	270	DEY	
C046 CA	280	DEX	
C047 10 E3	290	BPL READ	
C048 60	300	RTS	
C04A 20 FC	310	SUB	STA MUTATO+1
C04C A0 00	320	LDN #0	
C04E 04 FE	330	STY MUTATO	
C050 01 FE	340	MVOL	LDN (MUTATO),Y
C052 01 FE	350	STY (MUTATO),Y	
C054 08 FE	360	INC MUTATO	
C056 00 FE	370	BNE MVOL	
C058 0E FC	380	JLE MUTATO+1	
C05A 0A	390	TAN	
C05B 05 FC	400	CMF MUTATO+1	
C05D 00 F1	410	BNE MVOL	
C05F 00	420	RTS	
C060 01 00 06	430	CJM	WORD#0001,#0000,#F000,#F070,#F070,#F001
C062 30 F3 A0	440	WORD	WORD#F000,#F000,#F000,#F000,#F000,#F000
C070 35 E5 33	450	ERTEK	:BYTES,IRAS,1,1,3E,37
C07E 10 21 98	460	:BYTE,1E,01,100,231,0,102	
C084	470	VEGCIM :END	

### 1. lista

### 2. lista

```
100 FORK=40960TO40151:POKEK,PEEK(K):NEXT
110 FORK=57344TO65535:POKEK,PEEK(K):NEXT
120 POKE488E,2E9:POKE387,21
130 POKE3956,41:POKE3865,3E
140 POKE3875,37:POKE3869,1E
150 POKE446E,01:POKE4430,150
160 POKE446E,231:POKE446B,0
170 POKE4419,162:POKE1,53
```

C64 / VC20	BEITE:1
00F0 100 MUTATO = 251	
C02E 11E # * = 20000	
C02E A9 A0 120 LDN #000	
C02E A2 C0 130 LDN #0C	
C024 20 A0 CB 140 JBR SUB	:INTERPRETER

A halmaz pontjai tehát feketék lesznek (c4-en túl is iterálhatók, a komplex szám értéke mégsem megy 2 fölé), a halmaz közelében lévő pontok az általunk megadott színűek lesznek, a halmaztól távol eső pontok pedig megint csak feketék. A színbeállítást a három színes sávon közvetlenül láthatjuk. Az F-1, F-3, F-5 billentyűvel állíthatjuk tovább a megfelelő színt, a beállítást az F-7 billentyűvel fejezzük be.

Az alapadatok beállításai után célszerű a megfelelő memóriaterületek törlése; ez még lefordítja is kitesz 15-20 másodpercet. A keret megrajzolása után elkezdődik a kép rajzolása, ami 18-36 óráig is eltarthat,

attól függően, hogy mennyire dominálnak a képen a halmazon belüli pontok, ezek kiszámítása tart ugyanis letögvább. Én éjszákánként futtatom a programot, így egy-két éjszaka elkészül a kép. A képkészítés fel-függesztéséhez lenyomjuk a balra nyíl billentyűt (a bal felső), és lenyomva tartjuk a következő pont megrajolásáig, vagy ha ez nem látszik, kb. 30 másodpercig. Ezután meg kell várni, amíg befejeződik a sor, és a felkész kép, valamint adatai elmentésének befejezésével feltűnik a menü.

F-5 (folytatás): a kép nevének megadása és a színek beállításai után a program behívja a képet, és folytatja a rajzolását. Ha elké-

szült a kép, automatikusan elmentődik, majd ismét megjelenik a képernyőn a menü, ezért hosszabb futás esetén se kapcsoljuk ki a lemezegységet!

F-7 (kép behívása): az elkészült és tárolt képek bármikor behívhatók, és az F-3, F-5, F-7 billentyűkkel átszínezhetők.

F-8 (vége): megjelenik a READY és a kurzor. Ekkor még RUN-nal újraindíthatjuk a programot.

DR. HORVÁTH-PAPP IMRE

Három képet közlünk mutatónak, természetesen csak a képek fekete-fehér változatait. Adataik a következők:

190C CLOSE3,0,3	5018 FORI=25520 TO 24551:POKEI,(PEEK(I)+ND240)/000:NEXT:RETURN
1910 GOSUB4000:GOTO40	5019 FORI=55296 TO 55296+999:POKEI,R3:NEXT:RETURN
2000 R1=PEEK(56276):POKE56276,150:R2=PEEK(640):POKE640,92	5020 PRINT"Z"
2005 R3=PEEK(53272):POKE53272,121	6005 K1=K1+K2+K00=###
2010 R4=PEEK(53265):POKE53265,PEEK(53265)/0032	6010 K1=K1+K2+K00=###
2015 R5=PEEK(53278):POKE53278,PEEK(53278)/0R16	6020 K1=K1+K2+K00=### KILEPES
2030 RETURN	6030 GOTO1100
2040	6040 IF#0=##THE#K1+K2=##:R1=R1+16:IFR1>255THE#R1=0
3000 FOR I=BASE TO BASE+7999:POKEI,0:NEXT I	6050 IF#0=##THE#K1+K2=##:R3=R3+1:IFR3>15THE#R3=0
3005 FORI=25520 TO 24551:POKEI,R1+R2:NEXT I	6060 IF#0=##THE#R1+R2=##:R1=R1+16:IFR1>255THE#R1=0
3010 FORI=55296 TO 55296+999:POKEI,R3:NEXT I	6070 IF#0=##THE#R1+R2=##:R3=R3+1:IFR3>15THE#R3=0
3020 RETURN	6080 IF#0=##THE#INT(R1/16)
3030	6090 IF#0=##THE#R1
3035 Y=0:FOR#0TO316:GOSUB960:NEXT X	6095 IF#0=##THE#R1
3040 V=199:FOR#0TO7316:GOSUB960:NEXT X	6096 GOTO6530
3050 W=1:FOR#0TO2000:GOSUB960:NEXT X	6098 POKE211,K1:POKE214,K2:SVS58640:RETURN
3060 X=317:FOR#0TO2000:GOSUB960:NEXT X	7010
3070 RETURN	8000 SVS5781200,0,1
3080	8010 POKE193,0:POKE194,96:REN #0000
4000 POKE53265,R4:POKE53278,R5	8020 POKE174,0:POKE175,120:REN #0000
4010 POKE640,R2	8030 SVS52907:REN SVME
4020 RETURN	9040 RETURN
4040	
5000 FORI=25520 TO 24551:POKEI,(PEEK(I)+ND15)/0001:NEXT I:RETURN	
5005	

Kép-12: valós határai : -0.75111567 és -0.751114522

Képzetes határai : 0.0294705627 és 0.0294705627

színhatárok : 24,130,240,360

Kép-18: valós határai : -0.192593 és -0.185739882

Képzetes határai : -0.8177 és -0.82418

színhatárok : 40,100,200,400

Kép-25: valós határai : -1.75947 és -1.75988

Képzetes határai : 0.019389 és 0.019058

színhatárok : 37,44,76,360



## PTA-4000

# Disassembler program

Bizonyára hasznos segítőtárs lesz a PTA-4000 (PC-1500) használóinak az alábbi disassembler program. A szoftverhiány kényszerítette arra, hogy hozzáfogjak a megírásához. Segítségével számos hasznos

### 1. lista

#### 800C6-803C8

```

00 03 08 23 08 43 08 04 48 00 08 20
00 0C 08 2C 08 13 08 03 01 82 01 02
81 42 01 67 81 47 01 28 02 88 01 75
81 55 81 79 81 59 82 C3 62 C8 42 2F
02 2F 43 22 83 62 43 64 CE FF FF 51
04 28 04 4E 87 88 48 6D 48 33 89 0C
48 23 08 23 48 A3 08 83 08 C8 08 0E
80 0E 4E C5 4E 31 6E 31 8F 8E 0F 12
91 83 51 64 51 03 71 8E 91 4E 91 AF
91 CF 71 33 71 03 52 61 72 21 52 A3
92 83 52 64 72 87 92 47 32 48 72 C8
92 89 92 6F 92 CF 52 33 82 13 73 C8
53 33 85 62 95 42 75 44 75 C 85 75
95 79 95 59 95 47 95 67 28 28 28 28
28 61 62 28 28 28 28 28 2D 28 28 28
41 28 28 28 58 28 28 28 35 C 28 28
28 28 28 28 28 28 4C 28 28 28 48 28
53 28 28 28 78 78 7A 7B 7C 7D 7E 7F
98 FF 78 7D 7F 7F FF FF 82 88 88 83
14 82 88 89 F1 83 08 89 85 14 8E
0A 12 3E 8A 13 83 88 12 33 89 13 81
0C 12 91 8C 13 87 8D 12 37 8D 13 85
0E 12 95 8E 13 8F 8F 14 8F 11 12 9F
11 13 8D 12 12 90 12 13 88 13 12 98
13 13 89 14 12 99 14 13 F7 16 88 86
97 88 86 97 89 87 18 14 4E 98 18 4C
98 19 DF 18 84 42 98 88 46 98 88 8D
1E 18 D0 21 84 48 A1 88 44 A1 88 BA
23 28 84 64 88 84 A4 88 95 25 14 46
A5 18 48 05 19 A4 25 28 47 A8 88 45
A8 88 88 78 18 38 7A 88 88 2D 14 F9
31 88 D1 34 88 08 33 88 E3 35 88 88
36 88 8A 37 88 9A 38 88 88 89 88 88
89 89 81 3A 18 43 38 88 FB 30 88 09
3E 88 05 3F 88 41 C1 88 8E 42 28 E1
43 88 0A 44 88 8A C5 88 88 C3 89 F5
47 88 C3 49 18 C1 4A 18 C0 4C 18 CF
40 18 C8 4E 18 C9 4F 18 C5 58 18 C7
51 18 C8 48 88 08 48 28 E8 48 88 F8
48 88 C2 48 88 D2 48 28 E2 48 18 F2
48 88 C4 48 38 D4 48 18 E4 48 18 F4
48 28 C8 48 88 D8 48 18 E8 48 88 F8
48 88 CA 48 18 DA 48 88 EA 48 88 FA
48 88 CC 48 18 DC 48 88 EC 48 88 FC
48 88 CE 48 38 DE 48 18 EE 48 88 FE
48 88 83 88 87 A3 88 21 4F 81 17 EF
85 31 81 89 A4 87 A3 84 21 49 85 17 E9
85 31 40 8F 17 ED 8F 31 8F 98 87 AF
18 21 87 97 87 A7 17 21 8C 95 87 8C
9A 87 D7 1C 87 D3 10 87 8D 9F 87 AD
1F 21 85 A4 87 A5 24 21 88 AC 87 AB
2C 21 48 AD 17 EB 20 31 81 89 87 A1
39 21 8E C5 87 AE 45 21 CA 82 88 CE
06 88 DE 87 88 CC 88 88 EC 85 88 8E
15 88 42 98 85 81 28 88 48 A1 89 8A
22 88 8E A2 88 48 27 8A 58 27 85 4C
28 88 8A 2E 84 8A AE 88 C8 2F 84 88
AF 88 C8 38 88 8E 32 88 C1 3C 88 81
48 88 4A C8 88 4E 46 8A 5E 46 85 8A
48 88 68 78 8A C8 84 8E 83 84 01 84
8E 83 85 81 84 8E 83 84 01 84 8E 83
85 01 85 88 2E 9A F1 89 8F 87 8A 83
83 83 38 9A 83 38 9A
    
```

```

10: "A" CLEAR : CSIZE 1: COLOR 1: INPUT
"KEZDOCTI=" : X
20: S=8: A=X: GOSUB 240: LCURSOR 8:
LPRINT A#:
30: B=PEEK X: IF B=25360TO 68
40: H=58: FOR I=422TO 839STEP 3: GOTO
258
50: NEXT I: GOTO 98
60: X=X+1: B=PEEK X: LCURSOR 6: LPRINT
"FD
70: H=88: FOR I=752TO 917STEP 3: GOTO
258
80: NEXT I
90: LCURSOR 13: LPRINT "7": FOR I=486
TO 419: IF PEEK I=B60TO 228
100: NEXT I: FOR I=428TO 421: IF PEEK I
=B60TO 218
110: NEXT I: GOTO 238
120: LCURSOR 18: LPRINT B#:
130: LCURSOR 8: A=B: GOSUB 240: LPRINT C
#:
140: E=(PEEK (I+1))AND 127: %2+198: F=
((PEEK E)AND 31)+65: G=(PEEK (E+1)
)AND 31)+65
150: H=((PEEK (E))AND 224)+((PEEK (
E+1))AND 224)/32+48
160: LCURSOR 12: LPRINT CHR# F: CHR# G:
CHR# H:
170: E=PEEK (I+2)AND 153: %4+362:
LCURSOR 17
180: LPRINT CHR# PEEK E: CHR# PEEK (E+
1): CHR# PEEK (E+2): CHR# PEEK (E+
3):
190: E=(PEEK (I+2)AND 48)/16+1: ON E
GOTO 238, 228, 218, 208
200: A=PEEK (X+3): GOSUB 240: LCURSOR 2
7: LPRINT C#: S=S+1
210: A=PEEK (X+2): GOSUB 240: LCURSOR 2
5: LPRINT C#: S=S+1
220: A=PEEK (X+1): GOSUB 240: LCURSOR 2
3: LPRINT C#: S=S+1
230: LF 1: X=X+S+1: GOTO 28
240: U=INT (A/250): C=A-U*250: POKE I*9
+8, %2, U: C: CALL 928, A: C=#: MID# (A
#, 3, 2): RETURN
258: D=(PEEK (I+1))AND 128: IF D=128
GOTO 268
268: IF PEEK I=B60TO 138
278: GOTO H
288: IF PEEK I=BTHEN LET B=#"X": GOTO
128
298: IF (PEEK I)+16=BTHEN LET B=#"Y":
GOTO 128
308: IF (PEEK I)+32=BTHEN LET B=#"U":
GOTO 128
318: GOTO 278
    
```

### 2. lista

és praktikus ROM-rutin visszafejthető, s így jelentősen növekszik a gépen megoldható feladatok száma.

A program egy adatmezőből (1. lista) és az ezt kezelő BASIC programból (2. lista) áll.

#### A BASIC program felépítése

10-20 Kezdőértékek beállítása, az első cím kinyomtatása.

30 Megvizsgálja, hogy a kérdéses utasítás az 1. vagy a 2. lapra vonatkozik-e.

40-50 A 1. lapra vonatkozó utasítások közül a kérdéses kikeresése az adatmezőből.

60-80 A 2. lapra vonatkozó utasítások közül a kérdéses kikeresése az adatmezőből.

90-110 Megállapítja az ismeretlen utasításokhoz tartozó adatbájtok számát.

120-180 Az utasításkód és a mnemonik kinyomtatása.

190-200 Az adatbájtok számának megállapítása és kinyomtatásuk.

230 Az adatbájtok utáni első utasításbájt megkeresése, a 20-as sorba visszatérve a folyamat újra indul.

240 Decimális-hexadecimális konverter.

250-310 Sorutastás-vizsgáló rutin.

A BASIC program felépítése olyan, hogy időigényesebb részei egy nyelven is könnyen megírhatók, és egyszerű a program assembler kiegészítése is. A program a bővíthető gépeken is futtatható, ekkor a BASIC program 240-es sorában lévő címet és

```

A891 BE SJP B8E8
A894 BE SJP B116
A897 FD98 PSH# Y
A899 BE SJP ACBB
A89C 85 LDI A 82
A89E AE STA (ab) 79F4
A8A1 BE SJP AAD9
A8A4 BE SJP A9E4
A8A7 BE SJP ABE1
A8AA BE SJP A386
A8AD 6A LDI UL 88
A8AF 4A LDI XL 88
A8B1 BE SJP A28E
A8B4 FD88 PSH# U
A8B6 BE SJP A519
A8B9 FD88 PSH# Y
A8BB 85 LDI A 19
A8BD BE SJP AC2F
A8C8 85 LDI A 19
A8C2 BE SJP AC2F
A8C5 FD1A POP# Y
A8C7 FD2A POP# U
A8C9 68 INC UL
A8CA 88 CP1 UL 84
A8CC 88 BZS + 84
A8CE 4A LDI XL 28
A8D8 9E BCH - 21
A8D2 BE SJP A9E4
A8D5 6A LDI UL 88
A8D7 BE SJP A519
A8DA BE SJP ABE1
A8DD FD1A POP# Y
A8DF 8E BCH + D6
A8E1 85 LDI A 82
A8E3 AE STA (ab) 79F4
A8E6 85 LDI A 84
A8E8 43 SDE
A8E9 85 LDI A 88
A8EB 8E STA (X)
A8EC BA JMP AAB4
A8EF BE SJP AC97
A8F2 85 LDI A 88
A8F4 AE STA (ab) 79E7
A8F7 AE STA (ab) 79E8
A8FA AE STA (ab) 79EA
A8FD AE STA (ab) 79E9
A8C0 AE STA (ab) 79F2
A8C3 AE STA (ab) 79F8
A8C6 9A RTN
    
```

### 3. lista

a &398-as címen kezdődő konvertáló rutin megfelelő abszolút ugróutasításait kell átírni.

A 3. listán a program által disassemblált TEST rutin látható.

# COMMODORE 64

## HELP bővítő

Szívesen dolgozom a Print Technik 8 k-s BASIC bővítésével. A program módosításának, illetve kiegészítésének gondolata akkor merült fel, amikor a C64 „ROM alatti” területeire lettem kíváncsi; például a SIMON's BASIC, az Easy Script stb. programok egy része itt, RAM-ban található.

A megírt kiegészítés így elsősorban a ROM-ok alatti RAM-ok vizsgálatát és használatát segíti.

A bővítés alapja egy lemezen meglévő HELP+ program, mely két részből áll, egy

```
400 PRINT "COMMODORELEMET KEREK..."
500 IFR=0 THEN#A1 LOAD"HELP-1",8:1
510 IFR=1 THEN#A2 LOAD"HELP-2",8:1
610 POKE$2-128:POKE$6:128
620 CLR
630
730
760 POKE768:139 POKE769:227 POKE772:124
770 POKE773:165 POKE844:128
780 POKE195:1 POKE196:0
790
810 SVS33000
820 SVS40744
870 REM FLOPPY FPM-BA FORDITAS
880 REM POKE39014,135 -TEL LEHETSEGES
890 NEM
```

### 1. lista

BASIC-betöltőből és egy gépi kódú programból. Ez utóbbinak végcíme: \$9F02, így a bővítő gépi kódú program elhelyezhető a \$9F03-\$9FFF tárterületen; jelen esetben ez \$9F28-\$9FFF.

### A program további szolgáltatásai

A HELP+ monitor üzemmódjában a program írja-olvassa a \$A000-\$BFFF, \$E000-\$FFF és a \$D000-\$DFFF címirtomány RAM-ot. Ez utóbbi területhez úgy lehet hozzáférni, hogy a programban kapcsoljuk mind a BASIC, mind a KERNAL ROM-ot, tehát nem azonos azzal a \$D000-\$DFFF címirtományal, melyet többek között a VIC, a SID és a CIA veszi igénybe. A \$9FDA rekesz megfelelő átirásával (\$32) természetesen olvashatjuk a karakter ROM-ot is.

A HELP+ assembler üzemmódjában lehetőség van arra, hogy assembly programot mint gépi kódú programot fordítsunk be a fent említett területekre (\$A000-\$BFFF, illetve \$D000-\$DFFF). A lemezegység puffertárába (RAM) is fordíthatunk be assembly programot (8-as egység). Ez a floppy \$300-\$7FF tárterületét jelenti; a VC 1541 lemezegységgel próbáltam ki.

Fordítás előtt a POKE39014, 135 parancsot kell kiadni. A fordítás befejeztével a rekesz tartalma felveszi eredeti értékét,

9F28	A9 20	LDR	#20
9F2A	A2 9F	LDR	#9F
9F2C	A0 9F	LDR	#9F
9F2E	8D 18 9C	STR	9C18
9F31	8D 36 8C	STR	9C36
9F34	8D 88 8C	STR	9C88
9F37	8D 65 98	STR	9C65
9F3A	8E 23 88	STR	9823
9F3D	8E 1A 8C	STR	9C1A
9F40	8E 9A 8C	STR	9C9A
9F43	8E 82 8C	STR	9C82
9F46	8E 6E 91	STR	916E
9F49	8E 67 98	STR	9867
9F4C	8C 18 8C	STR	9C18
9F4F	8C 36 8C	STR	9C36
9F52	8C 83 8C	STR	9C83
9F55	8C 68 98	STR	9C68
9F58	8C 69 98	STR	9C69
9F5B	8C 69 98	LDR	#7D
9F5D	A2 4C	LDR	#4C
9F5F	A0 86	LDR	#86
9F61	8D 22 88	STR	8822
9F64	8E 6C 91	STR	916C
9F67	8C 6D 91	STR	916D
9F6A	A9 08	LDR	#08
9F6C	A2 87	LDR	#87
9F6E	A0 F3	LDR	#F3
9F70	8D 39 8C	STR	9C39
9F73	8D 61 8C	STR	9C61
9F76	8E 19 8C	STR	9C19
9F79	8C 66 98	STR	9866
9F7C	68	RTS	
9F7D	A9 00	LDR	#00
9F7F	85 38	STR	38
9F81	8D 84 82	STR	8284
9F84	4C 22 E4	JMP	E422
9F87	3A	TAX	
9F89	48	FHR	
9F8B	A5 FB	LDR	FB
9F8D	48	FHR	
9F8F	A5 FC	LDR	FC
9F91	48	FHR	
9F93	A2 57	LDR	#57
9F94	B9 4E 00	LDR	#04E.Y
9F97	85 FD	STR	FD
9F99	36	CLC	
9FA0	18	CLC	
9FA2	65 41	ADC	41
9FA5	85 FB	STR	FB
9FA7	A9 00	LDR	#00
9FA1	65 42	ADC	42
9FA3	A5 FC	STR	FC
9FA5	A9 00	LDR	#00
9FA7	28 CC 8C	JSR	8CCC
9FAA	68	PLA	
9FAB	85 FD	STR	FD
9FAD	68	PLA	
9FAE	A5 FC	STR	FC
9FAB	68	PLA	
9FAD	85 FB	STR	FB
9FB3	68	PLA	
9FB4	AA	TAX	
9FB5	60	RTS	
9FB6	A9 F3	LDR	#F3
9FB8	8D 66 98	STR	9866
9FB9	AD 28 02	LDR	0288
9FBE	09 80	ORA	#80
9FC0	AD	TRY	
9FC1	A9 00	LDR	#00
9FC3	AA	TAX	
9FC4	34 D9	STR	D9..X
9FC6	18	CLC	
9FC7	69 28	ADC	#28
9FC9	38 01	BCC	9FC9
9FCB	C8	INX	
9FCC	E8	INX	
9FCD	E0 1A	CPX	#1A
9FCE	00 F3	BNE	9FC4
9FD1	A9 FF	LDR	#FF
9FD3	95 D9	STR	D9..X
9FD5	4C 86 E3	JMP	E386
9FD6	79	SEI	
9FD9	A9 34	LDR	#34
9FDB	85 01	STR	01
9FDD	B1 FB	LDR	(FB)..Y
9FDF	85 FD	STR	FD
9FE1	A9 37	LDR	#37
9FE3	85 01	STR	01
9FE5	58	CLT	
9FE6	60	RTS	
9FE7	78	SEI	
9FE8	A9 34	LDR	#34
9FEA	85 01	STR	01
9FEC	A5 FD	LDR	FD

9FEE	91 FB	STR	(FB)..Y
9FF0	4C E1 9F	JMP	9FE1
9FF3	78	SET	
9FF4	A9 34	LDR	#34
9FF5	85 01	STR	01
9FF8	B9 4E 00	LDR	004E..Y
9FFB	91 41	STR	(41)..Y
9FFD	4C E1 9F	JMP	9FE1

### 2. lista

vagyis a következő fordítás ismét a C64 memóriájába megy.

A lemezegység RAM-ban lévő gépi kódú program futtatható a „M — E” utasítással, a HELP+ monitor üzemmódjában pedig a \* jellel.

A lemez HELP egy SYS 64738 (RESET) után, bejelentkezésének megfelelően, a BASIC RAM végét \$A000-ra állítja. Egy futó BASIC program elronthatja a HELP+ végét. Ekkor a sztringterület eleje \$A000! Védelem ez ellen: RESET után a BASIC RAM vége \$8000-ra állítódik, s a szabad BASIC bajtok száma is ennek megfelelően íródik ki a képernyőre.

### A HELP bővítő kialakítása, használata

A már meglévő BASIC-töltő programot módosítani kell az 1. lista alapján. A 2. lista szerinti HELP—2 nevű gépi kódú programot lemezre kell rögzíteni. Így a BASIC-töltő futásakor a meglévő HELP—1 gépi program mellett betöltődik és lefut a mi HELP—2 nevű programunk is.

A program használata a bejelentkezés után a szokásos. Ha ismét a BASIC vagy KERNAL ROM területét, illetve a VIC, SID, CIA rekeszeit szeretnék vizsgálni monitor üzemmódban, akkor a \$9FDA, \$9FE9 rekeszre tartalmát \$37-re kell átírni.

Megjegyzés: A tár 1-es címének olvasásakor monitor üzemmódban \$34-et kapunk, mivel ekkor ki van kapcsolva a KERNAL és BASIC ROM.

BÖDÖR LAJOS

MEGALAKULT A HCC

CASIO-SHARP

SZEKCIÓJA

Az állásértékelésnek két fő töréspontja van. Az egyik: átmenet a megnyitástól a középjátékba, a másik pedig: átmenet a középjátéktól a végjátékba. Nagyon nehéz meghatározni, hogy meddig tartanak az egyes részek, mert meglehetősen szubjektív, hogy a szóban forgó játszmában a lépéseket meddig tekintjük a megnyitáshoz tartozónak, és mikor kezdődik a középjáték. Ugyanígy: hol van a határ a középjáték és a végjáték között? Egyes programok a lépésszámtól teszik ezt függővé, mások pedig a táblán lévő anyagi érték összegétől.

A megnyitás és a középjáték közötti átmenetet általában a lépett figurák számával határozzák meg. Ez egyszerű, de hatásos módszer. Nagy segítséget nyújt az is, hogy a programok általában az első lépéseket nem számítják, hanem egy — adatbázisszerű — megnyitási könyvtárból veszik. Ez a megnyitási könyvtár a különböző programoknál és az eltérő típusú sakkszámítógépeknél más és más; nagyságuk is sokban eltérhet. Így például az ismert CRAY BLITZ program 5000 lépést, a MEPHISTO 20 ezer lépést, az AWIT 10 ezer lépést tartalmaz; a KEMPELEN C64-be e sorok írója 3000 lépést táplált be.

A megnyitási könyvtár tanácsos mindkét színhez külön-külön elkészíteni. Az egyik változat világosnak csak a jó lépéseit tartalmazza, viszont az összes lehetséges sötét választás benne van; a másik pedig csak sötét jó lépéseit tartalmazza az összes lehetséges világos választással. Ennek az a célja, hogy ha például a gép játszik világosként, akkor azt a könyvtárat veszi, amelyikben világos jó lépései és az összes sötét választás

## BITEK ÉS FIGURÁK

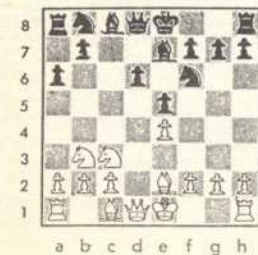
### Megnyitáselmélet és pozíciótábla

található. A program a könyvtárból csak jó lépéseket adhat, és megtalálja a rá vonatkozó válaszlépéseket is. Így az algoritmus nem tévedhet el a megnyitási változatok elvesztőjében, viszont követhet egy „gyilkos” könyvtárat, amelynek lényege, hogy olyan változatokba tereli a játékot, amelyek könyvtára nagyon részletesen ki van dolgozva, és tartalmazza az ide vonatkozó megnyitási sakkcsapdákat is.

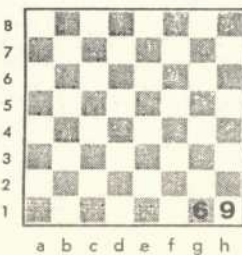
A megnyitási könyvtárat nagyon precízen kell kidolgozni, mert főként ez határozza meg a későbbiekben a játék stílusát. Nem mindegy, hogy a megnyitás végállásától az algoritmusnak egy éles szicíliai védelmet kell folytatnia, avagy egy vezércselt. Ezért a megnyitás lépéseit úgy a legcélszerűbb kiválasztani, hogy az elkészült sakkprogrammal több játszmat is lejátszunk: kipróbáljuk, hogy egyes helyzetekre hogyan reagál, melyik lépést találja meg magától is az algoritmus, és hol szorul segítségre. Ezek alapján megállapíthatjuk, melyik megnyitási rendszert kezelni jól az algoritmusunk, és ennek alapján döntjük el, hogy mely megnyitásokból állítsuk össze a könyvtárat. Ha nincs elég memóriánk, akkor főként azokkal a lépésekkel bővítjük

a könyvtárat, amelyek helyett a program algoritmus a rossz szakat lépne. Ezek után bővíthetjük egyéb lépésekkel, hogy a játszma végére több idő maradjon. Ugyanis a megnyitás

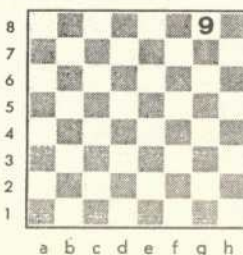
1. ábra. A szicíliai védelem egyik közismert változatának alapállása



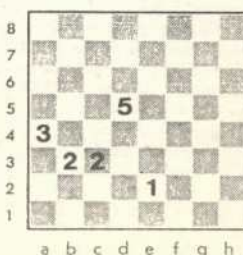
2/a. ábra. Világos király pozíciós táblája



2/b. ábra. Sötét király pozíciós táblája

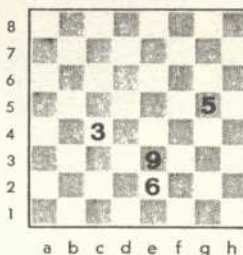


5/a. ábra. Világos huszár pozíciós táblája



3/a. ábra. Világos gyalog pozíciós táblája

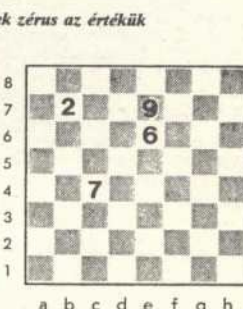
Az üresen hagyott mezőknek zérus az értékük



4/a. ábra. Világos futó pozíciós táblája

Az üresen hagyott mezőknek zérus az értékük

3/b. ábra. Sötét gyalog pozíciós táblája



4/b. ábra. Sötét futó pozíciós táblája

Az üresen hagyott mezőknek zérus az értékük



sal megtakarított időt a versenyfokozaton a program a közép- és végjátékban felhasználhatja, ezzel is növelve a program játékerejét.

A megnyitási könyvtár a legtöbb sakkprogramnál konkrét lépések sorozatát tartalmazza, aminek az a hátránya, hogy a program nem ismeri fel a lépéscserével létrejövő állásokat. Hacsak azok is nincsenek letárolva a memóriában, az algoritmus nem tudja azonosítani az állást, és kénytelen a saját értékelőfüggvényét használni. Egyes sakkprogramok ezért főként nem konkrét lépéseket tárolnak, hanem minden egyes megnyitásra külön algoritmust írtak, amelyek az egyes megnyitások stratégiai elveit súlyozzák nagyobb mértékben. Előny a kisebb tárkapacitás, hátrány, hogy a program már az első lépésekkel is időt veszít.

A KEMPELEN ATARI program a kettő közötti átmenetet jelenti: a konkrét lépések vannak betáplálva, de amikor elfognak ezek a lépések, akkor a könyvtár az értékelő algoritmusnak egy súlyozott sakkjátszót ad át, amelyet pozíciós táblának nevezünk el. Ennek a táblának a súlyozása a megnyitástól függően változik. A pozíciós táblával a program minősíthet a különféle megnyitásokból kialakult állások között, így a megnyitások más-más pozícióban különbözőséget tehet az egyes stratégiák között, amelyek nem általánosíthatók az összes állásra. Például más a megnyitási stratégiája egy szicíliai védelemnek, ahol a másik oldalon elhelyezkedő királyt kell megtámadni, mint egy Benoni-védelemnek, ahol a vezérszárnyon folyik a küzdelem. A szicíliai védelemben sötét a c vonalon, világos a g vagy a h vonalon indít támadást az ellenséges király el-

len. A Benoni-megnyitás sötétje a világos gyalogcentrum aláknázására, fellazítására törekszik b7—b5 lépéssel, míg világos vagy az e4—e5 áttörést kísérel meg, vagy a b2—b4—b5 stratégiát választja.

Hasonló stratégiai elemek figyelembevételével nagyon egyszerű a pozíciós táblával, amelyet elkészíthetünk külön-külön minden figuratípusra az egyes megnyitási változatoknál, mind világgal, mind sötéttel. Ekkor tehát tizenkét pozíciós táblát kell kezelniük megnyitásonként, de megéri a fáradságot. A következőt javaslom.

Egy teljesen külön memóriában tároljuk ezeket a pozíciós táblákat, olyan módon, hogy tizenkettővel (figuratípusonként), sorszám szerint tudjunk rájuk hivatkozni, így a megített lépéssel tárolt pozíciós táblák sorszámát kapja meg. Ezeket a táblákat átmásolja magának a megfelelő helyre, és így dolgozik tovább.

A pozíciós táblák súlyozását többféle módon szervezhetjük. A legegyszerűbb az egyes figurák kedvező elhelyezkedési mezejét felpontozni. Ezzel kompenzálhatjuk például azt az esetet, amikor a sötét vezér a5-re lép c7 helyett, mivel a5-ön nagyobb lesz a mozgékonyasága és a centrumértéke. Az 1. ábrán bemutatott állás pozícióitábláit láthatjuk a 2—5. ábrákon.

A megnyitási könyvtár lépésválasztásának másik szempontja, hogy olyan lépéseket válasszunk ki, amelyek során a könyvtárból való kilépés pillanatában a gyalogok helyzete igazodik a kibontakozó középjátékhoz. Vagyis törekedjünk arra, hogy a játszó algoritmus kevés gyaloglépést tegyen. Minden egyes gyaloglépést nagyon meg kell fontolni, mert ezzel a bábbal tilos a visszafelé lépés, és az esetleges rossz lépést képtelenség visszavenni. Ha a megnyitásban már kialakítjuk a középjáték végleges gyalogszerkezetét, akkor az értékelőfüggvényben a gyaloglépéseket megfelelően leponthozhatjuk, kivéve a gyalogham vagy gyalogtámadás esetét. Így nagyobb az esélye, hogy programunk nem lép gyaloggal rosszul.

KOVÁCS P. ATTILA



## Interaktív kompakt lemezek

A képen látható szedéssel, kiemelten hozta a hírt a 68 Micro-nevű szaklap arról, hogy a Sony és Philips cégek szabványosították az interaktív kompakt lemezeket, és az ezeket kezelő szoftvert a Microware nevű cég, és nem a Digital Research (a CP/M készítője), nem a Microsoft (az MSDOS készítője) írta.

Miért olyan fontos ez az esemény? A kompakt lemezek kapacitása sokkal nagyobb, mint a hagyományos hanglemezeké, és lényegesen jobb a hangminőségük. Ezen okok miatt a hanglemezgyártás újra virágzásnak indult, sőt: az eladott darabok száma új csúcsokat ért el. Természetesen tűnt, hogy digitális információárolásuk miatt érvényesülnek a számítástechnikában is. Az ún. interaktív változatokra ez máris igaz.

Mi az interaktív lemez? Egy speciális lejtáron leforgatható kompakt lemez. Mint fontos tulajdonság megemlítendő, hogy az ezekhez készült lemezek elfogadják a közönséges kompakt lemezeket is, és néhány interaktív lemez hagyományos lejtáron is használható. (Ez utóbbi esetben azonban az új tulajdonságok nem érvényesülnek.) Az új és számunkra alapvető fontosságú tény: kódolt képek és adatok tárolhatók!

Az interaktív lemezeknek a mágneslemezekhez hasonlóan van tartalomjegyzék és állomány része. A fájlleíró rész a fájl típusára vonatkozó információt is tartalmaz (a fájl lehet: hang, kép vagy egyéb típusú). Az egyes fájlakon belül a típus blokkonként változhat. A lemez kezelését a lejtáron tárolt operációs rendszer végzi. Ez az új szabvány szerint az OS9 elnevezésű. (Erről az operációs rendszerrel a következőkben számunk adunk ismertést. Itt csak annyit: ez egy több felhasználós, osztott idejű rendszer.) Az operációs rendszer a fájl típusától függően két videochipen, egy audiochipen keresztül, vagy közvetlenül olvassa az információkat. (A hardver egyébként egy Motorola 68000 típusú — tehát 16 bites — mikroprocesszorú célszámítógép.)

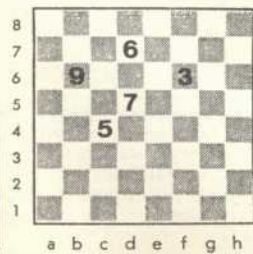
A hangok tárolását az eddig megszokottaktól eltérően a jel deriváltjával oldják meg. A lemez kapacitása 640 Mbájt. A keresési idő igen kedvező, de az átviteli sebesség nem éri el a merev lemezeknél megszokottat.

Ezek miatt a kezelő szoftver igyekszik az információkat a lehető legkedvezőbb helyzetben tartani, ezért a fájlneveket egyúttal megkeresi az elérési idő szempontjából még rendelkezésre álló legjobb helyet. Az OS9 rendszer hagyományos változata a fájlleíró részt a fájl olvasása közben kapja meg, ezért hangtípusú információk esetén nem használható. Az új változatnál a fájl elé teszik a leíró részt. A fájlok itt mindig statikusak!

A szabvány alapvetően a hang rögzítésére szolgáló rendszerekre készült, de szövegek, adatbázisok, nagyméretű programok tárolását is célul tűzték ki. A nem elégséges sávszélesség miatt a képek minősége jelenleg még nem olyan jó, mint a videoszalagoknál.

Mivel a piacot uraló két cég ezt az operációs rendszert fogadta el, ebből várhatóan többen fognak érdekesíteni, mint az eddig bármilyen célra kidolgozott operációs rendszerekből.

5/b. ábra. Sötét huszár pozíciós táblája



# Kétórás játékok – miért ne?

## COMMODORE 64

A Commodore 64-re eddig viszonylag kevés számú játékkora-programot publikáltak. Ezek közös jellemzője, hogy csak egy óra működését tették lehetővé.

Egy részük BASIC nyelven íródott, és az elindított órát csak RUN/STOP-pal lehetett megállítani. Másik részük már lényegesen jobb volt, mivel assembler nyelven készültek: az órát programból vagy valame-

lyik billentyű lenyomásával lehetett elindítani, és ugyanilyen módon kellett megállítani. Az igényesebb felhasználást azonban akadályozta, hogy a leállítás, majd az újbóli indítás után az óra nem a képernyőn kijelzett időről indult.

Eredetileg olyan programot akartam készíteni, amely megfelel az alábbi követelményeknek:

– assembly nyelven megírva két óra váltakozva működjön;

– az órák a képernyőn kijelzett (leállított) időértékről induljanak; programból és billentyűzetről egyaránt működtethetők legyenek;

– a két óra közül bármelyik indítható legyen elsőként,

– újbóli betöltés nélkül az órák ismét aktiválhatók és hívhatók legyenek.

A Mikroszámitógép Magazin 1985/6. számában Müller Gábor –Laky Zsolt „Megszakítás I.” című cikkükben a szerzők bemutatják, hogy a CIA-chipek megszakítási rendszerét miként használhatjuk fel saját céljainkra. Bevezetik az ENGED bajt fogalmát, melynek programozásával az óra indítása-leállítása vezérelhető.

Azzal, hogy az előbb leállított óra az újraindításkor milyen időértékről indul, a szerzők nem foglalkoztak, mivel céljuk az ébresztés programozásának ismertetése volt.

A „Megszakítás I.” című programot tüzetesen áttanulmányozva arra a következtetésre jutottam, hogy az ENGED bajt programozási módszere alkalmas lehet két, váltakozva működő óra vezérlésére. Kísérletezni kezdtem, és végül sikerült a fenti követelményeknek megfelelő programot írnom.

A program a CIA#1 és a CIA#2 chip óráját működteti. Két megszakítást tartalmaz. Az egyik az óratun aktivizálásához, a másik a KERNAL visszakapcsolásához szükséges. Betöltés után az óraprogramot SYS 49164 parancs aktivizálja. Ekkor a képernyő bal és jobb felső sarkában keretbe foglalva megjelenik a „B” és az „A” óra θ-ra beállított kezdőértéke. Ezt követően választhatunk, melyik órát indítjuk. Az FI gomb lenyomására az „A”, az F3-ra a „B” óra indul elsőnek. A két billentyű váltakozó lenyomásával az éppen működő órát leállíthatjuk, a másikat újraindíthatjuk. A „hívott” óra a képernyőn kijelzett időértéktől számol tovább.

A SYS 49161 parancsra a működő óra leáll. A SYS 49164 ismét aktivizál, és az előbbi folyamat megismételhető.

Az óraprogram játékok készítéséhez jól használható. A SYS parancsok a játékprogramokba utasításként beépíthetők, az FI és az F3 billentyűk lenyomása pedig szimulálható. Mint ismeretes, az éppen lenyomott billentyű kódszámát a C\$C(197) cím tartalmazza. Ezért, ha programunkban a szükséges helyekre a POKE 197,4 majd később a POKE 197,5 utasításokat beírjuk, az „A” óra fog elindulni, és a „B” órára akkor fog átváltani, amikor terveztük.

Mindkét óra zavartalanul működik 0 és 12 óra között. Az órák időértéke programból vagy billentyűzetről állítható be. Például POKE 49152, 17:POKE 49154,89:POKE 49156,86 beírása 11 óra, 59 perc, 56 másodpercet jelez ki a képernyőn. (Az óra beállítására lásd a fent hivatkozott cikket.)

ADROVICZ ISTVÁN

```

0 REM *** KET ORA A JATEKOKHOZ ***
1
100 FORI=49152:049642
110 READX:POKEI,X,S=5+X:NEMT
120 DATA 0,0,0,0,0,0,1,1,1,76,155,192
130 DATA 120,173,20,3,141,233,193,173,21,3,141,234
140 DATA 193,169,181,141,20,3,169,192,141,21,3,169
150 DATA 128,13,14,220,141,14,220,169,128,13,14,221
160 DATA 141,14,221,173,0,192,141,11,220,173,1,192
170 DATA 141,11,221,173,2,192,141,10,220,173,3,192
180 DATA 141,10,221,173,4,192,141,9,220,173,5,192
190 DATA 141,9,221,169,58,141,34,4,141,2,4,141
200 DATA 37,4,141,5,4,162,9,169,1,157,30,216
210 DATA 157,255,215,157,70,216,157,39,216,202,208,241
220 DATA 169,93,141,8,4,141,31,4,169,109,141,71
230 DATA 4,169,125,141,48,4,169,64,162,8,157,71
240 DATA 4,157,39,4,202,208,247,88,96,49,234,169
250 DATA 1,141,6,192,141,7,192,141,8,192,120,173
260 DATA 153,192,141,20,3,173,154,192,141,21,3,88
270 DATA 96,165,197,201,4,208,24,169,1,141,6,192
280 DATA 173,7,192,201,1,208,3,32,244,192,173,8
290 DATA 220,141,8,220,76,15,193,201,5,208,5,169
300 DATA 0,141,6,192,173,6,192,208,46,173,7,192
310 DATA 201,0,208,3,32,109,193,173,8,221,141,8
320 DATA 221,76,136,193,173,11,221,141,11,221,173,10
330 DATA 221,141,10,221,173,9,221,141,9,221,173,8
340 DATA 221,169,0,141,7,192,96,173,11,220,41,16
350 DATA 208,8,169,48,141,32,4,76,35,193,169,49
360 DATA 141,32,4,173,11,220,41,15,24,105,48,141
370 DATA 33,4,173,10,220,41,240,74,74,74,105
380 DATA 48,141,35,4,173,10,220,41,15,24,105,48
390 DATA 141,36,4,173,9,220,41,240,74,74,74,74
400 DATA 105,48,141,38,4,173,9,220,41,15,24,105
410 DATA 48,141,39,4,173,8,220,173,8,192,201,1
420 DATA 240,30,76,232,193,173,11,220,141,11,220,173
430 DATA 10,220,141,10,220,173,9,220,141,9,220,173
440 DATA 8,220,169,1,141,7,192,96,173,11,221,41
450 DATA 16,208,8,169,48,141,0,4,76,156,193,169
460 DATA 49,141,0,4,173,11,221,41,15,24,105,48
470 DATA 141,1,4,173,10,221,41,240,74,74,74,74
480 DATA 105,48,141,3,4,173,10,221,41,15,24,105
490 DATA 48,141,4,4,173,9,221,41,240,74,74,74
500 DATA 74,105,48,141,6,4,173,9,221,41,15,24
510 DATA 105,48,141,7,4,173,8,221,173,8,192,201
520 DATA 1,208,5,169,0,141,8,192,76,254,255
530 IFS<>51300THENPRINT"MIHIRA A DATASOROKBAN...!":END
540 PRINT"JAFENDBEN...!"
    
```

LASER—210

Labirintus

A négy irányjelző gomb segítségével végig kell vezetni a harcost jelképező képpontot a labirintuson, és távozni kell a kapun. Falhoz ütközés esetén a pont kimerededik (meghal), és indul az új harcos, természetese-

sen előlről. Négy kapun kell átjutni, amelyeket szörnyek vigyáznak. Ha a szörnyeket legyőzték, a kapuk kinyílnak. Tíz harcos élete áll egy játékos rendelkezésére.

BORZA VIKTOR

```

4 CLS:PRINT
5 PRINT "A HALADÁS IRÁNYA GOMBNYOMÁSSAL VÁLTOZTATHATÓ"
10 PRINT "A HÁLÓKON A SZÖRNYEK LEGYŐZÉSÉVEL LEHET ÁTJUTNI"
20 PRINT "A FALNAK ÜTKÖZŐ HARCOS MEGHAL ES A KAPUK BEZÁRULNAK"
25 PRINT "A HALADÁS SÉRÉSSÉGE 1-TŐL 5-IG VALASZTHATÓ"
30 PRINT
35 INPUT "A KIVANT SÉBESSÉGFOKOZAT:1A1
40 CB=125-25*A1
45 MDDE(1):COLOR2
50 FOR I=1 TO 126
55 SET(1,0):SET(1,63)
60 NEXT I
65 FOR I=0 TO 63
70 SET(0,I):SET(127,I)
75 NEXT I:AO=-29696
77 REM VIZSZINTES FALAK
80 POKE 30862,1:POKE 30863,140
85 FOR I=1 TO 22
90 READ D
95 POKE AO+I,D
100 NEXT I
105 DATA 62,3,33,00,00,17,32,0,6,00,54,85,35,27,5,32,249,25,61
110 DATA 32,240,201
115 FOR I=1 TO 21
120 READ A1,A2,A3
125 POKE AO+4,A1:POKE AO+5,A2:POKEAO+10,A3
130 A=USR(B):NEXT I
135 DATA 192,112,3,197,112,4,203,112,4,209,112,6,218,112,6
140 DATA 201,113,8,218,113,3,222,113,2,192,114,2,204,114,4
145 DATA 215,114,11,194,115,5,215,115,5,222,115,2
150 DATA 196,116,3,206,116,8,222,116,2,198,117,17
155 DATA 222,117,2,136,118,7,73,119,8
157 REM FÜGGŐLEGES FALAK
160 FOR I=1 TO 15
165 READ D
170 POKE AO+I,D
175 NEXT I
180 DATA 33,00,00,17,32,0,6,00,54,85,25,5,32,250,201
185 FOR I=1 TO 18
190 READ A1,A2,A3
195 POKE AO+2,A1:POKE AO+3,A2:POKE AO+8,A3
200 A=USR(B):NEXT I
202 DATA 194,113,12,34,116,36,229,113,15,197,117,14,135,118,11
204 DATA 40,115,37,43,112,5,205,114,20,46,115,13,49,113,25
206 DATA 67,118,11,116,115,27,148,118,10,55,112,13,215,115,22
208 DATA 218,113,6,58,116,25,61,113,49
209 REM KIJÁRAT, AJÁRUK
210 FOR K=1 TO 10
212 READ X,Y
215 FOR I=0 TO 3
220 RESET(X,Y+1)
225 NEXT I:NEXT K
226 DATA 29,46,32,52,33,52,67,58,120,14,120,30,120,46,126,6,126
227 DATA 22,126,38
228 RESET(127,3)
229 REM KAPUK:
230 FOR I=11 TO 20:SET(1,7):NEXT I
235 FOR I=107 TO 116:SET(1,56):NEXT I
240 FOR I=8 TO 15:SET(22,1):NEXT I
245 FOR I=41 TO 46:SET(46,1):NEXT I
249 REM SZÖRNYEK
250 COLOR3
255 SET(4,22):SET(47,3):SET(100,32)
260 COLOR4:SET(52,24):COLOR2
264 REM INDUL A HARCOS
265 M=A:N=62
270 A=3:GOTO 420
275 REM IRÁNYVÁLTÁS
280 FOR S=1 TO CB:NEXT S
300 IF INKEY=" " THEN A=1
310 IF INKEY="M" THEN A=2
320 IF INKEY=" " THEN A=5
330 IF INKEY=" " THEN A=4
335 IF A=1 THEN 350
340 IF A=2 THEN 380
342 IF A=3 THEN 420
344 IF A=5 THEN 420
349 REM JOBBRA
350 M=M+1
352 IF POINT(M,N)>1 THEN 500
358 IF M=127 THEN 510
360 SET(M,N)
365 RESET(M-1,N)
370 GOTO 200
374 REM BALRA
380 M=M-1
385 IF POINT(M,N)<<3 THEN 390
386 GOSUB 850
387 GOSUB 400
390 IF POINT(M,N)>1 THEN 500
400 SET(M,N)
405 RESET(M+1,N)
410 GOTO 200
415 REM FELFELE
420 N=N-1
421 IF POINT(M,N)<<4 THEN 425
422 GOSUB 890
423 GOTO 440
425 IF POINT(M,N)<<3 THEN 430
426 GOSUB 750
427 GOTO 440
430 IF POINT(M,N)>1 THEN 500
440 SET(M,N)
445 RESET(M,N+1)
450 SET(M,N)
455 REM LEFELE
460 N=N+1
465 IF POINT(M,N)>>3 THEN 470
466 GOSUB 790
467 GOTO 400
470 IF POINT(M,N)>1 THEN 500
480 SET(M,N)
485 RESET(M,N-1)
490 GOTO 200
495 REM HARCOS SZÁMLÁLÁS
500 G7=67+1
502 IF G7=10 THEN 720
504 GOSUB 950
505 GOTO 238
508 REM SIKERÜL A KIJUTÁS
510 PRINT " *** gratuláció! ***"
570 FOR I=1 TO 20
575 IF I<<15 THEN 580
576 FOR K=1 TO 150:NEXT K
580 READ X,Y
570 SOUND X,Y
600 NEXT I
610 DATA 23,6,24,3,26,3,26,3,24,3,23,3,21,3,19,3,19,3,21,3,23,3
620 DATA 23,5,21,1,21,4,23,6,24,3,26,3,26,3,24,3,23,3,21,3,19,3
630 DATA 19,3,21,3,21,3,21,5,19,1,19,5
640 GOTO 735
710 REM SIKERTELEN KISERLET
720 CLS:PRINT#66,"ledkoszlop talan sikerul"
730 FOR I=31 TO 1 STEP-1:GOTO I,1:NEXT I
735 END
740 REM KAPUNYITÁS ZENEVEL
750 FOR I=107 TO 116:RESET(1,56):NEXT I
760 GOSUB 900:RETURN
770 FOR I=11 TO 20:RESET(1,7):NEXT I
800 GOSUB 900:RETURN
850 FOR I=8 TO 15:RESET(22,1):NEXT I
860 GOSUB 900:RETURN
880 FOR I=41 TO 46:RESET(46,1):NEXT I
890 GOSUB 900:RETURN
900 FOR I=22 TO 26 STEP2
910 SOUND 1,1
920 NEXT I
930 SOUND 31,7
940 RETURN
945 ZENE ÜTKÖZÉSKOR
950 FOR I=30 TO 18 STEP-4
960 SOUND 1,1
970 NEXT I
980 RETURN
    
```

## Kapcsoltam?

A televízió és a rádió vetélkedőiben gyakran előfordul olyan feladat, hogy megadnak néhány számot, és megkérdezik, hogyan folytatódik a sorozat.

Például: 1, 2, 3, ?

Az emberek többsége hajlamos ilyenkor ráválni, hogy 4 a következő szám. Ha valaki azt mondaná, hogy 10, mivel:

$1 = (1-2)^2 + 2$ ,  $2 = (2-2)^2 + 2$ ,  
 $3 = (3-2)^2 + 2$ , tehát:  $10 = (4-2)^2 + 2$ , akkor felkészülhet egy kiadós vitára a játékvezetővel, hiszen annak más van a papírján.

A baj az, hogy az ilyen feladatoknak nemcsak egy-két, hanem végtelen sok helyes megoldásuk van! Sőt, tetszőlegesen folytatott sorozathoz találhatunk utólag képzési szabályt.

Tekintsük a megadott számokat egy függvény értékeinek az 1, 2, 3, 4, ... helyeken, amint egy sorozatot a természetes számok halmazán értelmezett függvénynek szokás definiálni (lásd az 1. ábrát).

Nyilván a megadott pontokon keresztül végtelen sok görbét rajzolhatunk, ezek mindegyike tekinthető képzési szabálynak. Mivel mindegy, hogy hány pontot használunk fel a függvény grafikonjának megrajzolásához, ezért felvéve a nekünk rokon-szenves — egyébként tetszőleges — folytatást, azon is átvezethetjük a görbét.

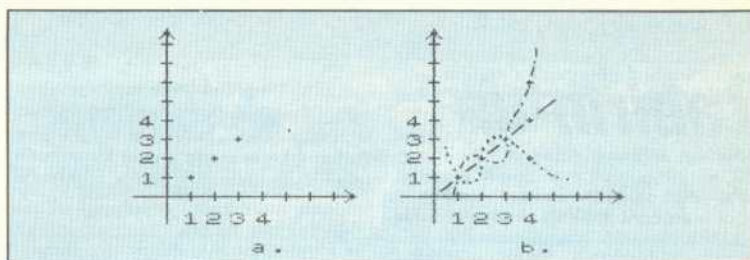
Egyesek talán nem élelgesznek meg annak tudatával, hogy létezik az adott pontonon áthaladó görbe, szeretnék látni a függvényt képletben is. Az ő és a vetélkedők résztvevőinek kedvéért készítettem a 2. ábrán látható programot. Az első DATA sorba kell beírni, hogy hány számunk van, a másodikba pedig a számokat. RUN-ra a program meghatározza azt a polinomot, amelynek helyettesítési értékei az adott számok az 1, 2, 3, ... argumentumokra.

Az általános Lagrange-interpoláció feladatát tehát speciális egyenlő közű esetre oldottam meg. A közlőt algoritmus gyors és stabil. A megoldás a következő egyszerű megmondásokon alapul. Ismert, hogy n ponton pontosan egy minimális fokszámú, legfeljebb n-1-ed fokú polinom halad át. Jelölje ezt  $y=f(x)$  ( $=a_1+a_2x^1+a_3x^2+\dots+anx^{n-1}$ ). A pontok az (1, y1), (2, y2), ... (n, yn) számpárok legyenek. Teljesülni kell tehát a következő egyenlőségeknek.

$$\begin{aligned} f(1) &= y_1 & a_1 + a_2 \cdot 1^1 + a_3 \cdot 1^2 + \dots + a_n \cdot 1^{n-1} &= y_1 \\ f(2) &= y_2 & a_1 + a_2 \cdot 2^1 + a_3 \cdot 2^2 + \dots + a_n \cdot 2^{n-1} &= y_2 \\ f(n) &= y_n & a_1 + a_2 \cdot n^1 + a_3 \cdot n^2 + \dots + a_n \cdot n^{n-1} &= y_n \end{aligned}$$

Az  $a_1, a_2, a_3, \dots$  ismeretlenekre, a keresett polinom együtthatóira tehát adott az n egyenletből álló rendszer, amely speciális szerkezeténél fogva könnyen megoldható a Gauss-féle kiküszöböléses módszerrel. A 2. ábrán közlőt program még ezt is egyszerűsíti, rögtön a kiküszöböléses eljárás eredményeként létrejövő háromszög alakú egyenletrendszert írja fel és oldja meg.

A 3/a, b, c ábrán látható futási eredmények az 1, 2, 3, 4, az 1, 2, 3, 10 és az 1, 2, 3,



1. ábra

```

10>PRINT "Values for interpolation : "
PRINT
20 READ n
30 DIM g(n)
DIM a(n,n)
40 FOR j=1 TO n
50 READ g(j)
60 PRINT j,g(j)
70 LET a(1,j)=1
80 FOR i=2 TO j
90 LET a(i,j)=i*a(i,j-1)+a(i-1,j-1)
100 LET g(j)=(g(j)-g(i-1))/(j-i+1)
110 NEXT i
120 NEXT j
130 PRINT
PRINT "The interpolated polinom : "
PRINT
140 FOR i=n TO 1 STEP -1
150 FOR j=i+1 TO n
160 LET g(i)=g(i)-a(i,j)*g(j)
170 NEXT j
180 PRINT g(i),"*x^";i-1
190 NEXT i
200 DATA 4
210 DATA 1,2,3,22
    
```

2. ábra

3/a ábra

```

Values for interpolation :
1 1
2 2
3 3
4 22

The interpolated polinom :
3 *x^3
-18 *x^2
34 *x^1
-18 *x^0
    
```

```

Values for interpolation :
1 1
2 2
3 3
4 10

The interpolated polinom :
1 *x^3
-6 *x^2
12 *x^1
-6 *x^0
    
```

3/b ábra

3/c ábra

```

Values for interpolation :
1 1
2 2
3 3
4 4

The interpolated polinom :
0 *x^3
0 *x^2
1 *x^1
0 *x^0
    
```

22 sorozatokhoz készült interpolációs polinomokat mutatják. Bármely számmal is folytatunk egy sorozatot, a program megmagyarázza választásunkat!

A számtalan függvénytypus közül egyszerűségük miatt a polinomokat választottuk. Ebben az értelemben teljesítettük azt a meg nem fogalmazott kikötést is, hogy a képzési szabály ne legyen bonyolult. Ha valakinek az egyszerűség itteni kritériuma nem felel meg, gondolkodjon kissé, s rájön, hogy az egyszerűség meghatározása milyen bonyolult.

Mindebből az következik, hogy a „Hogyan folytatódik...?” kérdésre a legprecízebb válasz: „Bárhogy!”.

LOVRICS LÁSZLÓ

**Nagyon sok olvasónk küld géppel vagy kézzel írott programokat. Ezeket sajnos nem mindig tudjuk letesztelni, mert arra nincs lehetőségünk, hogy három-négy oldalas programokat gépbe billentyűzzünk. Akik azt szeretnék, hogy programjaikat közöljük, azok ne adathordozón küldjék el munkájukat, hanem nyomtatott és nem géppel vagy kézzel írott programlistán (egy sorban max. 40 karakter), mert csak így tudunk elfogadhatóan tördelt oldalakat szerkeszteni. Előre is köszönöm az olvasók szíves megértését és aktivitását.**

*Iff. Szadai József, Nyékládháza,*

*Kandó K. út 14. 3433*

Ne haragudjon, hogy levelemmel zavarom, de van néhány olyan problémám, amelyre nem tudok választ kapni, de remélem, hogy Maga, vagy ismerősei meg tudják magyarázni azokat a dolgokat, amelyet tapasztaltam.

Negyedik osztályos tanuló vagyok, a leninvárosi Kun Béla középiskola vegyész szakán tanuló.

Júliusban kaptam egy COMMODORE 64-es számítógépet, hozzá tartozó magnóval. Előtte már jól megtanultam a BASIC programnyelvet és a 6510-es mikroprocesszor assembly nyelvét.

Édesapám munkatársától kaptam három jó programot: a PROFI—ASS 64-et, a SUPERGRAPHIK 64-et és a SIMON'S BASIC-et. Sajnos az örömmöm nem sokáig tartott: a SIMON'S BASIC program jól működött, de a másik kettő csődöt mondott. A PROFI—ASS 64 assemblerről azt olvastam, hogy magnóval is működik, nem kell hozzá lemezegység. A betöltés után azonban, amint egy sort beütöttem, az „OUT OF MEMORY ERROR” hibaüzenetet adta. FRE(0)-val lekérdeztem a szabad byte-ok számát, amelynek értéke —4 lett. Sokat próbálkoztam, de eredménytelenül.

A másik programnál más volt a baj. A betöltő programot betöltöttem és RUN-nal elindítottam. Törölte a képernyőt és kiírta a READY-t. A betöltő programnak kellett volna betölteni a másik két főprogramot, a GRAPHIC IEC-t és a GRAPHIC NOR-t. De sajnos nem történt semmi. Akitől a programot kaptam, megkérdeztem, hogy mi lehet a baj. Véleményem szerint az a baj, hogy mivel neki lemezen van ez a két program (a PROFI—ASS 64 és a SUPERGRAPHIK 64), lehet hogy az a baj, hogy a betöltő programban a 8-as lemezegységszám van, és erre, pontosabban a lemezegységre vonatkozik, szemben a magnó 1-es azonosító számával. Szerinte tehát azokat a tárcimeket, amelyen a lemezegy-

ségre vonatkozó 8-as szerepel, 1-esre kell átírni, és akkor működni fog. De szerintem, ha így lenne, akkor a „DEVICE NOT PRESENT” hibajelzést kéne hogy adja, mivel a lemezegység nincs a géphez kapcsolva.

Mivel tudomásom szerint többféle SUPERGRAPHIC 64 program van, leírom, hogy az enyém, amelyiben működne, hogy jelentkezne be:

SUPERGRAPHIK CBM 64  
(C) 1983 BY DATA BECKER  
ENTWICKELT ZUM 03. 10. 1983  
VON AXEL TENGE — MEERBUSCH

Várom azoknak a Commodore tudós olvasóinknak a jelentkezését, akik vállalkoznak a távdiagnosztikára.

*Gutai Zsolt, Jászberény,*

*Hunyadi J. u. 30. 5100*

A nyár végén hozzájutottam egy Commodore 128-as típusú számítógéphez, azonban az a hiba, hogy a gépkönyv német nyelvű. Mivel sem én, sem barátaim nem tudnak ilyen szakszöveget lefordítani, ezért olvastam egy régebbi (1985/5.)  $\mu$ M számában örömmel azt a levelet, amelyben a Babilon Nyelvtudó írja az olvasóknak, hogy vállalkoznak kezelési utasítások magyar nyelvére lefordítására is. Ez így rendben is volna, csak hogy miután feladtam a levelet a Babilon Nyelvtudó részére, kb. 1 hét múlva visszakajtott a levél, mert a címzett a Posta által ismeretlen helyre költözött címszóval a posta visszahozta a levelet.

Ezért kérem az Ön segítségét, hogy ha lehetősége van rá, az alábbi borítékra írva a nyelvtudó címét, a részükre elküldeni szíveskedjék.

*Elküldték.*

*Halmi Zsolt, Budaörs*

*Kassai u. 3. 2040*

Minden hónapban a megjelenés időpontjában megveszem a Mikroszámítógép Magazin. Az árával semmi problémám. Érdekes, szerteágazó, széles körű, jól meg szerkesztett lap. Én magam túl vagyok a tizenúton. Alig két éve foglalkozom számítástechnikával, de eléggé kezdetleges módon. A számítógépet egy BASIC-tanfolyam ismertem meg közelebbről. Primőn tanultam, de már akkor beleszerettem a Spectrumba. Barátom unszorgatására és jómagam ösztönzésére ez év májusában vásároltam is egyet. Azóta irtam két kalandos témájú programot, és kezdtől a programozást kedvelem. Játsszani csak pihentetőül szoktam.

A Magazin olvasva sok érdekes dolgot olvashatok, találok. Több olvasóval egyet-értésben én is keveslem a Spectrumbal kapcsolatos programokat, szoftver ismereteket, hardver ötleteket. „Commodore centrikusság” jellemző a lapra. A gépi kódú ismertető cikk is Commodore gépre íródott. Sajnos!

Szeretnék a lapban több Spectrum-cikket olvasni.

Folyamatosan kapjuk a Spectrum programokat keveslő leveleket. Lassan azt hiszem, hogy a Spectrum-tulajdonosok csak olvasni szeretnek, de programot írni nem, míg a Commodore-tulajdonosoknál a helyzet egészen fordított. Küldjenek jó Spectrum programokat, közzölni fogjuk.

*Dr. Tikász Attila, Berettyóújfalva,*

*József A. u. 5. 4100*

A 10. számban olvastam Képiró Róbert levelét, illetve az erre adott választ a Spectrum-tulajdonosokkal kapcsolatban. Használó tartalmú levél az Ölet szeptemberi számában is megjelent. Ott is hasonló volt a szerkesztőség reagálása.

Mint „Spectrumos” szintén csatlakozni tudok a levélírókhoz, és nem tudom maradtélanul elfogadni a szerkesztőség választ.

Akárhogy nézzük, a Spectrum a másik 3 gép (Commodore, HT és Primo) mellett a legelterjedtebb; ugyanakkor nem dűkálunk nagyon az irodalomban. Bárhogy is tagadnánk, a Commodore favorizált helyzetben van a többi géphez viszonyítva. Elég benneni egy nagyobb könyvesboltba és szemrevételezni a számítógépes irodalmat. Egyik kezünkön meg lehet számolni, mennyi könyv foglalkozik a Spectrumbal (a HT és Primo még rosszabb helyzetben van). E könyvek színvonalra sem olyan mint a Commodore-é. Nem sokkal mondanak többet, mint az eredeti gépkönyv.

En úgy érzem, hogy a spectrumosoknak nem az fájt, hogy kevés a programlista akár az Önök újságjában, akár az Ötleben vagy máshol, hanem kevés a szakirodalom, egy-két részletkérdéssel foglalkozó komolyabb szakkikk. Akár a 10. számot, akár a korábbi számokat nézzük, erről azonnal meg lehet győződni.

Azt hiszem, valamennyi spectrumos örülne, ha egy olyan könyvsorozatot vehetne kézhez, mint például a Data-Becker sorozat a Commodore-hoz. Külföldön van ilyen színvonalú iradalm akár német, akár angol nyelven. Egyszer jó lenne gondolni a Spectrum-tulajdonosokra is egy ilyen sorozat kiadásával.

De Önök is segíthetnének. Például nagyon elképzeltetőnek tartanám, ha megjelenik egy komolyabb cikk a HT alkalmazásával kapcsolatban (éppen a Z80-as processzorra tekintettel), közlőnk azokat a változatokat, amelyek elvégzése a Spectrumon is használható lenne a cikkben említett megoldás. Ezt az olvasók többsége nem tudja otthon megcsinálni — éppen a szakirodalom (HT) hiánya miatt.

Meggyőződésem, hogy ilyen kis segítségnek is örülnének a spectrumosok, éreznék, hogy nem annyira mellőzöttek. (En sem vagyok előfizetője lapjainknak, éppen Képiró Róbert levelében írtak miatt. Esetenként megveszem a lapot, amikor Spectrumot vagy esetleg HT-t érintő cikket találok benne.)

Lenne ellenben két javaslatom, biztosan örülnének neki a spectrumosok. Egy cikk-sorozatban jó lenne ismertetni, részleteiben írni a Spectrum rendszerváltozóról, alkalmazásának lehetőségéről. Az Ipari In-

formatikai Központ ROM-könyve is csak a 16 384. címig foglalkozik a géppel. Nagyon kevés ezekről a részletes információ. Ehhez kapcsolódóan a másik javaslat: egy összehasonlító cikk vagy táblázat a HT és a Spectrum rendszerváltozóról. A fentebb említett „átrászkokhoz” segítséget nyújtana.

**Ismét a Spectrumról.** Elgondolkoztam a levélben, hogy miért is nincs elég szakirodalom ehhez a géphez, azt hiszem rájöttem a megoldásra. A Sinclair gépek minden darabja magánimport úton került az országba. A kiutazó turisták azért vitték meg ezt a gépet, mert kb. feleannyiba került, mint a hasonló tudású Commodore. Sinclair gépet itthon legfeljebb bizományiban árultak, ennek a gépnek sohasem volt hivatalos importőre.

Nem úgy a Commodore-nak. Elsősorban a Novotrade, de az Apisz is és mások is importálók. Professzionális kereskedőkről lévén szó, jól tudták, hogy kiváló gépkönyv, alkalmazói kézikönyv és szoftver leírás nélkül a hardvert eladni nem lehet. Érdekiük volt és ma is az a könyvkiadás. Talán szükség lenne hasonló és pénzes vállalkozóra, aki megkocskoztatná néhány könyv kiadását. A Műszaki Könyvkiadó és a M. E. egyeztetve, 1985-ben kiadtuk a Votisky Zsuzsa szerkesztette „Csupa játék ZX Spectrumra” c. könyvet. Egyelőre újabb Spectrum könyvre nem vállalkozunk.

**Honti József, Csákvár,**

**Május 1. út 11. 8083**

Hosszú hallgatás után jelentkeztem ismét. E hosszú szünetnek részéről több, igen alapos oka is volt. Hadd kezdjem az elején!

Augusztus végén immár másodszer rendeztünk számítógépes bemutatót közszűnkben. A bemutatonak egyértelműen nagy sikere volt. 30 db számítógépet sikerült összeszedni, ami már magában sem kis teljesítmény. A gépek döntő része Commodore típusú volt, elsősorban 64-esek. Szerepelt a bemutatón PRIMO és természetesen itt volt a VIDEOTON TVC gépeivel. Az időn nemcsak a laikus érdeklődőkre számítottunk, hanem azokra is, akik már rendelkeznek géppel, vagy legalábbis hozzájutnak a számítógéphez. Ezért a bemutatókat a mikroszámítógép barátai első Fejér megyei találkozásának is szántuk. Sok meghívót küldtünk szét a kluboknak és sokan elfogadták meghívásunkat. Az érdeklődők között volt budapesti, hatvani, salgótarjáni és berettyóújfalui illetőségű is.

Kérésnek — hogy a találkozóról írjak beszámolót az újság részére — azért nem láttam értelmet elegendet tenni, mert rendkívül hosszúnak találok a nyomdai átfutási időt, és mire megjelenne a cikk, már régen nem aktuális a téma. Egyébként úgy vélem, hogy sokkal jobban szolgálok ezt az általam önként felvállalt ügyet, ha itt, Csákváron teszek meg minden tőlem telhető a számítástechnikai ismeretek terjesztéséért.

Hosszú hallgatásom másik oka az volt, hogy egyik barátommal egy ún. Klubmeleg összejelítésem dolgoztunk. Ez a lemez a C16-os gép lehetőségeinek jobb megismerését és hatékonyabb kihasználását kívánja elősegíteni. Tulajdonképpen igyekeztünk minden olyan alapismeretet összegyűjteni a hazai és külföldi folyóiratokból, ami segíti az ezzel a géptípussal folyó munkát. Azt, hogy sikerült-e ezt a célt elérni, vagy legalább megközelíteni — döntse el más, aki ebben nálunk avatottabb.

A lemez dokumentációját mellékeltelen megküldöm. Érdeklődés esetén — eszmetiszteletjűk ellenében — szívesen rendelkezésükre bocsátjuk és átengedjük a terjesztés jogát. Mindenképpen szeretnénk véleményt kérni a munkánkról, hiszen az segítséget és iránymutatást adna a továbbiakhoz is.

Örömmel tájékoztatom arról, hogy községünk iskolája 12 db C—116-os gépet kapott. Így a gyerekek a jövőben könnyebben hozzájutnak a számítógépekhez úgy, hogy a használatáért semmilyen címen nem kell fizetniük. Az iskola vezetésének előrelátású dicséri, hogy már az elmúlt tanév folyamán a nevelők közül a megyeszékhelyeken szervezett tanfolyam keretében felkészíteték a gyerekek oktatására. (Kár, hogy csak egy főt!)

A művelődési ház kezelésében levő 6 db számítógép és a most kapott 12 új gép már olyan komoly gépparknak számít, amire két évvel ezelőtől még álomban sem mertem gondolni.

Fentiek ismeretében a logikus lépés az lenne, ha a két gépparkot összevonnánk. Sajnos erre különböző okokból nincs lehetőség. A Művelődési Ház továbbra is ragaszkodik a saját gépeizhez, az iskola vezetése meg úgy látja, hogy a rendelkezésre álló gépeket sem fogja tudni megfelelően üzemeltetni. En az hiszem, hogy mégis csak az öszevonás lenne helyes lépés úgy, hogy az iskola adna otthont a felnőtt oktatásnak is. (Persze ezekhez a dolgokhoz nem sok közöm van, és beleszólási jogom sincs.)

**Közszűnk a beszámoló!** Valahol a település összefog az informatika társadalmatás érdekében, akkor ott csodákat lehet tenni. Csákvár példája ezt mutatja, de azt is, hogy az összefogásnak is vannak fokozatai és azon is lehet még javítani.

**Járóka László, Budapest,**

**Szív u. 3—5. fszt. 5. 1063**

Egyik lapjukban olvastam az Internacia Komputadó lap hirdetését, de a hirdetésből az derült ki, hogy 92 országban jelenik meg. Szeretném felhívni a figyelmüket, aki nem ismeri a lapot, az nem tud róla semmit, hogy hol és milyen nyelven. En tudom, és minden száma megvan. Itt szeretném felhívni a figyelmüket egy Hollandiában működő DLT-rendszerre, aminek az eszperantó nemzetközi nyelv az alapja. A DLT-rendszerrel kellene egy kis cikk, amivel ismertetni kellene, mit is jelent ez. És ha sikerült, akkor egy számítástechnikai nyelvet, mivel a — DLT — egy fordító rendszer és az alapja az „Eszperantó nyelv”. Mivel a szerkesztőségben van egy pár eszperantista, és a nyelvi problémák hamarabb megszűnnek.

Remélem, hogy a levelém megfontolásra kerül és nem szemétkosárba.

*Nem szoktuk a leveleket a szemétkosárba*

*dobni. Örülnek viszont, ha tájékoztatna bennünket, esetleg cikket is írna a DLT-rendszerről.*

**Sebestyén László, Budapest,**

**Lévai Oszkár u. 12. 1084**

Jelenleg egyetemi hallgató vagyok és sokat foglalkozom számítástechnikával. Lapjuknak rendszeres olvasója vagyok, az első évfolyamtól az összes szám megvan. Nekem egy C64-es számítógémem van, ezért főleg a C64-es programok érdekelnek. Nagyon tetszett a KERNAL rutinok ismertetése, ámbar tapasztaltam egy-két hiányosságot. Nagyon hasznos lett volna megemlíteni, hogy C64-nél a LOAD rutin nem működik, ha használatával egyazon időben sprite-okat is használunk. A sprite megjelenítése a processzor idejéből sok órácklust elvesz, így a soros busz nem tudja szinkronizálni. Ezért programbetöltésnél a sprite-okat ki kell kapcsolni. Nagyon sok fáradságomba került, mire rájöttem, ezért kérem, hogy mielőbb publikálják ezt a kiegészítést, más ne fáradjon hiába.

1986. februártól nagyon tetszik a sakkprogramozás rovatuk is. Kovács P. Attila nagyon jól bemutatja a sakkprogramok felépítését. Ámbar úgy gondolom, hogy a Bitek és figurák sorozat összegyűjtve broszuraként, vagy könnyű formátumban is megjelenhetne, mert a havonta megjelenő számokból nagyon keveset tud meg egyszerűen az olvasó, pedig aki ilyen programot szeretne írni, annak az algoritmus globális áttekintése is szüksége van. En szívesen írnék sakkprogramot, ezért egy-két problémámmal meg szeretném keresni a cikksorozat szerzőjét. Kérem segítségnek! Kérem, írják meg azt is, hogy az NJSZT által kiírt sakkprogramozási versenynek mi lett a végeredménye!

**Közszűnk a tanácsot.** Sakkprogramozás ügyben a téma magyar „pápját”, dr. Lindner Lászlót, sakkrovatunk vezetőjét tudja felhívni (tel.: 318-096).

**Gruber László, Budapest,**

**Buday László u. 7. III. 1. 1024**

... A levélért különben is csak saját magukat „okolhatják”, mivel levelezési rovatuk elején erre biztatják a „kedves olvasó”-t (Hát érdemes volt?)

1. A lapot egészében nézve igen jók és megfelelő színvonalúnak tartom, az első számtól kezdve gondosan gyűjtöm. Néhány észrevételem azonban lenne.

1.2. Némi tördelési megjegyzés. Feltételezve néhány olvasónál azt a „szentségőrtést”, hogy házi programkönyvtára érdekében — ne adj isten — olóval mérészel a lapozó nyúlón, néha igencsak bajba kerül. Ha pl. teszem azt a 86/10. számmal külön akarná választani a 25. oldalon kezdődő C64 programtíkosítást és a 27. oldali ZX sprite rajzolat, akkor bajban van, mert vagy az egyiket, vagy a másikat teszi tönkre.

Ráadásul a 28. oldal alján levő „programlista kódok” olyan táblázat, melyet ez-

után mindig használni kell, tehát központi hely „illeti meg”. De nem lehet kívágni, mert akkor mindkét említett cikk károsodik.

Nos a javaslat: Úgy kellene tördelni a lapot, hogy a programok, vagy táblázatok „hátuljára” olysmi kerüljön, amiért nem kár. Pl. hirdetés, vagy a 2., illetve a 24. oldal cikk, de lehetne a 22. oldal is, vagy — horribile dictu — a 43., 44. oldalak is.

Lehet, hogy így „széttördelődének” a rovatok, de ez az olvasót vajmi kevésbé érdekli, az előző szempont — a „szétválaszthatóság” — szerintem sokkal fontosabb lenne.

2. „Öröközöld” téma az egyes számítógépek inkompatibilitása, az egykire írt programok nem futtathatók a másikon. Nagy gyakorlattal és hozzáértéssel persze „átruházzák”, de ez nem túl egyszerű feladat.

Úgy tűnik, a probléma nem megoldhatatlan. Nekem pl. van egy programom (Floppy lemezen 104 block), amelyet C64-be betöltve és lefuttatva, a C64-et „átalakítja” ZX Spectrummá (a „bejelentkezés” is ennek megfelelő lesz) és a DATASET-be (1531) minden további nélkül behívható egy ZX programkazetta, az behívható és futtatható.

Úgy gondolom, ha ez két ilyen különböző gép esetén megoldható, akkor 2 Commodore között is.

Óriási dolog lenne egy olyan program közlése, mely a +4-be betöltve az fogadni tudná a C64 programokat!!!

Fordítva már nem tudom elképzelni a olgot, mivel a +4 többet tud a C64-nél, egy a „tanítás” nehéz lenne.

Bizonyára nem egyszerű dolog ilyen program írása, s nem tudom, hogy a szerző kellően lenne-e dotálva a közlés díjával, de én az hiszem, ha csak pénzért lehetne vásárolni, akkor is minden +4 tulajdonos boldogan megvenné, feltéve persze, hogy egyszerű áron (max. 1-2 E Ft) adnák.

3. S végül egy apróság. A C64 perifériái jök a +4-hez is. Olyannyira, hogy pl. a +4-hez árult botkormánynak C64 csatlakozója van, s külön átalakító kábelt adnak hozzá, hogy a +4 DIN csatlakozója „fogadni” tudja. Csak a DATASET-re nem érvényesek fentiek. Ráadásul, míg a C64 gépkönyve részletesen ismerteti az 1530-as bekötését, addig a +4 gépkönyve erről mélyen hallgat.

A kéréssem tehát: Ha lehetséges, közöljenek egy kis bekötési rajzot, mely egy kábel egyik végén egy DIN aljzatot (az 1531-eshez illeszkedőt) míg a másik végén egy C64 DATASET csatlakozót tartalmaz, s melynek használatával az 1531-es magno használható lenne C64-hez is.

Olvasóink leveléből részleteket közöltem, ezenkívül javaslatot tettem, hogy hirdetések helyett inkább árat emeljünk, elmondta, hogy miért nem való a lapba a Teleteaching '86-ról szóló írásom (sajnálom, hogy a cikkből nem derült ki, mi is a kapcsolat a távoktatás és a mikroszámítógépek között) és megjegyzéseket tett a Novotrade árainról. A fenti-eket éreztem a levélből közérdekűnek, el is gondolkodtunk a javaslatokon; a bekötési rajzzal kapcsolatban pedig a Commodore Klubhoz fordulunk.

Leveleiket továbbra is várja:

KOVÁCS GYÖZÖ



**Jánoki Lajos—Kocsis János:**  
**Számítógépes termelésirányítás**  
(Budapest, 1986.  
Műszaki Könyvkiadó,  
329 oldal. Ára: 90,— Ft.)

A termelés- és gyártásirányítás — különösen az utóbbi évek egyre bonyolultabbá váló gazdasági helyzetében — az ipar és a mezőgazdaság különböző ágazataiban mind több szakembert érint. Az iparvállalat mint irányított rendszer, hierarchikus rendszer, amelynek a termelésirányítás az alsó szintjét, azaz az ún. alapszintjét képezi, el látva a legfontosabb műszaki-gazdasági információi feladatokat, és ezzel megteremtve a felsőbb szintű döntések objektív alapját.

A könyv elsődleges célja, hogy megismertesse az olvasóval a termelés- és gyártásirányítás bonyolult részfeladatait (gazdaságos gyártás-előkészítés, anyagellátás, operációkutatási feladatok, üzemi irányítás, termékértékesítés, ezek információs rendszerei stb.), valamint a részfeladatok egymáshoz való kapcsolódását a hierarchikus irányítási rendszeren belül, a legkisebb területekre is kiterjedően. Így hatékony segítséget nyújt a vállalatok, termelő üzemek minden irányítási szintjén érintett vezetőknél, mérnököknek, közgazdászoknak, matematikusoknak.

Külön fejezetekben foglalkozik a vállalatvezetési információs rendszerrel, a számítógépes termelésirányító rendszer felépítésével, a termelésirányítás hardvereszközével, a termelésirányítással és készletgazdálkodással és végül bemutat egy megvalósított, hazai termelésirányító és anyaggyártó rendszer.

**Prague, Cary N.—Hammit, James E.:**  
**Adatbázis-kezelés.**  
**Programozás. dBASE III.**

(Budapest, 1986.  
Műszaki Könyvkiadó,  
273 oldal. Ára: 98,— Ft.)

A dBASE III hatékonyan alkalmazható adatbázisnyelv. Felhasználóközeli utasításokat tartalmaz, amelyek adatbázis-szerkezetek létrehozására, valamint adattételek hozzáadására, módosítására, kiválasztására és törlésére alkalmasak. A dBASE III az adatbázisban lévő rekordok keresését, kiválasztását és megjelenítését is lehetővé teszi. Segítségével könnyen lehet jelentéseket tervezni és kinyomtatni. A mikroszámítógépek egyre táguló világában gyorsan nő a programnyelv népszerűsége.

A dBASE III „teljes” programnyelv, amely alkalmas csaknem valamennyi adatfeldolgozási feladat megoldására. Az adatbázis-technika használata miatt a dBASE III egyszerűbbé teszi a programozást, mint amilyen az a hagyományos nyelvek esetén.

A könyv nemcsak az újdonsült programozókhöz szól, hanem a tapasztaltakhoz is, akik új nyelvet akarják bővíteni eddigi ismereteiket. Feltételezi az olvasórol, hogy létrehozott legalább egy adatbázist, és előállított már néhány egyszerű jelentést. A kezdő programozót a programozási alapelvekbe, az adatbázis-technikába is bevezeti. A gyakorlott programozó megfelelő útmutatót talál arra az esetre, ha az általa már ismert bármely nyelvről a dBASE III nyelvre akar áttérni.



**Dahmke, M.:**  
**Mikroszámítógépek operációs rendszerei**  
(Budapest, 1986.  
Műszaki Könyvkiadó,  
199 oldal. Ára: 79,— Ft.)

A kötet szerzője az alapoktól indulva, a szoftver oldaláról közelítőleg mondja el mindazt, amit a mikroszámítógépek operá-

dős rendszereiről a felhasználónak tudnia érdemes és kell. A legfontosabb hardver-, technológiai és szoftverfogalmak tisztázása után részletesen szól a kis, közepes és nagy rendszerek jellemzőiről. Kitér a többfelhasználós rendszerekre, a hálózatokkal, a memóriagazdálkodással, a gépfüggetlenséggel kapcsolatos ismeretekre. Az olvasó megismerkedhet több konkrét operációs rendszerrel is.

A függelékben rövid leírás található a CP/M, az MP/M és a UNIX operációs rendszerről.



**Bakos Tamás:**  
**Pascal PC-seknek**  
**(Budapest, 1986.**  
**Műszaki Könyvkiadó,**  
**161 oldal. Ára: 53,— Ft)**

A könyv ismerteti a személyi számítógépek is mindjobban terjedő, a BASIC helyébe lépő programozási nyelv, a Pascal elemeit, jellemzőit és a nyelv alkalmazásától elválaszthatatlan programozási módszereket. Míthogy a BASIC nyelv ismeretére alapozva, azzal való folyamatos összehasonlítással vezeti be a Pascal-beli új fogalmakat, a személyi számítógépek felhasználói BASIC-tudásuk alapján gyorsan elsajátíthatják az új nyelvet.

**Útmutató**  
**a számítástechnika**  
**alkalmazására**  
**vonatkozó érvényes**  
**jogszabályokról**  
**és irányelvekről**  
**(Készítette: Szabó József**  
**és Borda József)**  
**(Budapest,**  
**SALDO Pénzügyi Szervező**  
**és Tanácsadó Vállalat,**  
**1985. (1986.) 203 oldal.**  
**Ára: 362,— Ft.)**  
**A Pénzügyminisztérium**  
**hivatalos kiadványa**

A kötet tartalmazza a számítástechnika területére vonatkozó pénzügyi, számviteli,

ellenőrzési, árképzési, installálási, aktiválási, elszámolási, archiválási, adatvédelmi, adatbiztonsági szabályozásokat és magyarázatokat, mind az ágazatért felelős KSH, mind pedig a bizonylati rendet szabályozó és a működtetést, alkalmazást ellenőrző PM előírásainak megfelelően. A kötet összeállítását az a cél vezette, hogy a számítástechnikai ágazatokban alkalmazandó joganyagot fogalommagyarázatokkal ellátva, külön kiadványként adják közre, megkönnyítve ezzel a gazdálkodó és az ellenőrző szervek egységes szemléletű jogalkalmazását.

## Az ismeretlen C16

**Erdős. Iván: COMMODORE ROM lista PLUS/4, C-16, C-116 (Budapest, 1986. LSI ATSZ, 281 oldal. Ára 248,— Ft.)**

Kérdés, hogy helyesen választottam-e ki a könyv címét, ugyanis a borítón és a belső címlapon a cím négy különböző változata szerepel. Nekem ez volt a legrokonszenvesebb. A könyv az LSI „csikos” sorozathoz tartozik, és magán viseli e sorozat korábbi Commodore tárgyú kötetének ismertetőjegeit: a sok értékes új információt a hibák olyan tömegével, amely az információk használhatóságát kétségessé teszi. Nem tudom, hogy mit higgyek el, mit ne.

A könyv legnagyobb részét a közel 32 kilobájtnyi disassemblált ROM lista foglalja el. Bár a szerző több alkalommal leírta, hogy ez a lista a PLUS/4 ROM-jából készült, én valószínűbbnek tartom, hogy a másik két géptípus valamelyikéből. Egészen pontosan: jobban hasonlít a lista a C16 belső felépítésére, mint a PLUS/4-ére, de nemcsak az alkalmazói programok — a bevezetőben megindokolt — hiánya miatt.

A Műhelytitkok dióhéjban című részben megismerkedhet az olvasó a könyv elkészítéséhez kijelölt alapelvekké és megvalósításuk módjával. Megtaláljuk ebben a részben a könyv készítéséhez használt segédprogramok forráslistáját is, a PLUS/4-re (?) írt disassembler program bőségesen el van látva közbeiktatott magyarázatokkal. A programok szövege egy kicsit nehezen olvasható, látszik, hogy nem közlés céljára íródott.

A könyv függelékében keresztreferencia táblázatot találhatunk. Zavaró, hogy a hivatkozott címek rendezettek ugyan, de a hivatkozási helyek címei teljesen össze vannak keverve egy-egy hivatkozáson belül. A táblázatban sok a hamis hivatkozás; ezek többsége egy kicsit gondosabb munkával kiszűrhető lett volna. Olyan hibákra gon-

dok, mint kétszeri hivatkozás a \$FFA9 címre. Viszont nehéz elkerülni az olyan problémákat, mint például a sztring veremre történő hivatkozások egy része, mely lát-szólag — és a keresztreferencia lista szerint — a \$00 címre szól, a valódi címet az X regiszter tartalmával való indexelés adja.

Másik hibája a keresztreferencia listának, hogy sok olyan cím nem szerepel benne, melyre a ROM lista szerint van hivatkozás. Ilyenek többek között a táblázatokban elhelyezett címek.

A ROM lista készítésének egyik alapelve volt, hogy kizárólag hexadecimális számokat használt a szerző, a \$ jeleket elhagyva. Jó lett volna ehhez az elvhez ragaszkodni a magyarzó szövegek írásánál is, ahol decimális számok ugyanúgy találhatók, mint hexadecimálisak, ez utóbbiak néha \$ jellel, néha anélkül. A magyarzó szövegek másik hibája a tömörségre való törekvésből ered. Néha érthetetlenek, az is előfordul, hogy félrevezető. Példa erre a parancsmód ellenőrzését végző szubrutin hívása, ahol a „direct mode only”, illetve a „direct mode only”, ha” magyarzó szöveget olvashatjuk. Másik példa a 206. oldalról: a „Monitor hívása” és a „Monitor break” címekből nehezen lehet kitalálni, hogy tulajdonképpen miről is van szó. Az ilyen apróságoktól eltekintve a magyarázatok jók, és kitűnően eligazítják azokat, akik felszavakból is értenek.

Nagyon jó — bár nem teljesen eredeti — ötlet, hogy Lxxxx alakúak azok a címek, amelyekkel a szerző megjelöli a programnak azon pontjait, melyekre valahonnan hivatkozás jöhet. Elvértve így is előfordulnak hamis, illetve hiányzó címek, de ez a feldolgozás módjából eredően elkerülhetetlen.

A feldolgozás egyik legnehezebb része az utasítások és az egyéb információk (táblázatok, ASCII szövegek, kódok) szétválasztása. Ez az esetek túlnyomó részében kitűnően sikerült, de itt is előfordult melléfogás. Jó példa erre a Lab88 címével kezdődő három sor, melyek az előzőleg felszerelt BASIC tokenek disassemblálásából keletkeztek, a tokeneket gépi kódok néve.

A 168. oldalon furcsa hibára akadtam. Az indirekt indexelt címzési mód y-ja a hozzá tartozó vesszővel együtt a zárójelek közé került. Nem értem, hogyan történhetett ez meg.

Összegezve: akit érdekel a C16-nak és családjának lelkivilága, és nem új neki ez a téma, sok hasznos információt megtalálhat ebben a könyvben. Én is sok munkától megszabadulham volna, ha a könyv kb. 10 hónappal korábban megjelenik. Mégis azt ajánlom, hogy aki teheti, válassza a Novotrade Rt kiadásában megjelenő, hasonló tárgyú Data Becker könyvet, amelyről e sorok írásakor csak annyit tudok, hogy várható a megjelenése, de remélem, mire ez a cikk az olvasó elé kerül, már kapható lesz. Ha mégsem, akkor is érdemes megvárni.

BARNA LÁSZLÓ



## Sakkozó Elefánt

Aki híve saját sakkdátusa továbbfejlesztésének, vagy szeretné gyermekének lehetővé tenni, hogy önállóan megtanuljon sakkozni, az remek tanítómesterhez jut a Pintér József nemzetközi nagymester közreműködésével készült Sakkozó Elefánt nevű program alkalmazásával. Ez a sakkpedagógia legújabb eredményeit felhasználva, módszeresen összeállított feladványokat ad, segítséget nyújt a megoldáshoz, és rendre értékeli a tanuló tudását. Biztosítja az alapvető sakkszabályok tökéletes elsajátítását, a taktika és stratégia megismerését s a tanultak gyakorlását a beépített játékprogram segítségével.

A programcsomagot az Economix Közgazdász Egyetemi Kiszövetkezett forgalmazza a lemezmezghajtóval ellátott Commodore 64 és 128-as gépekre.

## Olcso PC

Csehszlovákiában a szolnai VÚVT 1984-ben mutatta be négy gépből álló mikroszámítógép-családját, köztük az SP-01 jelű háziszámítógépet. Az utóbbinak a billentyűházába IBM PC-vel kompatibilis gépet építettek, melyet PP-01.16 néven mutattak be. E gép külön érdekessége, hogy ez a Szovjetunióon kívül az első alkalmazása az új szovjet mikroprocesszornak, az Intel 8088-nak megfelelő KM 1810 VM88 típusnak. A gép ára meghökkentően alacsony: a billentyűzettel együtt csupán 25 ezer korona (55 ezer forint), bár a tervezett gyártási sorozat túlságosan kicsi; az idén kétszáz darabot, jövőre pedig hétézret kívánnak előállítani.



Szovjet szakértő vizsgálja a PP-01.16-ot

## Radiosoft

Jugoszláviában már 1984 óta sugároznak számítógépes programokat rádióadásokba építve. Ennek eredményeképpen a hazánk déli körzeteiben élő Commodore 64- és Spectrum-tulajdonosok jó néhány programmal gazdagodhattak. Ehhez csupán jó minőségben kazettára kellett venniük az adást, amihez egy közönséges rádiós magnetofon is elegendő felszerelés volt, s máris tölthették a friss programot a gépbe. Az „elhangozott” programokat Budapestem már csak komoly antennával lehet venni, hiszen a minőség ez esetben rendkívül fontos.

Tavaly ősszel kísérleti jelleggel a Magyar Posta is sugárzott az URH-adón számított programokat — a radiosoft nevű új szolgáltatás előkészítéseként. A megnyíló távlatok igen szélesek — mindenkinek szárnyalhat a fantáziája! —, de kézenfekvő például, hogy a lapunkban megjelenő programokat meghatározott időpontban rádióadással is lehetne terjeszteni, így mentesülne néked olvasónk a begépelés fáradtságos, sok hibával és azok keresésével járó munkájától.

## Pascal-bővítések

Világszerte növekszik a Pascal nyelv népszerűsége. Az IBM PC-vel kompatibilis gépekben futó Turbo Pascal széles körű alkalmazását követi az itthoni tendencia: ebben a gépkategóriában hazánkban is az alkalmazói szoftverek jelentős része készül ma már a Turbo Pascal fordítóval.

E programnyelvet előnyös tulajdonságai kedveltették meg a felhasználókkal. A strukturált programozás közvetett támogatásával könnyebb a programokban való hibakeresés, a későbbi módosítás; a konkrét végrehajtást a nyelv saját szerkesztője segíti. A programfordítás rendkívül gyors, maga a programírás is szaporább: sok programozó esküszik például arra, hogy egy 1000 soros CO-

BOL programot Pascalban 200-300 sorban megír.

A Turbo Pascalban programozóknak több hatékony szoftvereszköz, például az Open Access integrált adatbázis-kezelő és a Turbo Acces index-fájelkezelő rendszer is rendelkezésére áll. Az ilyen komplex szoftvereken kívül ma már szerte a világon jelennek meg a Turbo Pascalhoz forrásszinten beépíthető, adott felhasználói rendszerek fejlesztésénél jól hasznosítható eszközök is (például Nyugaton forgalmaznak egy grafikai célú gyűjteményt). Hazánkban a Számszöv Kiszövetkezett kezdeményezett ilyen irányú fejlesztéseket, főképpen a többfelhasználós rendszereket kiszolgáló programok készítésének megkönnyítésére. Alaptermékük a T RANDOM, mely rekordszintű adatvédelemmel javítja a többfelhasználós rendszerek kiszolgálását, s többek között képernyőkezelővel és fájlnyilvántartóval is ki lehet egészíteni. Ezen kívül számos autonóm működűsű, a Pascal programozást segítő eszközt kínálnak, például nyomkövető-tesztelőt, programméret-optimalizátort, maszkgenerátort és lekérdezőt, sőt magyar ékezetes Turbo Pascal is.

## Világító billentyűzet

Már kezdenek megjelenni a hazai kirakatüvegekben is az érintéses billentyűzetkivezetések. Ezeknek egy új nemzedékét jelenti a Sintz Kiszövetkezett szabadalma: olyan világító billentyűzet, melyet közönséges zsírkérréval lehet feliratozni, s a ráírt szöveg, ábra a zsírkérré árnyalatával megegyező színben fénylik. Még jobban megragadja a járókelők figyelmét, ha a zsírkérré egy ügyes grafikus kezébe adják!

A gyakran változó, gyors feliratozásra is kitűnően alkalmas ez a módszer: ez azt jelenti, hogy több szoftverterméket is futtathatunk a kirakatban, naponta akár többször is cserélve a kirakatüvegre kivezetett billentyűk értelmezését.

Az eljárás természetesen nem csupán az aktív billentyűzet, hanem passzív hirdetőtá-

lák kivitelezésére is alkalmas. Különösen a diszkrét világító-sú helyeken érvényesül a tündöklő tábla; például az elegáns éttermekben hangulatosan hívhatja magára a figyelmet a konyhafőnök ajánlata.

## Földrengésjelzés

Befejezték az ország rádiótávközlésű, automatikus földrengésjelző rendszerének kialakítását. A 17 millió forintos beruházás első szakaszában a piszkésteleti és a budapesti, most pedig az újkígyósi, valamint az ózdi állomást építették meg.

Az állomások sorra kapcsolódtak be az országos hálózatba, de eddig csak analóg jelek küldésével szolgáltatott adatokat. A most beszerelt mikroprocesszoros elektronika által már nem csupán küldenek adatokat, hanem azok fogadására is tárolására is képesek. A korszerűsítés eredményeképpen az év elejétől kezdve ezek az állomások már emelt felügyelet nélkül működnek. Automatikus azonnal jelzik és közvetítik a budapesti obszervatóriumnak az ország és a világ bármelyik részén keletkező földrengéseket. Itt a szeizmogramokat számítógéppel folyamatosan értékelik, így gyorsan megállapíthatják (becsülhetik) a rengések időpontját, helyét és erősségét.

## Forgalomirányítás

1986 őszén kezdte meg kísérleti üzemét az a rendszer, amelyet a Közlekedéstudományi Intézet kutatói dolgoztak ki Székesfehérvár jelzőlámpás csomópontjainak központi ellenőrzésére és irányítására.

A rendszerkidolgozás első szakaszában a jelzőlámpák központi felügyeletét, működésű folyamatok figyelemmel kísérését oldják meg számítógép segítségével: így szükséges a jelzőlámpás csomópontok naponkénti bejárása, megvizsgálása.

A munka második szakaszában, az idén kapcsolódik be a

számítógép a lámpák néyleges irányításába. Érzékelődetektorokat építenek az útburkolatba, melyek folyamatosan jelzik a számítógéphez a forgalmat. A gép az adott csomópontra kidolgozott programok közül automatikusan választja ki a pillanatnyi igényeknek megfelelő, optimális változatot, s ennek alapján vezérli a lámpákat. Ezáltal a közlekedés még a csúcsforgalom idején is zavartalan lehet.

A székesfehérvári számítógépes forgalomirányító rendszer parameterezhetően készül, így alkalmas lesz más, hasonló nagyságú városok forgalmának irányítására is.

## AT-kompatibilis

Megjelent a hazánkban kívüli szocialista országok első, IBM PC AT-vel kompatibilis gépe, a bolgár Intelat. A szófiai Információs Rendszerek és Rendszermérnöki Szolgáltatások nevű termelési egységül által készített gép hétféle típuskonfigurációban kerül forgalomba. A legkisebb tipikus AT-kiépítés (640 k operatív tár, 20 Mbájt Winchester-tár), a legnagyobb pedig a grafikus alkalmazásokhoz szükséges speciális perifériákkal is ellátott.

## Elektronika Indiában (Eddig még kimaradtunk...)

India elektronikai ipara az ország nemzeti jóvedelmének mintegy 0,8 százalékát képviseli; a fejlett országokban ez az arány 1–10% között mozog. India kormánya az ország 1985–90-ig tartó ötéves tervének végére ezt 3 százalékra kívánja növelni.

A fejlesztés során a hangsúlyt eleinte a licencvásárlásokra és a külföldi gyártástechnológiák átvételére helyezik. Ennek során szorosan együttműködnek a szocialista országokkal is. Az indiai elektronikai minisztérium és a kereskedelmi minisztérium különböző elektronikai vonatkozású, kétoldalú egyezményeket kötött

Csehszlovákiával, az NDK-val, Romániával és a Szovjetunióval.

## 1 kontra 64 Mbit

A Siemens és a Philips már több mint egymilliárd NSZK márka állami támogatást kapott a nagy kapacitású tár kifejlesztéséhez. Az első 1 Mbit kapacitású tár előállításához részben felhasználják a Toshiba technológiáját, és még az idén piacra viszik ezt a terméket. Ez a határidő azonban nemzetközi viszonylatban túlságosan késői. Már most tapasztalni lehet ugyanis, hogy a 256 kbit kapacitású félvezető táruk elterjedése gyorsabb lesz, mint ahogy eredetileg számították, sőt a még nem mindenütt beszerezhető 1 Mbit kapacitású RAM táruk iránti érdeklődés ideje is várhatóan rendkívül rövid lesz: minden valószínűség szerint Japán már 1989-től kezdve forgalomba fogja hozni a 64 Mbit kapacitású tárukat.

## Kerüld ki a Piccadillyt!

A brit közlekedési minisztérium laboratóriumában kisszámítógépet fejlesztettek ki, amely személyautókba, teherautókba szerelve alkalmas a forgalmi dugók jelzésére, s útmutatást is ad a vezetőnek arra, hogy miként kerülje ki a zsúfolt közutakat. A rendszer lényege, hogy az autótak mentén felszerelt különleges érzékelők figyelik a forgalom alakulását, és mérésük eredményéről rádiójelekkel tájékoztatják az autókban lévő számítógépeket, amelyek a kapott információk alapján azonnal jelzik a legkevésbé zsúfolt útvonalat. Az idén már kapható készülék kb. 150 fontba kerül, viszont — mint a brit közlekedési minisztérium illetékesei ki-fejítették — tömeges alkalmazásával legalább 100 millió fontot lehet nyerni üzemenyában, a balesetek elkerülésével és mindenekelőtt a megtakarított idővel.

## Virágzó szoftverpiac

Az USA-ban továbbra is kifejezetten kedvező a számítástechnikai szoftver piaca. 1985-ben a 30%-os növekedéssel 12,5 milliárd USD értékű forgalmat ért el. Az 1986. évi előzetes adatok szerint 16,4 milliárd dollár értékű szoftver kelt el, ami az előző évhez képest 31%-os többletnek felel meg. Különösen a személyi számítógépi szoftver területén ível fölfelé a fejlődés: üteme 55%-os, és értéken eléri a 6,2 milliárd USD-t. A szoftverforgalom negyedét az IBM bonyolítja le.

## Univerzális illesztő (magyar!)

Az ELTE Általános Technikai Tanszéke — az OMFB támogatásával és elsősorban oktatási célokra — kifejlesztette a technoMIR nevű, számítógéptípustól független, univerzális illesztőrendszert. Feladata a számítógép és a külvilág — a fizikai, kémiai, biológiai és technikai környezetünk — közötti kapcsolatteremtés. A technoMIR rendszer egységei valamilyen sajátos funkciót látnak el: a külvilágból érkező információt alakítják a számítógép által feldolgozható adatformára, vagy éppen ellenkezőleg, a gépből adat formájában érkező parancsot formálják a fizikai-kémiai folyamat befolyásolására alkalmas alakra. Az egységek külsőre azonos kialakításúak, és könnyedén egymásba dugaszolhatók. Az összekapcsolható modulok fajtáját az adott kísérlet vagy a vezérelni, szabályozni kívánt folyamat határozza meg. Sorrendjük, számuk tetszőleges, így akár egyszerre több folyamatba is beavatkozhat ugyanaz a számítógép.

A technoMIR kiválóan alkalmazható laboratóriumokban, sőt ipari alkalmazása is vonzó lehet, mivel a mintegy 4000 forintos átlagárú modulok az ipar számára árusított illesztőkhöz képest rendkívül olcsók. A rendszer a Z80 mikroprocesszorral működő mikro-

számítógépekkel együtt közvetlen alkalmazható, más típusokhoz pedig a gyártó Electroscon Kisszövetkezet illesztőmodult szállít.

Az elmúlt tanévben az Országos Pedagógiai Intézet 20 db, egyenként 10 modulból álló technoMIR készletet osztott ki az iskoláknak tapasztalatgyűjtési, módszertan-készítési kötelezettséggel. Ebben a tanévben a kedvező vélemények alapján különböző szintű oktatási intézmények — az általános iskolától az egyetemig — csupán decemberig már 200 készletet vásároltak. A technoMIR sikeres már a debreceni Kossuth Lajos Tudományegyetemen és több tanárképző főiskolán is, például Szegeden, Szombathelyen és Pécsen.

## Programvédelem PC DOS és MS-DOS alatt

Hazánkban még nem bővelkedünk a szoftvervédelemre szolgáló ajánlatokban. Ezért is jelent újdonságot a Sofinvest által forgalmazott EIGUARD nevű programvédelmi rendszer. Ez valamennyi PC DOS, illetve MS-DOS operációs rendszerrel működő IBM vagy vele kompatibilis személyi számítógépen futtatható program (.EXE és .COM fájlok) utólagos másolás, illetve illetéktelen felhasználás elleni védelmére alkalmas.

Az EIGUARD-dal védett programok, illetve a programok futtatásához szükséges kulcselemek semelyik ismert másolóprogrammal (COPY, COPYWRITE, SUPERCOPY stb.) sem másolhatók eredményesen, és a kulcselemek hiányában az eddig bevált védelemelhardtó rendszerekkel (ZERODISK, NOKEY, UNGUARD, NOGUARD stb.) sem indíthatók el.

Az EIGUARD védelmi rendszer alkalmas egy vagy több program azonos vagy egymástól elkülönülő védelmére is. Jól használhatják mindazok, akik az általuk készített vagy forgalmazott szoftver valamennyi példányát óvni akarják a másolástól.



A ZEBRÁT AZ

RAJZOLÓGÉPE

KÉSZÍTETTE:



Cikkünk a 16. oldalon

