

mikro

számítógép

magazin

Ára: 30 Ft



AJÁNDÉK LÓNAK...

1988/7



DTP

DeskTop Publishing program
+  64 konfiguráció =
saját nyomda

Cikkünk a 26. oldalon

A NEUMANN JÁNOS SZÁMÍTÓGÉPTUDOMÁNYI TÁRSASÁG LAPJA

A szerkesztőbizottság vezetője:
Kovács Győző

A szerkesztőség munkatársai:
Babos János (tervezőszerkesztő)
Bakos Tamás (programozástechnika)
Broczkó Péter (hírek)
Kovács Győző (levelezés)
Petróczy Judit (könyvek)
Pinke György (NJSZT, alkalmazások)
Simonyi Endre
Szabenszki Sándor
Szulyovszky Csaba
Tamásné Lakó Erika
Terebessy Ákosné
Vizessy Mária

Címképünk:
Kiss Ilona munkája

 mikro számítógép
magazin



Felelős szerkesztő:
Könyves Tóth Pál

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

Levélcím:
1371 Budapest
Pf. 433.

Kiadja az Ifjúsági Lap-
és Könyvkiadó Vállalat

Felelős kiadó:
dr. Király G. István
igazgató

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

Terjeszti a Magyar Posta
Előfizethető a hírlapkézbesítő
hivataloknál
és a Posta Hírlap-előfizetési
és Lapellátási Irodáján
(1900 Budapest XIII.,
Lehel u. 10/A)
vagy átutalással a 215-96 162
pénzforgalmi jelzőszámra.

Megjelenik havonta.
Egy szám ára 30,— Ft
Előfizetési díj:
egy évre 360,— Ft
fél évre 180,— Ft
Külföldön terjeszti
a Kultúra,
1389 Budapest, Pf. 149.
és a Magyar Média
1932 Budapest, Pf. 279.
86-0253



Szika Lapnyomda
Budapest (88-0785)
Felelős vezető:
Csöndes Zoltán vezérigazgató

INDEX: 25 629
ISSN 0236-6088

Tartalom

Szekszárd 1988	2
Feladatok — megoldások	7
Rendszerfejlesztési eszközök	22
Ajándék lónak (ne) nézd a fogát!	24
A Printfox és a Pagefox	26
Geopublish, avagy kiadvány készítése Geos programmal	28
Az Enterprise-zal hosszú távon számolhatunk	29
Egy sarokkal olcsóbb!	32
Közprogramok	33
Merre tart a világ?	34
μINFORM	39
Mit tud a QUALIGRAPH?	40
Programtermék	45
Adok — veszek — cserélek	46

ISKOLA-SZÁMÍTÓGÉP

Programírás felsőfokon	3
A Nemes Tihamér Országos Középiskolai Számítástechnikai Verseny döntője	4
TechnoMIR	5
Ékezetes névsorok rendezése	6

DIÁKROVAT

Zenei segédlet	8
TOP-lista	8
Primo lapok	9
Spectrum INPUT rutin TVC-re	10
Hangdigitalizálás	11
Örökéletkód	11

PROGRAMOZÁSTECHNIKA

BASIC és gépi kód	12
A Precomp Plus előfordító program	14
C nyelv. Tanulás függvényekkel	17
Fordítás vagy értelmezés?	18
Örökmozgó szellemek	20

μPROGRAMOK

Fájlok feldolgozása magnóval	37
------------------------------	----

μKLUB

Adom a magyarázatot!	38
Ki ad magyarázatot?	38

SAKK

Gyalogvégjáték	42
----------------	----

AZ OLVASÓ ÍRJA

	43
--	----

KÖNYVEK

	46
--	----

HÍREK, ÉRDEKESSÉGEK

	47
--	----

PONTVADÁSZAT

	48
--	----

Szekszárd 1988

„Szekszárd rendezett tanácsú város, vármegyei székhely Tolna vármegyében (1920), 14 025 lakos magyar lakosságból 11 244 római katolikus, 1762 református, 288 ágostoni evangélikus és 675 izraelita. A rendszeresen épült város nevezetesebb épületei: a vármegyeháza, a városháza, törvényszéki palota, állami gimnázium, múzeum, római katolikus templom stb. A vármegye közigazgatási hivatalainak stb. székhelye, van selyemnyésztési felügyelője, polgári fiú- és leányiskolája, kereskedelmi fiúiskolája, bábaképzője, kórháza, többféle kulturális és jótékonyági egyesülete stb. Lakói jelenleg selyemnyésztők, gyümölcs- és bortermelők. Szekszárd a római időkben Alison. Az I. Béla-tól (1061) alapított bencés apátságát Mária Terézia (1777) a budapesti egyetemnek ajándékozta. 1812. Szekszárdon született Garay János, a költő. Emlékét szülőházán emléktáblával, a Garay téren szoborral örökítették meg.” (Tolnai Világlexikona XVI. kötet, 1929.)

Budapesten kívül, ahol már — kimondani is sok — harminchét éve élek, két város áll hozzám nagyon közel. Szekszárd, ahol születtem és ahol a középiskolát befejeztem, aztán Baja, ahol szüleimmel gyönyörűséges tíz évet töltöttünk, ott tanultam meg a betűvetést és kóstoltam bele a „magasabb” tudományokba a III. Béla Gimnázium első négy osztályában.

A szülőföld szeretete feltétel nélküli és állandó, még akkor is, ha volt egy időszak, amikor nem szívesen mentem „haza” Szekszárdra. Nehezen viseltem el ugyanis az Al-sóváros kedves házait — köztük a szülőházamat is — romboló gépek látványát és nem kevésbé a helyére épült, a szememnek ma is idegen betontömeg felépülését a maga jellegtelenségével, rideg falaival, falanszterszerű utcáival. Már csak azt remélem, hogy az Újváros és a Felsőváros szép paraszti portái megmenekülnek a rombolástól. Nem szívesen nézek fel a Kálváriára sem, ahol régen a lemenő nap fényében csak a három kereszt archaikus hangulatot árasztó sziluettjét lehetett látni, ma pedig egy nekem sok szempontból nagyon izléstelen szoborkompozíció és egy oda nem illő tv-adó antennája teszi idegenné a Felsővárost szeliden takaró hegyek vonulatát.

Ezért aztán sokáig csak a tízévenkénti érettségi találkozó és egy-egy rövid rokonlátogatás alkalmából jutottam el a városba, ifjúságom egyre fogyó emlékeit keresgélve. Ilyenkor aztán szívesen elméláztam a Garay téren, örültem a Béla téri épületek harmonikus együttesének, a városházának, a templomnak és a megyeházához.

Persze nagy és gyönyörű élmény belépni a gimnázium jól ismert épületébe, ahonnan — sajnos — kedves tanárim ma már hiányoznak. Közülük is az örökre eltávozottak hiányát a legnehezebb elviselni, hiszen szellemük egyre kedvesebb, alakjuk örökre összefonódott az iskolával.

Azt hiszem, a harmincéves érettségi találkozóon beszélgettem először Zentai András-sal, a gimnázium igazgatójával. Elmondta, hogy szeretné, ha az öregdiákok és az iskola között a jövőben rendszeres kapcsolat alakulna ki. Mindketten úgy véltük, hogy ez az együttműködés hasznos lehetne mindenki számára. Mindjárt meg is hívott,

hogy a legközelebbi Garay-napokon tartsak előadást a gimnázium diákjainak a hazai számítástechnikáról. Arra már nem nagyon emlékszem, hogy miről is beszéltem, de arra igen, hogy igen nagy volt a diákok érdeklődése a téma iránt.

A diákprogram után az igazgatói irodában folytattuk. Arról volt szó, hogy nem volna szabad lanyhulni hagyni az éppen felkeltett érdeklődést, tennünk kellene valamit annak érdekében, hogy a szekszárdi diákok számítástechnika iránti lelkesedése ne csak néhány napig tartson. Meg azt sem tartottuk elképzelhetetlennek, hogy a Garay Gimnázium az iskolai számítástechnika egyik bázisa legyen.

Akkor és ott határoztuk el a Garay-verseny megrendezését, méghozzá az öt évvel ezelőtt nagyon divatos játékprogram-kategóriában. Ugy döntöttünk, hogy országos versenyt rendezünk, a pályázatok közül először a Mikroszámítógép Magazin diáksúrije választja ki a legjobb pályaműveket, majd a kiválasztottak közül a szekszárdi döntőn az NJSZT, a µM és a Garay Gimnázium közös zsürije dönt a helyezésekről. Azt is elhatároztuk, hogy a verseny díjait az alapítók adják. Nem vártként a gimnázium közönségdíjat ajánlott fel, amelyet a zsüritől függetlenül a diákok adnak azóta is a nekik legjobban tetsző programok alkotójának.

Az első versenyen szekszárdi gimnazista még nem kapott díjat, azokat az ország különböző iskoláiból érkezett versenyzők nyerték el. A verseny azonban elérte a célját, azaz az iskolában folyamatosan nőtt az érdeklődés a számítástechnika iránt, amit semmi nem bizonyít jobban, mint az a tény, hogy az utóbbi versenyeken a hazai versenyzők már nemcsak jó helyezéseket, de több első díjat is elvittek.

A Garay-verseny azonban nem csak a gimnáziumban, hanem a város többi iskolájában is éreztette a hatását, hiszen a döntőn a gimnázium más iskolák diákjait is szívesen látja. Arról nem is beszélve, hogy a Garay-verseny résztvevőinek díjnyertes munkáit az Országos mikroszámítógépes találkozón is bemutatták, és ezzel a nyilvános szerepléssel a nyertesek neve országosan is ismertté válik. Nem kis örömet okoz számomra, hogy a nagy számítástechnikai versenyek, mint például a Nemes Tihamér Országos Tanulmányi Verseny döntőinek résztvevőit nézve, nagyon sokszor találkozom a Garay-versenyen díjat nyert diákok nevével, akik sokszor Szekszárdon kezdték szárnyaikat bontogatni.

Néhány éve a versenyre már nem csak játékprogram, de oktató-, vagy ahogy én szívesebben nevezem, távtanulási program kategóriában is lehet pályázni. Még nem tudtam pontosan megmagyarázni, miért, de örvendetes módon a távtanulási programok száma és színvonala évről évre folyamatosan emelkedik. Persze a játékprogramok színvonala sem csökken, hiszen min-

den évben találunk nagyon sok érdekes, jó és főleg eredeti ötletekkel teli játékprogramot is. Az idei verseny végeredményére most nem térnék ki, hiszen erről egy részletes cikkben számolunk be itt, a Mikromagazinban. Azt azonban elmondhatom, hogy az eddig rendezett öt verseny után — a jövő évben — ismét újtásra készülünk. Először is szeretnénk a díjak értékét jelentősen emelni, és nem csak a helyezetteket díjazni, hanem a különleges teljesítményeket (pl. dokumentáció, ötletesség, technológizáltság stb.) különdíjjal is jutalmazni. Arról is szó volt, hogy újabb kategóriákat is jelölünk, hiszen már az idén is vita volt, hogy a távtanulási kategóriában győztes szöveg-szerkesztő program odailik-e vagy sem? Azon is gondolkodtunk, hogy néhány versenyzőt külföldről is meghívnaunk, így aztán össze tudnánk mérni a mi diákjaink erejét más országok diákjainak teljesítményével.

A következő évi Garay-verseny pályázati felhívását igyekszünk gyorsan elkészíteni, hogy a nyár végéig a pályázatot meghirdethessük. A versenyt arra is felhasználtam, hogy a Szekszárdra meghívott számítástechnikai szakfelügyelővel, a Garay tanárral és persze az odajött diákokkal elbeszélgessek az iskolai számítástechnika tapasztalatairól, friss információkkal egészítettem ki múlt évi ismereteimet. Úgy tapasztaltam, hogy egy év óta a helyzet nem sokat változott, a jelenlegi iskolai számítógépeket — a tanárok is és a diákok is — szinte teljes mértékben programozásra használják, a gép gyakorlatilag még sehol sem tudott az iskolai oktatás, illetve a tanuló segédeszközévé válni. A számítástechnikához nem értő tanárok közül csak nagyon kevesen hajlandók az órán vagy az órán kívüli tanulásban az oktatóprogramokat alkalmazni, miután ezeket a programokat nagyon mereveknek, sokszor pedagógiailag alkalmatlannak minősítik, és így az alkalmazásukat kerek percc megtagadják. A beszélgetések ismételten megerősítették azt a véleményemet, hogy az oktatási kormányzatnak nagyon határozottan újra kell gondolnia az iskolaszámítógép-programot, és az eddigi hardver szemlélet helyett erőteljesen az alkalmazásra kell a hangsúlyt helyezni. Az alkalmazás pedig ma már jó technológia nélkül elképzelhetetlen, így a meglévő pénzeszközöket a gépvásárlás mellett valószínűleg technológiavásárlásra kellene fordítani. Olyan technológia beszerzésére kell törekedni, amelynek a segítségével már nem csak az ügyes programozók, de a programozáshoz egyáltalán nem értő gyerekek és főleg pedagógusok is képesek lehessenek számítógépes tanítási anyagot készíteni.

Ha a tervünk sikerül, akkor a következő Garay-versenyek valamelyikén már nem csak a programozni tudó, de a számítástechnikát ügyesen alkalmazó diákok is vették fogognak az ideizhez hasonló magas szintű alkotásokkal.

KOVÁCS GYÖZŐ

Programírás felsőfokon

Öt év nagy idő! Az első jubileum! Ötödik alkalommal rendezte meg ugyanis a már „Garay Pályázat” néven emlegetett versenyt az NJSZT, a szekszárdi Garay János Gimnázium és a Mikro-számítógép Magazin szerkesztősége. Öt év. De régen volt... S noha az emlékek nem halványulnak, a babérok nem pihenhetünk sokáig.

Idén mindössze egy hónap „türelmi idő”-t hagytunk a pályamunkák elkészítésére. Mégis szép számú csomagot hozott a postás. Összesen harminchármat. A különböző Commodore gépeken kívül HT 1080Z/64, Spectrum és TVC gépekre írt programokból válogattunk. Érdekes, hogy nem érkezett Atarira írt pályamű.

A döntőt 1988. március 14–15-én tartottuk meg Szekszárdon a Garay János Gimnáziumban, ahová 8-8 oktató- és játékprogram jutott. A zsűri tartalmi és formai szempontokon kívül értékelt az is, milyen a bemutatott program ismertetésének előadásmódja, finomságainak, rejtett kincseinek kifejtése, illusztrálása.

Glória a győzőknek

Ki tudja, talán a zsűri tagjainak (1. kép) „öreg” szíve miatt, az oktatóprogramok egy kicsit mindig előnyt élveznek a játékprogramokkal szemben. Engedve ennek a tanárkodásnak, kezdjük ezért az oktatókategóriával. Márcsak a különosság kedvéért is, mert az első díjas program szerzője, Jurkovic Károly, a budapesti Berzsenyi Dániel Gimnázium tanulója alaposan feladta nekünk a leckét HT gépre írt, igazi szövegszerkesztő programjával. Egy ilyen jellegű program inkább már önálló „alkalmazói”, „felhasználói” kategóriába tartozna.



1. kép

Szerzője szerint a 64 k-s gépre írt szerzemény a teljes magyar ábécé felhasználásával írott szövegek írására, nyomtatására, mentésére, betöltésére, javítására, formátumozására, szerkesztésére, nyomtatási képe átnézésére képes. Jutalma az NJSZT által felkínált Primo számítógép és egy MK 29-es magnó volt. Alkotása, illetve az annak segítségével készített felhasználói dokumentációja egyben elnyerte az első ízben kiadott, a legjobb dokumentációért járó, a Novotrade Rt. által felajánlott 2000 Ft-os vásárlási utalványt is.

Második díjat nyert Benkő Dávid, a budapesti Móricz Zsigmond Gimnázium tanulója, Spectrum 48 k-s gépre írt „Oszi” jellegű pályamunkájával. Az oszcilloszkópot helyettesítő, működésének megértését segítő, bemutató program olyan programozástechnikai megoldásokat alkalmazott, melyek kivívhatják egy profi programozó elismerését is! Harmadik díjjal a szentesi ifj. Gulyás László jutalmaztuk. Programja, amely C Plus/4-re íródott, az általános iskolai magyar nyelvtanban a névmások tanítását, gyakorlását segíti. Ifj. Gulyás László elnyerte a Novotrade Rt. 2000 Ft-os vásárlási utalványt is.

Miként az oktatóprogramok küzdelme, úgy a játékoké is izgalmas versenyt hozott. Első díjat a rendező szekszárdi Garay János Gimnázium tanulója, Pirisa Levente kapott, C16-ra írt, de Plus/4-esen is futtatható Tűzoltó nevű (2. kép) programjéért. Jutalma a KISZ KB KSZT által felajánlott Primo A48-as számítógép volt.

A játék grafikája, gyorsasága, színei irányításának egyszerűsége irigylésre méltó, méltán aratott közönség sikert. Külön érdeme, hogy a szerző egy C16-os memóriájába mindezt becsúfolta, amiért különdíjként neki ítelték a Tolna megyei NJSZT 1000 Ft-os vásárlási utalványát és a rendező gimnázium KISZ-szervezete által felajánlott közönségdíj 1000 Ft-ját is.

lasi utalványát és a rendező gimnázium KISZ-szervezete által felajánlott közönségdíj 1000 Ft-ját is.

Az immár veteránnak számító Pallagi László a második és egyben a harmadik díjat is elnyerte. Második díjat Horror a temetőben című logikai játékáért kapott, amely a HT képességeit maximálisan kihasználja.

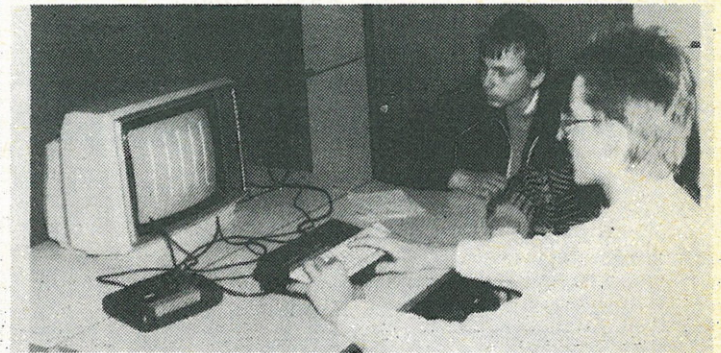
A harmadik díjat az Úrkaland nevű játékért kapta a szerző. Ennél a programnál is élt a HT nyújtotta összes lehetőséggel, messzerű, fantasztikus környezetbe ágyazva mondanivalóját.

Pallagi László második díjként a Tolna Megyei Tanács 2500, harmadik díjként a Novotrade Rt. 2000 Ft-os vásárlási utalványát érdemelte ki.

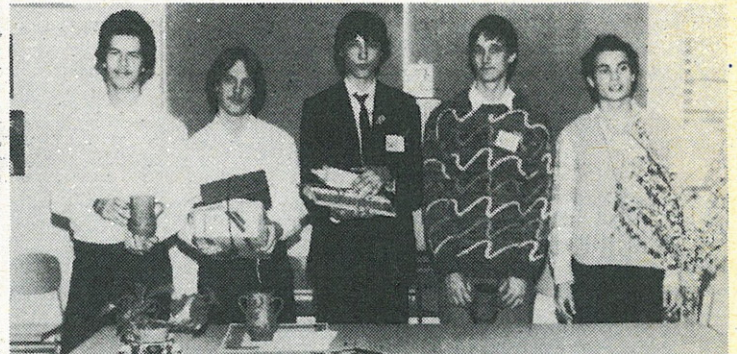
Üres kézzel nem távozott senki

A már említetteken kívül a Tolna megyei NJSZT különdíját, 1000 Ft-os vásárlási utalványt kapott Csókás Csaba, a tamási Béri Balogh Ádám Gimnázium tanulója C Plus/4-re írt Rezgések című programjéért.

Hasonló feladatot oldott meg Mészáros Tamás is Kiskunfélegyházáról. Programja azonban a legbonyolultabb függvények, a



2. kép



3. kép. A fődíj nyertesei (balról jobbra): Benkő Dávid, „Oszi”, messze a legjobb előadó, Jurkovic Károly, aki keveset beszél, de nyer, Pirisa Levente, az újdonsült játékrajnok, Pallagi László, a veterán, ifj. Gulyás László

magnó bemeneti jeleinek, a belső óra háromszög jelének, valamint a billentyűzetpuffer leütött és tárolt karaktereinek helyes, hanghatalmassal kísért megjelenítésén túl nem birkózott meg olyan egyszerű függvényekkel, mint a $2 \cdot X + 3$!

Négeli Péter, aki szintén a rendező gimnázium immár másodszor versenyző tanulója, a Tolna megyei NJSZT megyei versenyzőknek felajánlott 1000 Ft-os különdíját kapta, „Aktivitás” jellegű, Spectrumra írt kémiai oktatóprogramjéért. A rendező gimnázium KISZ-bizottságának 500 Ft-os különdíja ugyancsak az övé lett.

Nagy közönség sikert aratott, előadásával pedig a zsűrit is bámulatba ejtette Vidovics Ferenc Szentesről C Plus/4-re írt programjával. A gimnázium közönségdíjának 500 Ft-ját nyerte el. Az egyetlen fel nem emelt, majd helyet cserélt korszak helyének megalapítása még a zsűri legjobb szemű tagjainak sem sikerült mindig.

A díjak kiosztásakor, az elért eredmények kihirdetésekor azonban egyetlen versenyző sem távozott emléktárgy, oklevél, könyvjutalom nélkül.

A pályázat jövőre is folytatódik! A szerzőknek sok sikert kíván a szerkesztőség.

ÉNEKES FERENC

A Nemes Tihamér Országos Középiskolai Számítástechnikai Verseny döntője

A döntőre a versenybizottság meglepő feladatot tartogatott. A részt vevő ötven versenyző igencsak vakargatta a fejét, mikor átvette a feladatlapot, mert matematikalkalmazási kérdésre számított. A meglepetés elmúltával azonban mindenki munkához látott. Csak egy versenyző adta fel időközben. Nagyon szép megoldások születtek!

A diákok C64, HT és IBM PC/XT kompatibilis gépeken dolgoztak, de az előírások nem engedték meg a fejlettebb gépek beépített függvényeinek használatát. Így mindenki azonos eséllyel indult. A résztvevők egynegyede Pascal, a többi BASIC nyelven készítette el programját. A megoldásokat adathordozón kellett beadni. A versenybizottság csak a működő programokat vette figyelembe.

A feladat másságát az is bizonyította, hogy az első tíz helyezett az első forduló után még hátul kullogott. A verseny első helyezettje éppen csak bekerült a döntőbe. A tanuláson és gyakorlásán kívül sem volt azonban érdeklenni induli a versenyen, mert az első tíz helyezettnek nem kell matematikából felvételi vizsgát tennie, ha főiskolán vagy egyetemen akar továbbtanulni.

mot), majd értelmezni és végrehajtani azt! Egy programsorban az utasítást a paraméterétől és egyik utasítást a másiktól szóközzel kelljen elválasztani! A forrásprogram végét V utasítás jelzi (lásd lejjebb), ennek beolvasása indítja el automatikusan az értelmezést. A forrásprogram végrehajtása során keletkezett ábra annak lefutása után is maradjon látható mindaddig, amíg a felhasználó nem jelzi (egy tetszőleges billentyű lenyomásával), hogy az értelmező állapotba kerülhet, vagyis újabb grafikus program befogadására álljon készen! A forrásprogram elindulása előtt maga az értelmező gondoskodjon a képernyő letörléséről!

Az értelmezendő utasítások

M <hossz> — A ceruzát az aktuális helyről az aktuális irányba mozdítsa el <hossz> nyit! (1 egység legyen a gép legkisebb grafikus egysége!) Ha kell, akkor a mozgás nyoma látszódjon a képernyőn! (Lásd N, R parancsoknál.)

hely: = képernyő közepe, irány: = függőlegesen felfelé, és rajzoló üzemmódban legyen! (A hazamozgás során természetesen rajzolon, ha rajzoló üzemmódban van!)

V (Vége)

I <szám> (<utasítás-sor>) (Ismételd)

E <név> (<utasítás-sor>) (Eljárás)

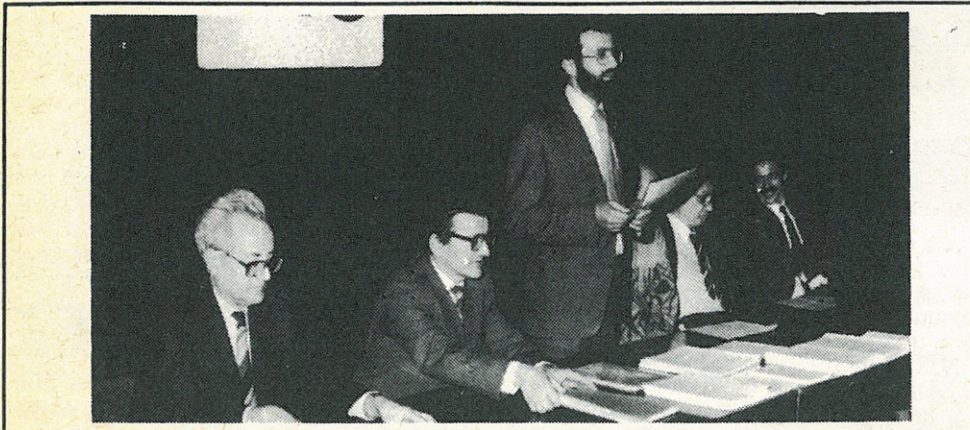
D <név> (Dobd el)

— Ez az utasítás jelezze a forrásprogram végét az értelmező számára! Ezután beírt program azonnal hajtódjon végre!

— Ez az utasítás tegye lehetővé, hogy az <utasítás-sor>-ban leírt teendők <szám>-szor végrehajtsódjanak!

— Ezzel az utasítással tudjunk az itt felsorolt utasításokból definiálni <név> névvel egy új, a többivel egyenrangú utasítást! (Ezket aktivizálni a <név> leírásával lehessen!)

— Ezzel lehessen a már feleslegessé vált, ÁLTALUNK definiált eljárásokat törölni, hogy ezáltal hely szabaduljon fel az újabbak számára!



Ünnepélyes eredményhirdetés

A versenybizottság jövő évre is várja az iskolák jelentkezését az NJSZT címére: Budapest V., Báthori u. 16.

A verseny eredménye

Első helyezett: Biczó Tibor, Zrínyi Gimnázium, Zalaegerszeg

Második helyezett: Gárdonyi Gergely, József Attila Gimnázium, Budapest

Harmadik helyezett: Nagy Bálint, Ságvári Endre Gimnázium, Budapest

Különdíjat kapott az első fordulóban való legjobb szerepléséért Drasny Gábor, Fazekas Gimnázium, Budapest.

A feladat

Készíts egy grafikus nyelvhez értelmező-programot BASIC vagy Pascal nyelven! A program legyen képes beolvasni az ezen a nyelven írt programszöveget (forrásprogra-

F <szög> (Fordulj)

N (Ne rajzolj)

R (Rajzolj)

T (Törölj)

H (Haza)

— Az aktuális helyen maradván az új haladási irány legyen a régi irány és <szög> összege! (A <szög> előjeles mennyiség, pozitív irány az óra járásának megfelelő irány.)

— Az utasítás kiadása után a következő R utasításig a ceruza mozgásának nyoma ne látszódjon!

— A következő N utasításig lehessen a képernyőn követni a ceruza mozgását!

— Hatására törlődjön a képernyő, de a ceruza aktuális helyzete (hely, irány és hogy rajzol-e vagy sem) ne változzon!

— Hatására a ceruza „kerüljön” az alapállapotba →

FONTOS!!!

— A kész program mellé be kell adni a program tervét (a fontosabb szubrutinok és a változók leírását!)

— ELŐSZÖR a vezérlő programrészt írd meg, majd ajánljuk, hogy az utasításokat a fenti sorrendben valósítsd meg!

— Bármilyen gépet és bármilyen nyelvet is használsz, a rajzolásához a programodban csak az egy pont rajzolásához és a képernyő törléséhez szükséges utasítást használhatod fel! (Ebből következik, hogy a szakaszrajzoló rutint is neked kell megírni!!) Ha az általad használt gépen vagy nyelvben nincs grafika, akkor a legkisebb egység nálad — természetesen — a betűméret lesz!

Bővítési lehetőségek

— Ha az eddigi feladatokat megoldotad, akkor próbáld meg úgy módosítani a programodat, hogy jelezze a felhasználónak a szintaktikus hibákat, majd a futás közbeni hibákat is!

— Ha ezt is túl kevésnek találod, akkor az eddigiek elkészítése után nekiláthatsz az utasítások neveinek bővítéséhez oly módon, hogy az egybetűs parancs alatt zárójelben levő szavakkal lehessen utasításokat beírni.

Mintaprogramok

1. M 20 F 60 M 20 F 60 M 20 V
2. T I 10 (R M 20 F 18 N M 20 F 18) V
3. E TETO (F 45 M 60 F 90 M 60) TETO V

TechnoMIR

Legújabb fejlesztésű modulok

A TechnoMIR rendszer előnye, hogy gyorsan és könnyedén képes alkalmazkodni a felhasználó igényeihez. De ez mégis leleményesebb amannál, mindig vannak új igényei! Így egyetlen rendszer sem állhat meg a fejlődésben, mert különben azonnal elavulna.

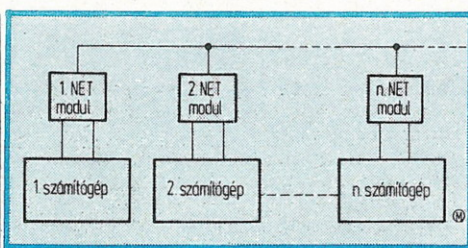
Az újabb fejlesztésű modulokra jellemző, hogy összetettebb feladatokat oldanak meg, mint a rendszer alapkészlete. Sokszor olyan igényeket elégítenek ki, amelyeket eredetileg sem vártunk a TechnoMIR-től. Tipikus példa erre a NET modul.

Optimista adatátvivő

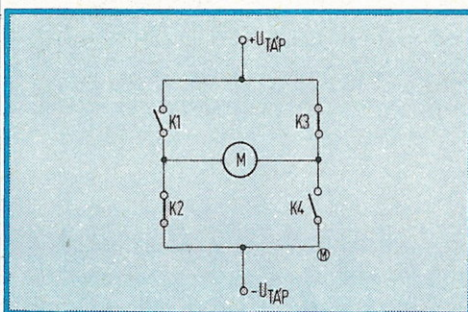
Lassacskán egyre több iskolában olyan szintű a számítástechnikai kultúra, hogy felvetődik a kérdés: miként lehetne a már meglévő számítógépek között adatátvitelt, sőt kommunikációt létrehozni? Az adatátvitelre láttunk már példát a DIN és DOUT modulokkal kapcsolatban. A NET modul azonban többet nyújt. A számítógépek között soros (tehát a vezetéken egymás után nyolc bit) adatátvitelt képes megvalósítani úgy, hogy szinte korlátlan mennyiségű számítógép fogadhatja a rendszer adatait.

A NET modulokat felhasználó, hálózattá szervezett számítógépek elvi elrendezési rajza az 1. ábrán látható. Az ábrából kitűnik, hogy a modul a TechnoMIR buszon keresztül párhuzamosan adatot fogad és ezt sorosan adja a kimenetén. Ez az adó üzem. Vételnél természetesen a dolog fordított; a vett soros adatot továbbítja az eszköz a buszon keresztül, már párhuzamosan. Ezt a bonyolult feladatot a modul „lelke”, egy Intel 8251 típusú IC (USART: Universal Synchronous/Asynchronous Receiver/Transmitter) valósítja meg.

Az áramkör programozása elég bonyolult, de maga a NET modul, mint a TechnoMIR rendszer minden tagja, BASIC-ből programozható. A modul programozása során nemcsak azt lehet beállítani, hogy adott esetben ADÓ vagy VEVŐ funkciót lásson el, hanem az adatátvitel gyorsaságának mértékét is.



1. ábra



2. ábra

A modulhoz olyan ún. tanár—diák program készült, amely korlátlan mennyiségű gépet is kezelhet, miközben az egyiknek, mondjuk a tanárnak, elsőbbséget nyújt. Tehát az oktatást közvetlenül is támogathatja, hiszen interaktív, párbeszédes rendszer szervezőt, amit nem vagy nem észrevehetően zavarnak a soros adatátvitelből adódó időproblémák.

Magyar viszonyok között az sem lebecsülendő, hogy akár egyetlen gyors háttértárral (például egy lemezmaghajtóval) és egyetlen nyomtatóval kiszolgálhatjuk a hálózatba szervezett összes számítógéppünket. A további felhasználáshoz pedig csak fantázia kell!

A TechnoMIR rendszer eddig már ismert moduljait is felhasználva nemcsak billentyűzeten beadott adatok továbbíthatók a hálózat többi

tagjának, hanem adott esetben a külső környezetet érzékelő és villamos jelekké, feldolgozható adatokká formáló DIN vagy A/D modulok jelei is. Tehát a rendszer moduljait felhasználva nehéz, nagyon bonyolult feladat is megoldható.

Hardwits, a motorvezérlő

Ez a modul is a felhasználók kényelmét és a rendszer alkalmazási területeinek további szélesítését szolgálja. Tulajdonképpen a szinkronmotorok vezérlésére fejlesztették ki, de — élve a működéséből adódó lehetőségekkel — más feladatok megoldására is felhasználható. A motor vezérlésén a forgásirány beállítása értendő. Ezt a feladatot a 2. ábrán látható elvi kapcsolási rajz alapján valósították meg.

A Hardwits nevű áramkör működéséről annyit érdemes tudni, hogy az egyes kapcsolókat, amelyeket a modulokban kapcsolótranszisztorokkal oldottak meg, úgy kell vezérelni, hogy egyszerre mindig kettő lehet zárva. Vagy az 1—4., vagy a 2—3. Így alakul ki a kívánt forgásiránynak megfelelő áramút. A modulban két áramkört helyeztek el, így egyszerre két motor vezérelhető. Tehát egy modul felhasználásával akár X—Y rajzolt is megvalósíthatunk.

Robot modulokból?

Egy másik, igen izgalmas elem a CPU modul, amelyet mostanság hoznak létre. A mikroszámítógépek központi egységének megfelelő modul a TechnoMIR rendszerhez kapcsolva részben nélkülözhetővé teszi a számítógépet. A CPU-val végrehajtható programot vagy egy EPROM-égető modulal beégetett EPROM-ban, vagy a modulhoz csatlakozó számítógép memóriájában tároljuk.

A POW, DOUT, DINP, CPU modulok összekapcsolásával akár robotot is építhetünk a megfelelő mechanikai kialakítás után.

Feltételezhető, hogy a rendszer fejlődése nem áll meg. Sőt, talán már most is újabb feladatok megoldására alkalmas modulokon gondolkodnak a fejlesztők. Mi csak érdekelteni akartuk a rendszer rugalmasságát és fejlődésének fő irányait.

Sorozatunk befejezésekként a következő számban konkrét, gyakorlati példákat mutatunk be a rendszer iránt érdeklődőknek.

ALBU LÁSZLÓ—KIRÁLY LÁSZLÓ

Zenei képességek vizsgálata számítógéppel

Most kissé nehezebb dolga lesz azoknak, akik az előző számban ismertett hallásérzékenységet mérő programot olvasták, s továbbra is szeretnének a hangok adta élmények világában kalandozni.

A nagyobb koncentráció miatt az értékelési szempontok adatait mindenki sorozza meg 1,5-del. A program három hangközt produkál: nagy tercet, kis tercet, nagy szeksztet. Nem kívánható minden olvasótól, hogy ezeket a hangközöket ismerje és felismerje. Ezért kezdődik a program úgy, hogy előbb meghallgatjuk a hangközt (80, 150), majd leénekeljük (190). Ez segít a későbbiekben, amikor hangolás lesz a feladatunk. A magasabb hang ugyanis kissé megváltozik. A változás irányát, nagyságát, a megközelítések nagyságát és számát a véletlen határozza meg. (270—330). Addig kell tehát nyomkodni a megadott billentyűket (hozzávetőlegesen 4—30-szor), míg szerintünk ismét tiszta nem lesz

a hangzás. Ekkor nyomjuk le a RETURN-t, s a többi már a program dolga. Megemlítem, hogy a hangközök nem a temperált, hanem az úgynevezett természetes hangrendszer frekvenciaviszonyai szerint szólnak meg. Ehhez igazodik az éneklés is. Elképzélhető, hogy például magasabb

osztályba járó zongoristák relatíve rosszabb eredményt érnek el, mint mások. Ez azért fordulhat elő, mert a temperálás előnyös tulajdonságain kívül a hallást elszűrkíti.

A program használatához sok sikert kívánok.
KALMÁR GYULA

```

10 PRINT"J":S=54272:FORI=0T024
20 POKES+I,0:NEXT
30 POKES+24,15:POKES+6,250
40 D=D+1:PRINTD",HANGKÖZ"
50 IFD=1THENO=4800:GOTO80
60 IFD=2THENO=6000:GOTO80
70 IFD=3THENO=4800
80 GOSUBS10:POKES,A:POKES+1,F
90 POKES+4,17:FORI=1T0800:NEXT
100 IFD=1THENO=6000:GOTO130
110 IFD=2THENO=7200:GOTO130
120 IFD=3THENO=8000
130 POKES,A:POKES+1,F:POKES+4,17
140 POKES+11,32:POKES+13,108:P=0
150 GOSUBS10:POKES+7,A:POKES+8,F
160 POKES+11,33:POKE198,0
170 PRINT"NYOMJ MEG EGY BILLENTYUT ES"
180 PRINT"MENEKELD EL A KET HANGOT!"
190 GETT$:IFT$="":THEN190
200 POKES+11,32:POKES+13,250
210 POKES+6,80:POKE198,0
220 PRINT"MOST USS LE EGY BILLENTYUT"
230 GETT$:IFT$="":THEN230
240 PRINT"NYOMJ MEG EGY BILLENTYUT"
250 PRINT"MENEKELD EL A KET HANGOT!"
260 PRINT"MENEKELD EL A KET HANGOT!"
270 PRINT"MENEKELD EL A KET HANGOT!"
280 PRINT"MENEKELD EL A KET HANGOT!"
290 PRINT"MENEKELD EL A KET HANGOT!"
300 PRINT"MENEKELD EL A KET HANGOT!"
310 PRINT"MENEKELD EL A KET HANGOT!"
320 PRINT"MENEKELD EL A KET HANGOT!"
330 PRINT"MENEKELD EL A KET HANGOT!"
340 PRINT"MENEKELD EL A KET HANGOT!"
350 PRINT"MENEKELD EL A KET HANGOT!"
360 PRINT"MENEKELD EL A KET HANGOT!"
370 PRINT"MENEKELD EL A KET HANGOT!"
380 PRINT"MENEKELD EL A KET HANGOT!"
390 PRINT"MENEKELD EL A KET HANGOT!"
400 PRINT"MENEKELD EL A KET HANGOT!"
410 PRINT"MENEKELD EL A KET HANGOT!"
420 PRINT"MENEKELD EL A KET HANGOT!"
430 PRINT"MENEKELD EL A KET HANGOT!"
440 PRINT"MENEKELD EL A KET HANGOT!"
450 PRINT"MENEKELD EL A KET HANGOT!"
460 PRINT"MENEKELD EL A KET HANGOT!"
470 PRINT"MENEKELD EL A KET HANGOT!"
480 PRINT"MENEKELD EL A KET HANGOT!"
490 PRINT"MENEKELD EL A KET HANGOT!"
500 PRINT"MENEKELD EL A KET HANGOT!"
510 PRINT"MENEKELD EL A KET HANGOT!"

```

Ékezetes névsorok rendezése

Az ékezetes betűket tartalmazó névsorok rendezése nehezebb feladat, mint amilyenek először látszik. A megoldás kulcsa ugyanis nem az, hogy rendelkezik-e a gépünk ékezetes karakterkészlettel; ha nem, ez a legtöbb gépen pótolható. A probléma egészen más.

A szövegek, sztringek rendezése azon az elven történik, hogy az összehasonlított karakterek kódja kisebb vagy nagyobb-e. A kisebb kódú karakter előrébb áll az ábécében. Ha ékezetes betűkkel is dolgozunk — akár voltak a gépünkön, akár magunk generáltuk őket —, ezeknek a kódja nem illeszkedik bele az eredeti ASCII kódrendszerbe, így rendezéskor sem a kívánt eredményt kapjuk. Tehát a szokásos rendezések nem alkalmazhatók a karakter eredeti kódja szerint; át kell kódolni a sztringeket igényeink szerint.

A következőkben megadom a lehetséges megoldást minden, az iskolákban előforduló géptípusra. Ez eléggé egyedi vonása az írásnak — azt hiszem, érdemes lenne azt az utat követni: egy-egy probléma megoldását egy írás keretében több géptípusra kidolgozni. Az oktatásban ez kézenfekvő, más területeken is csak hasznos lehet.

A megoldás alapja tehát egy bizonyos átkódolás. Ez a karakterek kódjának olyan átrendezését jelenti, amelynek során az ékezetes jelek kódjai az alap karakterek közé sorakoznak be, ábécébeli helyüknek megfelelően. Eközben figyelembe kell venni a névsorok rendezésének helyesírási szabályozását is (A magyar helyesírás szabályai, 14—16.). Már is jelzem, hogy megoldásom nem terjed ki minden finomságra: például a *cs* meg fogja előzni a *cz*-t, a *ty* a *tz*-t. A teljes körű megoldás olyan sok vizsgálatot jár, hogy lehetetlenül lassú lenne a rendezés.

A gyakorlati megoldás a következő. A rendezetlen fájl egy F\$ nevű tömbben helyezkedik el. Ennek tagjait átkódolva elhelyezzük egy R\$ tömbben. A rendezés bármely szokásos rendezőeljárással az R\$ tömb szempontjai szerint történik, de a rendezés minden cseréjét elvégezzük tükörképként az F\$ tömb megfelelő elemeire is. Így ott is rendezett tömböt kapunk.

Az átkódolás úgy történik, hogy megállapítjuk a sztring következő karakterének ASCII kódját. A kódot azonnal át is helyezi a gép: a kisbetűt azonosítja a nagygal. Aztán következik a leglényegesebb mozzanat: a kódnak megfelelő átkódolótömbben megkapja a program, hogy milyen jelekből építse fel az R\$ tömböt. Ebben számok, írásjelek is előfordulnak, de ez a gép belső ügye!

A különböző géptípusok megoldásai között három lényeges eltérés adódik. Az egyik: eltérőek a sztringműveletek (a jelsozások elemzése, felépítése, kódolása, válogása). Ilyen szempontból az egyik típusba tartozik az összes Commodore, a HT és a Primo — a másik típusba a Spectrum és a Videoton. A másik eltérés az adatfájl beolvasása és kiírása terén mutatkozik, ez gépenként eltér. A harmadik különbség az

```
10 REM *****
20 REM * ÉKEZETES RENDEZŐ (C-PLUS 4) *
30 REM *****
100 REM *** ELŐKÉSZÍTÉS, ADATBEOLVASÁS
110 N=...: DIM F$(N),R$(N),K(63),L(20),M(20)
120 FOR I=0 TO 63: READ K(I): NEXT
200 REM ** A RENDEZETLEN FILE BEOLVASÁSA
210 OPEN B,B,B,"FILENÉV,S,R"
220 FOR I=1 TO N: INPUT# B,F$(I): NEXT
230 CLOSE B
300 REM *** A STRINGEK ATKÓDOLÁSA ***
310 FOR I=1 TO N: R$(I)="
320 FOR J=1 TO LEN(F$(I))
330 K=ASC(MID$(F$(I),J,1))
340 IF K=45 THEN 370
350 K=K-32*(K=32)+64*(K>160 AND K<190)+128*(K>192)
360 R$(I)=R$(I)+CHR$(K(K-64))
370 NEXT: NEXT
400 REM *** RENDEZÉS (GYORSRENDEZŐ) ***
410 S=1: L=1: M=N
420 I=L: J=M: V$=R$(L+M/2)
430 IF R$(I)<V$ THEN I=I+1: GOTO 430
440 IF R$(J)>V$ THEN J=J-1: GOTO 440
450 IF I>J THEN 480
460 IF I<J THEN C$=R$(I): R$(I)=R$(J): R$(J)=C$: C$=F$(I): F$(I)=F$(J): F$(J)=C$: F$(I)=F$(I)
470 I=I+1: J=J-1: IF I<J THEN 430
480 IF I<M THEN L(S)=I: M(S)=M: S=S+1
490 M=J: IF L<M THEN 420
500 IF S>1 THEN S=S-1: L=L(S): M=M(S): GOTO 420
600 REM *** KIÍRÁS VAGY LEMEZRE VÉTEL **
610 OPEN B,B,B,"@FILENÉV,S,W"
620 FOR I=1 TO N: PRINT# B,F$(I): NEXT
630 CLOSE B
700 REM *** ADATOK AZ ATKÓDOLÁSHOZ ***
710 DATA 32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47
720 DATA 49,50,51,52,53,54,56,57,58,59,60,32,32,32,32,32
730 DATA 32,41,41,48,32,48,32,32,32,32,32,33,37,32,55,32
740 DATA 33,37,55,55,48,54,47,48,54,47,32,55,32,32,32
```

1. lista

```
10 REM *****
20 REM * AZ ALAFILE MEGÍRÁSA (PLUS 4) *
30 REM *****
40 N=...: DIM F$(N): OPEN B,B,B,"FILENÉV,S,W": FOR I=1 TO N: PRINT I;
50 INPUT ". NEV:"; F$(I)
60 PRINT# B,F$(I): NEXT: CLOSE B
```

2. lista

```
10 REM *****
20 REM * VÁLTOZAT PRIMO A-RA *
30 REM *****
110 N=...: CLEAR N*20: N=...: DIM K(39)
210 OPEN "FILENÉV"
220 FOR I=1 TO N: INPUT# F$(I): NEXT
230 CLOSE
350 K=K+26*(K>122 AND K<128)+30*(K=32)+32*(K>96 AND K<123)
360 R$(I)=R$(I)+CHR$(K(K-62))
710 DATA 32,41,37,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,47,48,33
720 DATA 5,54,37,48,48,33,55,55
730 DATA 5,54,37,48,48,33,55,55
```

3. lista

egyéni kódrendszerből adódik: gépenként más-más átkódolást igényel.

A közölt listák minden további útmutatást megadnak. Alkalmazásuk előtt arra érdemes még kitérni, hogyan készítsünk ékezetes ábécét azokra a gépekre, amelyeken eredetileg nincs. Ami a Commodore Plus/4-et illeti, az iskolák ékezetes karakterkész-

```
10 REM *****
20 REM * VÁLTOZAT PRIMO B-RE *
30 REM *****
110 N=...: CLEAR N*20: N=...: DIM K(43)
350 K=K+26*(K=32 OR K>122 AND K<128)+32*(K>96 AND K<123)
360 R$(I)=R$(I)+CHR$(K(K-58))
710 DATA 32,47,48,54,41,41,37,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,49,50,51,52,53,54,56,57,58,59
730 DATA 60,47,48,33,55,54,37,48,48,33,55,55
```

4. lista

```
10 REM *****
20 REM * VÁLTOZAT HT-1080Z-RE *
30 REM *****
110 N=...: CLEAR N*20: N=...: DIM K(34)
210 FOR I=1 TO N: INPUT# 1,F$(I): NEXT
340 IF K=45 OR K=96 THEN 370
350 K=K-10*(K>50 AND K<55)-22*(K>34 AND K<39)-24*(K=32)+32*(K>96 AND K<123)
360 R$(I)=R$(I)+CHR$(K(K-56))
710 DATA 32,48,48,55,55,48,48,55,55,33,34,35,36,37,38,39
720 DATA 40,41,42,43,44,45,46,47,49,50,51,52,53,54,56,57
730 DATA 58,59,60
```

5. lista

6. lista

```
10 REM *****
20 REM * ÉK.REND. (SPECTRUM) *
30 REM *****
100 REM *** ELŐKÉSZÍTÉS, ADAT *
110 LET n=...: DIM F$(n,20): DIM R$(n,20): DIM k(48): FOR i=1 TO 48: READ k(i): NEXT i
200 REM *** RENDEZETLEN FILE **
210 LOAD "FILENÉV" DATA f$(i)
300 REM *** ATKÓDOLÁS ***
310 FOR i=1 TO n: LET e=1
320 FOR j=1 TO 20
330 LET k=CODE a$(i,j)
340 IF k=45 THEN GO TO 370
350 LET k=k+32*(k>64 AND k<91)+64*(K=32)-21*(k>143 AND k<165)
360 LET r$(i,e)=CHR$(k(k-95)): L
ET e=e+1
370 NEXT j: NEXT i
400 REM *** RENDEZŐ (SHELL) ***
410 LET m=n
420 LET m=INT(m/2): IF m=0 THEN GO TO 610
430 LET j=1: LET k=n-m
440 LET i=j
450 LET l=i+m: IF r$(i)>r$(l) THEN LET c$=r$(i): LET r$(i)=r$(l): LET r$(l)=c$: LET c$=f$(i): LET f$(i)=f$(l): LET f$(l)=c$: LET T i=i+m: IF i>0 THEN GO TO 450
460 LET j=j+1: IF j>k THEN GO TO 0 420
470 GO TO 440
600 REM *** ÚJ FILE KAZETTÁRA *
610 SAVE "FILENÉV" DATA f$(i)
700 REM *** ADATOK ***
710 DATA 32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,49,50,51,52,53,54,56,57,58,59,60,33
720 DATA 32,32,55,37,55,32,54,41,41,48,48,48,48,47,47,33,55,37,55,54
```


Feladatok – megoldások



```
10 REM *****
20 REM * ALAPFILE (SPECTRUM) *
30 REM *****
40 DIM f$(n,20); FOR i=1 TO n
50 INPUT f$(i); NEXT i
60 SAVE "FILENÉV" DATA f$()
```

7. lista

```
10 REM *****
20 REM * ÉKEZETES RENDEZŐ (VIDEOTON) *
30 REM *****
100 REM *** ELŐKÉSZÍTÉS, ADATBEOLVASÁS *
110 N=...: DIM F$(N)*20,R$(N)*20,K(40)
120 FOR I=0 TO 40: RED K(I): NEXT
200 REM *** RENDEZETLEN FILE BEOLV. ***
210 OPEN INPUT "FILENÉV"
220 FOR I=1 TO N: INPUT#5:F$(I): NEXT
230 CLOSE INPUT
300 REM *** A STRINGEK ATKÓDOLÁSA ***
310 FOR I=1 TO N: E=1
320 FOR J=1 TO LEN(F$(I))
330 P=ORD(F$(I)(J))
340 IF P=45 THEN 370
350 P=P+16*(P>143 AND P<153)-32*(P>64 AND P<91)-64*(P=32)
360 R$(I)(E)=CHR$(K(P-96))
370 NEXT: NEXT
400 REM *** RENDEZÉS (MEG KELL OLDANI!) *
410 REM *** A VIDEOTON HOSSZÚSÁG SZERINT HASONLÍTTJA ÖSSZE A STRINGEKET (<>=>!) ***
420 REM *** PÉLDÁUL AZ ELSŐ BETŰ SZERINT:
430 J=0: FOR I=2 TO N: IF ORD(R$(I-1))>ORD(R$(I)) THEN C$=R$(I-1): R$(I-1)=R$(I): R$(I)=C$: J=1
440 NEXT: IF J=1 THEN 430
600 REM *** KIÍRÁS VAGY KAZETTÁRA VÉTEL
610 OPEN OUTPUT "FILENÉV"
620 OR I=1 TO N: PRINT#5:F$(I): NEXT
630 CLOSE OUTPUT
700 REM *** ADATOK AZ ATKÓDOLÁSHOZ ***
710 DATA 32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47
720 DATA 49,50,51,52,53,54,56,57,58,59,60,32,32,32,32,32
730 DATA 33,37,41,47,48,48,54,55,55
```

8. lista

```
10 REM *****
20 REM * ALAPFILE RÖGZÍTÉSE (VIDEOTON) *
30 REM *****
40 N=...: DIM F$(N)*20
50 OPEN OUTPUT "FILENÉV": FOR I=1 TO N
60 INPUT F$(I): PRINT#5:F$(I): NEXT
70 CLOSE OUTPUT
```

9. lista

lettel felszerelt gépeket kaptak. Ezen a típuson praktikus az ékezetes betűk billentyűosztása: a QWER soron vannak a nagybetűk, az ASDF soron a kicsik. Ezt javaslom a Commodore 64-en is, ha például a Mikromagazin 1986/6-os számában közölt Földi—Énekes cikk szerint készítenek ékezetes ábécét (a cikk megoldása nyomtatásra is alkalmas!). Ha C16-os vagy ékezetes készlet nélküli Plus/4-es gépünk van, segíthet a karakterkészítésben a következő írás: BIT-LET 44. (1987. május, Eöry). Ha valakinek Plus/4-ese és C64-ese is van, a Plus/4 ékezetes adatait közvetlenül is a C64-be juttathatja! A Spectrum ékezetes ábécéje az Ötlet 1986/49. száma alapján készülhet.

A közölt listák némelyike teljes (Plus/4, Spectrum, Videoton). A többi ezeknek változata, csak az eltérő sorokat közöltem.

FEKETE GYÖRGY

Sorozatunkat elsősorban középiskolásoknak szánjuk, de reméljük, hogy minden olvasónknak tanulási lehetőséget és szórakozást nyújt.

A feladatok a Nemes Tihamér országos számítástechnikai verseny színvonalának felelnek meg. Minden esetben olyat választunk, amely röviden, gyorsan megoldható, de megoldásához ötletre van szükség. A megoldást mindig a következő számban közöljük.

Mivel változatosságra törekszünk, különböző programozási nyelveket használunk. Az is előfordul majd, hogy egy feladatra több programnyelven is közlünk megoldást, ezzel is elősegítve az ismeretszerzést.

A szerkesztőség várja az olvasók, a versenyzők leveleit. A legötletesebb program beküldőjét könyvtalánnyal jutalmazzuk. Ne feledjenek azonban a programhoz leírást is mellékelni!

2. rész

1. feladat: Önkiíró

Írjon olyan programot, amely kiírja önmagát!

Megoldás:

Minthogy a feladatkiírás megtiltott minden külső adatmezőhöz való hozzáférést, a programnak önmagát stringkonstans alakban kell tartalmaznia. A program azonban tartalmazza a stringet is. Nyilván ez nem lehet benne a stringben. Így tehát a stringbe a program teljes szövegét be kell írunk, kivéve magát a stringet. A program feladata pedig az lesz, hogy kiírja a stringet (ez maga a program) és idézőjelek közt újra a stringet (ez lesz a programszövegben elhelyezett stringkonstans).

Ekkor azonban újabb akadály előtt állunk: dolgozunk a stringkonstansban levő idézőjelek megkettőzése. A két probléma megoldásának programját az 1. ábra mutatja meg. Itt a 10-es sorban látható PRINT A\$ írja ki a program 5–50-es sorát és a 60-as sor elejét. Ezután a 20–50 sorok kiírják a stringkonstans, különös tekintettel az idézőjelekre (30-as sor).

Ez a program ZX-Spectrum BASIC-ben íródott, de annak csak két specialitását használja ki:

– A fizikai sorhossz 32 karakter.
– A logikai sorhossz nincs korlátozva.
Ezek figyelembevételével bármely gépre átírható.

A 2. ábrán látható PASCAL program példázza, hogy a feladatot meg lehet oldani géptől teljesen függetlenül is. Ennél a stringkonstansok a program közepén vannak, ezért a programszöveget két részletben írjuk ki. Előbb az 1–7. sorokat, majd a stringet, végül az A [8]–A [23]-ból a program végét.

```
10 REM ONMAGAT NIRO PROGRAM
10 READ H$ PRINT A$,
20 FOR I=1 TO LEN A$
30 IF A$(I)="" THEN PRINT ""
40 PRINT A$(I);
50 NEXT I PRINT ""
60 DATA " 5 REM ONMAGAT KIIR
6 PROGRAM " 10 READ A$ PRINT A
$ " 30 FOR I=1 TO LEN A
$ " 30 IF A$(I)=""
7 THEN PRINT ""
10 PRINT A$
50 NEXT I:
60 DATA
```

1. ábra

```
10 REM ONMAGAT NIRO PROGRAM
10 READ H$,
20 FOR I=1 TO LEN H$
30 IF H$(I)="" THEN PRINT ""
40 PRINT H$(I);
50 NEXT I PRINT ""
60 DATA " 5 REM ONMAGAT KIIR
6 PROGRAM " 10 READ A$ PRINT A
$ " 30 FOR I=1 TO LEN A
$ " 30 IF A$(I)=""
7 THEN PRINT ""
10 PRINT A$
50 NEXT I:
60 DATA
```

2. ábra

2. feladat: Snóbli

A snóbli különleges változatát a következőképpen játsszák. Két játékos (A és B) kezével letakar egy pénzdarabot. A játék előtt megegyeznek abban, hogy amikor felfedik a pénzeket, azonos oldaluk látszik, akkor A, ha különböző, B nyer. Belátható, hogy matematikailag ilyen feltételekkel és ha bármelyikük 0,5 valószínűséggel fejet vagy írást rejt, akkor hosszú ideig tartó játék során egyik játékosnak sem lehet nyerő stratégiája, azaz nem nyerhet.

Igen ám, de az ember nem képes így játszani, mert a játék alakulása befolyásolja döntését, azaz mindenképp kialakít valamilyen játékmódot.

Írjon olyan számítógépprogramot, amely kihasználja az ellenfele játékában felismerhető szabályszerűséget, és ez alapján hosszú távon biztosan nyer! Törekedjen arra, hogy a program minél többféle sorozatot felismerjen!

Írja meg azt is, hogy hányfélet!
A legjobb program beküldőjét könyvtalammal díjazzuk. PINTÉR GÁBOR

Zenei segédlet

COMMODORE 64

E kis rutin a hanggenerátorok programozását könnyíti meg, ugyanis elkerüli a sok POKE utasítást. A gépi kódú rutin mind programból, mind parancs üzemmódból meghívható a következőképpen:

SYS 828, a hanggenerátor száma, hang-

erő, impulzusszélesség felső, alsó bájtt, AD, SR, hang felső, alsó bájtt, kontrollregiszter.

Egy jó tanács! A zene vagy hangeffektus végén ne felejtsük el a hanggenerátorokat kikapcsolni, mert működtetésük kellemetlen zajt okoz. A kikapcsolás módja:

SYS 828, a hanggenerátor száma, 0,0,0,0,0,0,0

A könnyebb kezelhetőség kedvéért közüljük a BASIC programot, valamint az Assembler változatot is.

ROSNER GÁBOR

1. lista. BASIC változat

```

10 #=828
15 SZAM=#B7F1
20 JSR SZAM
25 CPX #1
26 BEQ FL
30 CPX #2
31 BEQ MA
35 CPX #3
36 BEQ HA
120 EL JSR SZAM
130 STX $D418
140 JSR SZAM
150 STX $D403
160 JSR SZAM
170 STX $D402
180 JSR SZAM
190 STX $D405
200 JSR SZAM
210 STX $D406
220 JSR SZAM
230 STX $D401
240 JSR SZAM
250 STX $D400
260 JSR SZAM
270 STX $D404
280 RTS
290 MA JSR SZAM
300 STX $D41A
310 JSR SZAM
320 STX $D40A
330 JSR SZAM
340 STX $D409
350 JSR SZAM
360 STX $D40C
370 JSR SZAM
380 STX $D40D
390 JSR SZAM
400 STX $D408
410 JSR SZAM
420 STX $D407
430 JSR SZAM
440 STX $D40R
450 RTS
460 HA JSR SZAM
470 STX $D41R
480 JSR SZAM
490 STX $D411
500 JSR SZAM
510 STX $D41W
520 JSR SZAM
530 STX $D413
540 JSR SZAM
550 STX $D414
560 JSR SZAM
570 STX $D40F
580 JSR SZAM
590 STX $D40F
600 JSR SZAM
610 STX $D412
620 RTS
630 .END
    
```

2. lista. Assembler változat

```

1 REM ***** ZENE-LOADER *****
2 :
5 X=0
10 FORI=828TO989
20 READA:POKEI,A
30 X=X+A
40 NEXTI
50 IFX<>220/3THENPRINT"HIRAS ADATOK!"
60 END
100 DATA 32,241,183,224,1,24W,8,224
110 DATA 2,240,53,224,3,240,98,32
120 DATA 241,183,142,24,212,32,241,183
130 DATA 142,3,212,32,241,183,142,2
140 DATA 212,32,241,183,142,5,212,32
150 DATA 241,183,142,6,212,32,241,183
160 DATA 142,1,212,32,241,183,142,0
170 DATA 212,32,241,183,142,4,212,96
180 DATA 32,241,183,142,24,212,32,241
190 DATA 183,142,10,212,32,241,183,142
200 DATA 9,212,32,241,183,142,12,212
210 DATA 32,241,183,142,13,212,32,241
220 DATA 183,142,8,212,32,241,183,142
230 DATA 7,212,32,241,183,142,11,212
240 DATA 96,32,241,183,142,24,212,32
250 DATA 241,183,142,17,212,32,241,183
260 DATA 142,16,212,32,241,183,142,19
270 DATA 212,32,241,183,142,20,212,32
280 DATA 241,183,142,15,212,32,241,183
290 DATA 142,14,212,32,241,183,142,18
300 DATA 212,96
    
```

TOP-lista

Listánkat felhasználói, illetve játékprogramokból állítjuk össze. A legjobbakat, legérdekesebbeket a beküldött javaslatok alapján rangsoroljuk. Ehhez kérjük az olvasók közreműködését. C64-re, ZX-Spectrumra, Enterprise-ra, TVC-re, Atarira és IBM-re készült programokat várunk havonta.

Címünk: Mikroszámítógép Magazin Szerkesztősége, 1371 Budapest, Pf. 433.

Diákszerkesztőség

Felhasználói		Játék	
1. GEOS 1.3	C64	1. Great Escape	C64
2. Printfox	C64 IBM	2. Nebulus	C64
3. Art studio 1.2c	C64 Spect.	3. Időregész	C64
4. Oxford pascal	C64 IBM	4. Microvocals	C64
5. Hires Master	C64	5. Test Drive II.	C64
6. Giga Load	C64	6. The Last Ninja	C64
7. Forth/a	C64 Spect.	7. Music Maker II.	C64
8. Sprite Machine	C64	8. Batty	Spectrum
9. Hisoft Pascal	Spect. Primo	9. Defender OTC.	C64
10. Game Maker	C64	10. Alt. World Games	C64

Primo lapok

A Primo A64 számítógépet a Z80 által címezhető memória legfelső határáig kiépítették ROM és RAM memóriákkal. Sokszor azonban szükség lenne a ROM által elfoglalt memóriaterületre. Kisebb memóriakiépítésű Primók bővítése is ilyen gondokat vet fel, mert a beégetett ROM program nem alkalmazkodik a megnövelt memóriához (lásd a Magazin 1986/10. számának 33. oldalán).

A megoldás mindkét esetben az, hogy a beépített ROM-ot eltüntetjük, és új memóriaterületeket nyitunk meg. Az ábrán látható áramkör alkalmas 16 kb-át RAM és/vagy további 2 x 16 kb-át EPROM belapozására a 0-3FFF címtartományba. Ily módon a rendszerprogramot lecserélhetjük, történetesen módosított BASIC interpreterre, assemblerre vagy valami másra. Készen van már a Spectrum ROM programjának átírása Primóhoz, így Spectrumnak BASIC és néhány gépi kódú programja Primón is futtatható.

A lapozás egy kimeneti porton kialakított lapozóregiszter feladata. Az új lapszám OUT utasítással állítható be. Az aktuális lapszámot az IC1 négy bites D tároló tárolja, amelynek beíró feltétele a W3 és MA1 vonalak aktív alacsony helyzete. Ez a tároló bekapcsoláskor 0-ra áll be, így alaphelyzetben az IC3 az eredeti ROM-ot választja ki. Ha a tárolóba új értéket írunk, ez az IC Y0-Y3 lábain a kívánt lap engedélyező jelét adja.

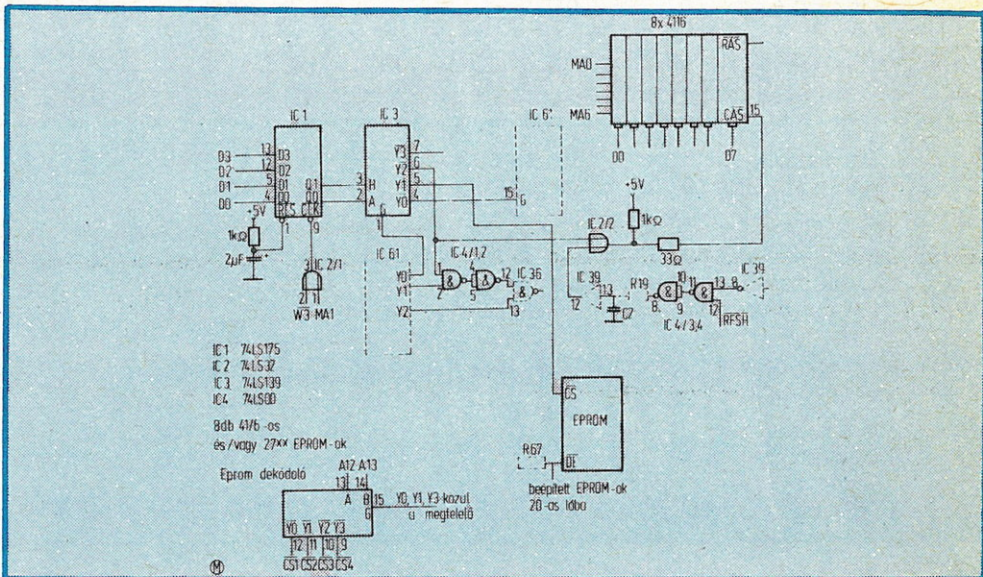
A 2-es lapon van a RAM. Az IC 4/1 és 4/2 lehetővé teszi, hogy a multiplexer akkor is aktivizálódjon, ha a plusz RAM-okhoz fordulunk. IC 2/2 adja a CAS jelet. IC 4/3 és IC 4/4 feladata, hogy a Primo egy kisebb géphibáját kijavítsa. A memória frissítésekor ugyanis a memória írási parancsot kap. A megjelenő (hamis) írási cím az I regiszterből (felső 8 bit) és az R regiszterből áll össze. A Primónál RESET után az I regiszter 0-t tartalmaz, és ezt nem is változtatja meg semmi. Ha azonban az I regisztert a saját RAM-ba állítjuk, mi is látni fogjuk, mi történik a 0-80 (hex.) memóriatartománnyal.

Futtatassunk le egy programot, amely A64-es gépen az I-be 232-t tölt. A képernyő felső részén a hamis „frissítési” írás a memóriatartalom elvesztését idézi elő. Az áramkör frissítéskor nem ad CAS jelet, így a problémát megoldja. Az EPROM-ok beépítése a rendszerbe sokkal egyszerűbb. Ha nincs szándékunkban a RAM-ot megépíteni, az IC 4/1, IC 4/2, IC 2/2 elmaradhat. Az EPROM-hoz hozzá kell kötni a D0-D7 adatvonalakat, az EPROM típusának megfelelő számú címvonalat és az eredeti EPROM-ok OE vonalát. Az EPROM CS vezetékekhez az Y0-Y3 vezetékek közül a lapszámot meghatározót köttjük. A rendszer bejelentkezéskor abban az EPROM-

```

10 POKE 16443,peek(16443)
    and127:out63,peek(16443)
20 ; 0D ERROR után : goto 100
30 i=50000
40 read a:poke i-65536,a:i=i+1:goto 40
100 for i=0 to 7
110 ?call(50000-65536)
120 next
130 input a §
140 goto 100
1000 data 62,2,211,253,33,0,0,62,1
1010 data 174,119,230,1,111,38,0,175
1020 data 211,253,201
    
```

1. lista



```

10 POKE 16443,peek(16443)and127:out63,
    peek(16443)
20 ; 0D ERROR után : goto 100
30 i=50000
40 read a:poke i-65536,a:i=i+1:goto 40
100 i=call(50000-65536)
1000 data 33,0,0,17,0,128,1,0,64,237,176
1010 data 62,2,211,253,33,0,128,17,0,0,1,0,64
1020 data 237,176,33,0,0,17,0,128,1,0,64
1030 data 237,176,175,211,253,201
    
```

2. lista

ban indul el, amelynek CS vonalát az Y0 vezetékekhez köttük. Egy 27128-as EPROM-ba a teljes BASIC-rendszert átégethetjük. Ha a tárat kénytelen-kelletlen kisebb kapacitású memóriákból állítjuk össze, az IC3 másik felét címdekódolásra használhatjuk.

Az áramkör megépítéséhez és felélesztéséhez bizonyos gyakorlat és felkészültség szükséges. Ennek hiányában gépünket nagyon könnyen tönkretelhetjük. A panelt egyszerű „lebegő” kivitelben az alaplemez fölé szerelni. Ügyelni kell arra (különösen az

adat- és címvezetékeknél), hogy minél rövidebb vezetékeket használjunk az összeköttetésre.

„A” sorozatú gépeknél előfordulhat, hogy a billentyűzet alatt elvezetett vezetékek miatt valamely gomb „beindul”. A csatlakozásokat legegyszerűbben az eredeti IC-k lábain oldhatjuk meg.

A 4116-os RAM IC-k -5 és +12V-os tápfeszültség-ellátása kritikus, ezért legegyszerűbb a benn lévő RAM tetejére forrasztani úgy, hogy a CAS lábat felhajlítjuk és panelunkhoz köttjük. Ilyen „anya” IC csak operációs RAM lehet. Előfordulhat — különösen 3,75 MHz-en futó processzoroknál —, hogy az új RAM-ok nem elég gyorsak.

Ezen segíthet, ha a C2-es kondenzátort 220 pF-ről 150 pF-re cseréljük.

Az EPROM-ok sebességével is lehetnek gondjaink. Ezen úgy segíthetünk, hogy az EPROM OE lábát a földre köttjük, CS lábára az eddigi CS és az OE VAGY kapcsolatba hozott jelét vezetjük. A kapcsolási rajzon szaggatott vonallal jelzett alkatrészek eredetiek és a jelölésük a Primo hardverleírásához igazodik.

Ha gondosan dolgoztunk, az áramkör bekapcsolás után azonnal működőképes. A lapozóregiszter a 253-as (hex. FD) címen érthető el. A kívánt lapot egy OUT 253,n utasítással állíthatjuk be.

Az áramkör vizsgálatát a mellékelt programokkal végezhetjük el. Az 1. lista programja egy cím tartalmának megadott bitjét billegtetni, a képernyőn felváltva 0-k és 1-esek kell hogy megjelenjenek. Az aláhúzott számokat kicserélve 2, 4, 8, 16, 128-ra, más biteket is tesztelhetünk. A 2. listán látható program a BASIC interpreter írja át RAM-ba, majd kiolvassa a 32768 címre, ahol ellenőrizhető. Ha egyezik, OUT 253,2 utasítással áttérhetünk az átirrt programra.

PALLER GÁBOR

Spectrum INPUT rutin TVC-re

A ZX81 és a ZX-Spectrum BASIC-jének egyedülálló tulajdonsága, hogy az INPUT-ra adott válasz tetszőlegesen bonyolult kifejezés is lehet. A válasz beadása után az interpreter kiértékeli a kifejezést, és ezután folytatódik a program futása. Ez a módszer nagyon jó többek között függvénykiértékelő és -ábrázoló program készítéséhez. Az

műveleti jelek. A program az INPUT-ba beadott kifejezéseket átalakítja úgy, hogy a kisbetűket nagybetűkre (17-es sor), a műveleti jeleket megfelelő tokenjükre cseréli (5–16. sor), majd elhelyezi egy BASIC sorban (ez a 2. sor). Amennyiben a beadott kifejezés szintaktikailag hibás, hibajelzést kapunk (** Not understood).

Az eredmény az A változóba kerül.
 – Ha sztring kifejezést akarunk bekérni: VT=1: GOSUB 3.
 A kifejezésben az idézőjel helyett aposztrofot kell használni. Az eredmény az AS változóba kerül.
 – GOSUB2 utasítással a kifejezés bármikor újra kiértékelhető.

```

1 DIMS$*254, A$*254: GOTO20
2 A=.....
.....
.....
3 INPUTPROMPT"KIFEJEZES:"; S$: X=6667: IFS$="" THENPRINT"HIBAS! !": GOTO3
4 IFVT=1 THENPOKE6665, ORD("$"): ELSEPOKE6665, ORD(" ")
5 FORI=TOLEN(S$)
6 IFS$(I)="(" THENS$(I)=CHR$(150)
7 IFS$(I)=")" THENS$(I)=CHR$(149)
8 IFS$(I)="=" THENS$(I)=CHR$(154)
9 IFS$(I)="+" THENS$(I)=CHR$(152)
10 IFS$(I)="-" THENS$(I)=CHR$(162)
11 IFS$(I)="*" THENS$(I)=CHR$(168)
12 IFS$(I)="/" THENS$(I)=CHR$(161)
13 IFS$(I)="↑" THENS$(I)=CHR$(159)
14 IFS$(I)="&" THENS$(I)=CHR$(151)
15 IFS$(I)=":" THENS$(I)=CHR$(253)
16 IFS$(I)="'" THENS$(I)=CHR$(34)
17 IFS$(I)>"Z" ANDS$(I)<="z" THENS$(I)=CHR$(ORD(S$(I))-32)
18 NEXTI: FORI=1 TOLEN(S$): POKEX, ORD(S$(I)): X=X+1: NEXTI: POKEX, 253: POKEX
+1, 217: POKEX+2, 253: GOTO2
20 REM *** PROGRAM ELEJE ***
    
```

1. lista

```

30 REM *** SPECTRUM INPUT DEMO 2. ***
40 REM *** FUGGVENYTABLAZAT ***
50 VT=0: GOSUB3: FORX=0 TO100 STEP5
60 GOSUB2: PRINT"X="; X, "Y="; Y
70 NEXTX: GOTO20
    
```

```

30 REM *** SPECTRUM INPUT DEMO 1. ***
40 VT=0: PRINT"NUMERIKUS ";: GOSUB3: PRINTA
50 VT=1: PRINT"STRING ";: GOSUB3: PRINTAS
60 GOTO20
    
```

2. lista

alábbi program ezt valósítja meg TV-Compu-teren.

Nézzük a működési elvét (1. lista). A TV-BASIC a függvényeket nem tokenizálja, hanem ASCII kód formában helyezi el, a BASIC sorban tokenizálnak viszont a

3. lista

A program használata: az első két sorát különös gonddal gépeljük be, különben futáskor felülírja önmagát! Saját programunkat a 20. sortól kezdve írhatjuk. Az INPUT rutin meghívása:

– ha numerikus kifejezést akarunk bekérni: VT=0: GOSUB 3.

A 2. és 3. listán látható program jól szemlélteti a rutin használatát. A második listán látható programmal egy függvény-táblázatot készíthetünk. A 3. program numerikus és sztring típusú kifejezés használatát mutatja be.

NAGY ISTVÁN

Memóriaterület mentése C64-en

Az alábbi – szubrutinként is alkalmazható – program igen hasznos lehet olyankor, amikor a gép memóriájából adatokat vagy gépi kódú programokat kívánunk adathordozóra menteni. Természetesen ezt nemcsak lemezre menthetjük, hanem bármilyen más perifériára is. Ilyenkor azonban a 3. sort módosítanunk kell. A kezdő és a végcím beállítására az 1. sorban nyílik lehetőségünk. Mivel a létrejött fájl PRG formátumú, a visszatöltése is igen egyszerű.

SURÁNYI PÉTER

```

1 INPUT"KEZDET"; KEZ: INPUT"VEG"; VEG
2 LO=KEZ AND 255: HI=INT(KEZ/256)
3 OPEN 1, 8, 1, "PROGRAMNEV"
4 PRINT#1, CHR$(LO); CHR$(HI);
5 FOR X=KEZ TO VEG: A=PEEK(X)
6 PRINT#1, CHR$(A);: NEXT
7 CLOSE 1
    
```

Hangdigitalizálás

Néha előfordul, hogy programunkat színesebbé, érdekesebbé szeretnénk tenni valamilyen hanghatással vagy beszéddel. A legegyszerűbb megoldásnak a hangdigitalizálás ígérkezik. Vagyis amikor a számítógépbe beolvasunk valamilyen perifériáról érkező jeleket, majd valamilyen formában elraktározzuk vagy megszólaltatjuk.

Persze tiszta beszédet vagy élethű hanghatást csak erre a célra készült műszerekkel készíthetünk. De ezek nélkül is próbálkozhatunk például a Primón.

A listán látható kis program a magnóról érkező jeleket szólaltatja meg a hangszóróján. A programot begépelés után mentjük ki vagy használjuk a menüben leírt M — parancsot, mivel a gépi kódú rész indulása után a gép „befagy”!

A digitalizálás programon belüli alkalmazására egy későbbi számban mutatok példát.
SZARKA ZOLTÁN

```

10 Cls
20 Print chr$(2); "Hangdigitalizálás"
30 Print chr$(18); "By Szarka Zoltán"
40 I=2871 : C=20000 : P=0
50 For I=1 to 25:Read A:P=P+A:Next I
60 If P<0 then Print "Hibas a DATA
sor !";chr$(7); Stop
70 For I=C to C+34:Read A:Poke I,A:
Next I
80 Print "A Programot elindítás után
nem tudja megállítani, ezért elő-
ször mentse ki kazettára!"
90 Print:Print chr$(2); "Menu:"
100 Print chr$(18); Print "M - mentes"
110 Print "1 - a Program indítása"
120 If inkey$="" Then 120
130 If inkey$="1" then 160
140 If inkey$<>"M" then 120
150 InPut "Milyen neven mentsem";a$:
If len(a$)>16 then a$=Right$(a$,16)
:Print "Indítsd el a magnót!";saveA$
155 Cls:Goto 90
160 Cls:Print "Helyezzen a magnoba vala-
milyen kazettát és indítsa el a
lejatszást!"
170 A=CALL(C)
180 REM Adatok
190 Data 33,59,64,62,127,119,211,0,243
200 Data 14,0,6,3,0,16,253,219,0,31,31
210 Data 31,48,6,62,109,211,0,24,236
220 Data 62,127,211,0,24,230
    
```

Ha van grafikus kártya

Az alábbiakban egy új és egy korábban már megjelent program IBM-változatát ismertetjük.

Az első listán az 1987/7. szám 7. oldalán található szoftver IBM-átírata látható. Ugyanezen az oldalon néhány grafika is van, ami szintén az alkalmazott módszer segítségével készült.

Az új program is grafikát készít, csupán ez esetben nincs lehetőségünk indulási paraméterek változtatására, így a végeredmény egyetlen kép. A programok egyébként bármelyik IBM BASIC interpreterrel futtathatók (BASIC, BASICA, GW-BASIC), amennyiben a gépnek van grafikus kártyája.
B. B.

```

10 REM BB SOFTWARE 1988
20 SCREEN 1:KEY OFF
30 INPUT A,B:CLS
40 I=1+.02
50 X=COS(I*A)*150+150
60 Y=SIN(I*B)*90+90
70 IF I=.02 THEN K=X:L=Y
80 LINE (K,L)-(X,Y),1
90 K=X:L=Y:GOTO 40
    
```

1. lista

```

10 REM BB SOFTWARE 1988
20 KEY OFF
30 SCREEN 1:COLOR 9:CLS
40 FOR I=0 TO 319 STEP 3
50 LINE (I,0)-(319-I,200),1
60 NEXT
70 FOR I=199 TO 0 STEP -2
80 LINE (0,I)-(319,199-I),1
90 NEXT
    
```

COMMODORE 16

Örökélet- kód

Szeretnénk folytatni a magazin korábbi számában elkezdett örökélet-kód sorozatunkat. A mostani POKE-okat Tuba Imre olvasónk küldte be, méghozzá extrákkal. Egyes játékoknál az egyéb szolgáltatások végtelenítésre is lehetőséget nyújtanak. Most pedig lássuk a medvét:

AIRWOLF		7151,0
ALIEN ATTACK		6106,165
		6144,165
APOLLO MISSION		13334,173
	energy	10281,234
BERKS III.		9846,0
BLAZE		8787,173
	bonus	13586,173
CANOE		5277,48
		5283,48
CLIMB IT		11737,48
COMMANDER		10471,173
	bonus	9171,173
COMMANDO		13197,173
CORMAN		4948,165
CYBORG		10692,173
	smt	16220,173
EXORCIST		16214,0
		16215,208
FANTATRON		5685,48
FINGERS		6095,0
		6281,0
		10187,0
FURY		7068,173
	bomb	8527,165
G-MAN		13507,173
GAMES DESIGNER		10676,173
GNASHER		8946,165
GULLWING		8870,173
	time	9030,165
GUNSLINGER		10425,173
		11322,173
HARBOUR ATTACK		6499,169
		6500,2
INVASION 2000		7371,165
JET SET WILLY		10874,173
KIKSTAR		10403,165
KUNG FU		8692,173
	energy	8685,169
		8686,0
LEGIONNAIRE		10416,173
LOCO CO-CO		12546,173
	time	12253,173
LOCOMOTION		4506,234
	szén és víz	8318,189
MANIC MINER		10889,234
	air	12082,199
MISSION MARS		4224,173
		4473,173
MONKEY		10656,173
OLYMPIC SKIER		6366,169

BASIC és gépi kód

Legutóbb a C16 hexadecimális konverziós függvényeinek megfelelő USR-függvények C64-es változatát ismertettem. Említettem az USRDEF eljárásnak azt a hibáját, hogy nem fér össze az FN-függvényekkel. Ez késztetett arra, hogy egy távolabbi tervemet előbb valósítsam meg: most az általam leggyakrabban használt FN-függvények gépi kódú megfelelőjét mutatom be.

A hiba okát nem sikerült pontosan megállapítanom, de azt már tudom, hogy a függvény kiértékelésénél van valami zűr. A VC-20-nál további problémákba ütköztem, azon a C64-en sikeres megoldás is rövid idő után a rendszer befagyásához vezetett.

A függvények

Az 1. listán látható a három FN-függvény. Az első kettő egy kétbájtos, előjel nélküli egész szám magasabb (High) és alacsonyabb (Low) helyértékű bájtyát adja eredményül. A harmadik dr. Úry László könyveiben FN DEEK néven szerepel, és egy „dupla pontosságú PEEK”-et definiál, ami tárcímek kiszámításához alkalmazható.

A gépi kódú rutinok

Az FNL és FNH függvényeknek megfelelő USR rutin C64-en disassemblált listája a 2. listán látható. A C16-os változat a program tárbeli helyétől és a ROM-rutinok belépési címétől eltekintve pontosan megegyezik vele. Az utóbbiakat a táblázat alapján hasonlíthatjuk össze.

A rutin mindkét függvény kiszámítására alkalmas, belépési címük 3 bájttal különbözik, szétválasztásukra a BIT utasítást (\$2C) alkalmaztam. Erről a módszerről a magazin 1987/10-es számában írtam részletesebben, akit érdekel, ott utánanézhet.

\$C12A, C12D: az A regiszterbe töltöm annak a nullalapon lévő bájtnak a címét, amely a GETADR rutin végrehajtása után kívánt értéket tartalmazza majd. Figyeljük meg, hogy az 1. listán lévő sorrendtől eltérően itt az alacsonyabb helyértéket visszaidőző függvény belépési pontja áll előbb. Ennek nincs jelentősége, két utasítás felcserélésével a sorrend megváltoztatható.

\$C12F—C133: az A regiszter tartalmát a veremben őrizzük, míg a GETADR rutin a

FAC tartalmát címformátumúvá alakítva a \$14—15 címekre teszi le.

```
100 def fnh(o)=int(o/256)
110 def fnl(o)=o-256*fnh(o)
120 def fnp(o)=peek(o)+256*peek(o+1)
```

1. lista

```
c12a a9 14      lda #$14
c12c 2c         byte $2c
c12d a9 15      lda #$15
c12f 48         pha
c130 20 f7 b7   jsr $b7f7
c133 68         pla
c134 aa         tax
c135 b4 00      ldy $00, x
c137 4c a2 b3   jmp $b3a2
```

2. lista

```
c13a 20 f7 b7   jsr $b7f7
c13d a0 01      ldy #$01
c13f b1 14      lda ($14), y
c141 85 62      sta $62
c143 88         dey
c144 b1 14      lda ($14), y
c146 85 63      sta $63
c148 84 0d      sty $0d
c14a a2 90      ldx #$90
c14c 38         sec
c14d 4c 49 bc   jmp $bc49
```

3. lista

```
10bf 20 e4 9d   jsr $9de4
10c2 a0 01      ldy #$01
10c4 b1 14      lda ($14), y
10c6 48         pha
10c7 88         dey
10c8 b1 14      lda ($14), y
10ca a8         tay
10cb 68         pla
10cc 4c 76 9a   jmp $9a76
```

4. lista

\$C134—C135: a veremből visszatöltött címet áttesszük az X regiszterbe, majd a \$00 (X tartalma) címről annak tartalmát az Y regiszterbe töltjük. Tehát a tényleges címet az X tartalmazza, a \$00-ra csak formálisan van szükség.

\$C137: egy olyan ROM-rutinra ugrunk, mely A tartalmát nullázva az A/Y regiszterpár tartalmát lebegőpontossá alakítva a FAC-ba teszi. Rövidebben: az Y-ban lévő 1 bájtos bináris értéket alakítja lebegőpontossá.

Az FNP-nek (vagy ha jobban tetszik FN DEEK-nek) megfelelő program már jobban eltér egymástól a két géptípuson. A C64-es változat a 3. listán, a C16-osé a 4. listán található. Nézzük először a C64-est.

\$C13A—C146: A GETADR rutinnal a már ismert módon a \$14—15-be visszük a függvény paraméterét. Itt most a keresett cím címe lesz. Indirekt indexelt módon kiolvassuk a keresett értéket, és a \$62—63 címeken helyezük el. Vigyázat, most a \$62-n van a magasabb helyértékű bájt.

\$C148—C14D: ez az utasítássorozat majdnem pontosan megegyezik az előző részben közölt DEC rutin befejező utasításaival, az eltérés csak annyi, hogy most a típusjelzőt az Y regiszterből nullázzuk. Az ezt követő előkészítő utasítások után a \$BC49-en levő konvertáló rutinra ugrunk.

C16-on valamivel egyszerűbb a helyzet. Itt kihasználhatjuk azt, hogy a konvertáló ROM-rutin elvégzi azokat a műveleteket, melyeket a 64-esen nekünk kellett elvégeznünk, ezenkívül az A/Y regiszterpár tartalmát is leteszi \$62—63-ra. C64-en is van olyan rutin, amely ezt elvégzi, de van egy apró hibája: az eredményként a FAC-ba tett számot kétbájtos előjeles egészként értelmezi. Ezt a rutint használja a FRE függvény is, melynek működését némely C64-es tulajdonosa nem éppen dicséző szavakkal illeti.

A betöltőprogramok

A BASIC nyelvű betöltőprogramok az 5. listán és a 6. listán találhatóak, és mindhárom függvény kódját tartalmazzák. Mivel a

ROM rutinok összehasonlító címtáblázata

c64	c16
\$b7f7	\$9de4
\$b3a2	\$9a81
\$bc49	-
-	\$9a76

```

100 rem fnl + fnh + fnp      c64
110 z=256 : kc=49450
120 k=kc : print chr$(147)
130 for l=1 to 4
140 s=0 : read a$,b
150 print chr$(19);peek(63)+z*peek(64);
160 for i=1 to len(a$) step 2
170 a=usr(mid$(a$,i,2))
180 poke k,a : s=s+a : k=k+1 : next
190 if b<>s then print "hiba!" : stop
200 next
210 def usr1=kc : def usr2=kc+3
220 def usr3=kc+16
230 data a9142ca9154820f7b768aab4,1411
240 data 004ca2b3,417
250 data 20f7b7a001b114856288b114,1384
260 data 8563840da290384c49bc,1076

```

5. lista

rutinok helyfüggetlenek, címmódosításra nincs szükség. Az elhelyezést a korábban megjelentekhez igazítottam, de ezen értelem szerűen lehet változtatni.

A C64-es változat futtatása előtt — az előző részben megadott előírásnak megfelelően — töltsük be és futtassuk le a

HEX\$+DEC, majd az **USRDEF**—2 betöltőprogramokat. Ezután szám nélkül **DEF USR** utasítással definiáljuk a **DEC** rutint.

Ezután lehet indítani a mostani betöltőprogramot. Lefutása után **USR1** néven hivatkozhatunk a fentebb leírt **FNL**-re, **USR2**-vel az **FNH**-ra, **USR3**-mal az **FPN**-re.

```

100 rem fnl + fnh + fnp      c16
110 z=256 : kc=4271
120 k=kc : print chr$(147)
130 for l=1 to 4
140 s=0 : read a$,b$ : b=dec(b$)
150 print chr$(19);peek(63)+z*peek(64);
160 for i=1 to len(a$) step 2
170 a=dec(mid$(a$,i,2))
180 poke k,a : s=s+a : k=k+1
190 next
200 if b<>s then print "hiba" : stop
210 next
220 def usr1=kc : def usr2=kc+3
230 def usr3=kc+16 : end
240 data a9142ca9154820e49d68aab4,556
250 data 004c819a,167
260 data 20e49da001b1144888b114a8,544
270 data 684c769a,1c4

```

6. lista

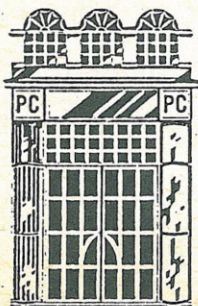
A C16-os változatnál természetesen csak az **USRDEF**—2-t kell előzetesen futtatni. Betöltése előtt **POKE 44,17: POKE 43,52,0: NEW** utasítássorozattal biztosítsunk helyet a gépi kódnak. Ez a betöltőprogram is elvégzi az **USR1**, **USR2** és **USR3** definiálását, az előző bekezdésben leírt módon.

BARNA LÁSZLÓ

Az Ön feje nem adattárolásra való, hanem fontos döntések hozatalára. Joggal várhatja el, hogy kezében legyen az eszköz, ami munkáját minőségivé teszi, döntését megalapozza. A számítástechnikában viszont a széles választékból nem könnyű a legjobb mellett dönteni.

az Ön fejével gondolkodtunk,

amikor létrehoztuk az első, Közép-Európában egyedülálló számítástechnikai szalon. Meghívtuk a legfontosabb gyártókat és forgalmazókat, hogy a választék együtt legyen áttekinthető, kipróbálható, tanulmányozható, összehasonlítható. Felkészült szakembereink várják az érdeklődőket, a leendő vásárlókat. Reméljük, döntésünk új korszakot nyit az Ön mindennapi munkájában.



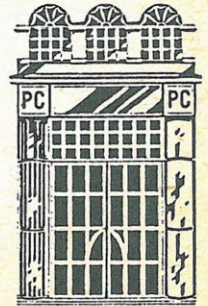
PC szalon

Budapest XIII., Sallai Imre u. 6.

☎ 315-136, 310-776

Lépjön új korszakba velünk.

NOVOTRIDE



A Precomp Plus előfordító program II. GET-BASIC-LINE

Plus/4, VC-15XX

Ez a gépi kódú program (2. lista) a már BASIC-ben olvasásra megnyitott fájlból kiolvas egy BASIC programsort (CHR\$(0)-ig). A program végét mint 0 hosszúságú programsort ismeri fel.

Az INIT részben lefoglalt C\$-be írható be az aktuális BASIC sor.

Az 1. táblázat 1-es része egy BASIC sort tartalmaz. Ez a sor a SYS 828 utasítással kerül be a C\$-be. A változótáblázat memóriaképét az 1/2-es rész mutatja beolvasás előtt és után. A sztring tényleges helyét mutató cím memóriaképe az 1/3-as képen látható. A BASIC sor végét jelölő CHR\$(0) után a sztring teljes hosszaiig space (CHR\$(32)) karakterek állnak. Az elemző-program erről a területről olvassa ki bájtónként a kifejezés elemeit. A munkasztring (C\$) hosszát az 1/4-es kép mutatja.

SCANNER

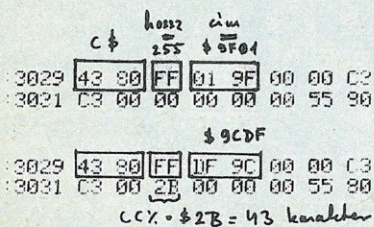
400-420; 550-870; 3180-3340 (lásd a 3. listát)

Az elemző ciklus

Az elemzés egy rövid cikluskeretben kezdődik. A külső „réteg” csak azt vizsgálja meg, hogy véget ért-e a beolvasás, a BASIC sor hossza egyenlő-e nullával (400-420).

1. táblázat

```
10 if a=0 then gosub100 : Print"a c$ egyenlo ezzel": end
ready. (1)
```



LÁNCOLÁSI CÍM	SORSZÁM
:9CDF	2D 02 0A 00 2B 20 41 B2
:9CE7	30 20 A7 20 3D 31 30 30
:9CEF	20 3A 20 99 22 41 20 43
:9CF7	24 20 45 47 59 45 4E 4C
:9CFF	4F 20 45 5A 5A 45 4C 22
:9D07	3A 20 80 00 20 20 20 20
:9D0F	20 20 20 20 20 20 20 20
:9D17	20 20 20 20 20 20 20 20
...	...

```
len(c$) = 43 (4)
```

Ha nem, a vezérlés a különlegesen kezelt karaktereket és kulcsszavakat vizsgáló ciklusra kerül (540-870 és 3180-3320). Az elemző meghatározza a munkasztring helyét a memóriában, és a munkasztringben lévő bájtokon sorba menve ismeri fel a kifejezéseket alkotó elemeket.

Kifejezések

Ha olyan bájtot talál, amely az elemzőt vezérli, közvetlen vezérlésátadással a megfelelő részre kerül a vezérlés, ahol visszalépés nélküli algoritmussal a kifejezésnek megfelelő szintaksziszmaszk szerint a hibaelemzés és az új kifejezés generálása valósul meg.

Különleges eljárások

Az előfordító helyes működéséhez az idézőjelben álló szövegek felismerése szükséges. Ezt kapcsolók segítségével valósítottam meg (MK%, F%).

GENERATOR

Az előfordító minden egyes kifejezéshez egy sorszámot rendel. A hozzárendelés elvégzéséhez az előfordítandó programot

GET 1 BASIC sor = 548 828

```
00 033c lda #$00
01 033e ldy #$09
02 0340 sta ($2d),y ; beolvasott BASIC
                                sor hossza = 0
03 0342 iny
04 0343 sta ($2d),y
05 0345 ldy #$03
06 0347 lda ($2d),y ; C$ címe a sztring
                                leíróba
07 0349 sta $22
08 034b iny
09 034c lda ($2d),y
10 034e sta $23
11 0350 ldx #$02 ; 2-es csatorna
12 0352 jsr $ffc6 ; CHKIN
13 0355 ldy #$00
14 0357 sty $90 ; ST nullázása
15 0359 jsr $ffe4 ; GET
16 035c sta ($22),y ; a beolvasott báj
                                t elhelyezése C$-be
17 035e bne $0368 ; ha nem 0, köv.
18 0360 cpy #$01 ; vége ha a CCX=0
19 0362 beq $0370 ; (a hossz = 0)
20 0364 cpy #$04 ; minimum 4 báj
                                t
                                (láncolási cím +
                                sorszám 2+2)
21 0366 bcs $036b
22 0368 iny
23 0369 bne $0359 ; vissza a ciklusba
24 036b tya
25 036c ldy #$0a
26 036e sta ($2d),y ; hossz elhelyezése
27 0370 jmp $ffc0 ; CLRCH, vissza az
                                elemzőhöz
```

2. lista

RENUMBER 10, 10 paranccsal át kell számozni. A 10-es lépésköz közé lesznek beillesztve az új sorok:

Eredeti:

```
10 FOR i=1 TO 100 :
PRINT i : NEXT
20 a=100 : b=200 :
INPUT c : IF c=300 THE
END
```

Generált:

```
10 FOR i=1 TO 100
11 PRINT i
12 NEXT
20 a=100
21 b=200
22 INPUT c
23 IF c=300 THEN END
```

Ennek a jelentősége, hogy a direktíva-elhelyezés így könnyebben megvalósítható és a CBM BASIC 2.0-ra való redukálás nem okoz gondot.

Az elemző által nyújtott információk alapján az elemzéssel majdnem párhuzamosan generálódnak az új sorok és a redukált kifejezések.

A GENERATOR az egyes utasításokat elemző programrészekbe van beépítve.


```

395 rem " * Forditási cilus
396 .
400 c#=u# " * Munka $ torlese
401 s9so% " * BASIC sor atvitale a munka #-ba
402 if cc%=0 then 430 " * Program vege ha a hossz=0 -val
410 h4#="" " * Generalt $ torlese
411 gosub 550 " * Sor elemzese
420 goto400 " * Kovetkezo sor
425 .
430 Print#4,chr$(0);chr$(0);chr$(0);rem" * Generalt fajl lezarasa
440 close2
441 close4
450 Print"###",tab(33);t1# " * Else menet vege
455 .
460 if m2%=1 then 2110 " * Masodik menet szukseges
470 Print#15,"n:Pre."+Pn#+"="+P1# " * Atnevezes (1.nev => Pre.nev)
471 goto2660 " * Program vege
480 rem=====
485 .
489 rem--- hibacsatorna olvasasa ---
490 Input#15,e0,e1#,e2,e3
500 if e0<20 then return
510 char1,0,24,"d.error:"+"###"+e1#+str$(e0)
520 close2
521 close4
522 close15
530 .
532 :getkeya#
533 Print"#####"
534 end " * Rendellenes befejezes
539 rem=====
540 rem--- also menet ---
545 .
550 Print"###:###";
560 ck=peek(vk+2)+peek(vk+4)*256+1 " * Olvasasi ciklus kezdete
570 cv=ck+cc%-2 " * Ciklus vege
571 f1%=0 " * If mutato
572 e1%=0 " * ELSE mutato
573 d1%=0 " * Direktiva kell-e
574 th%=0 " * THEN mutato (GOTO)
575 mk%=0 " * 'Macskakorom'-mod (QUOTE) mutato
580 sl%=asc(mid$(c#,3)) " * Szomszam also bajt
590 sh%=asc(mid$(c#,4)) " * Szomszam felso bajt
600 so=sl#+sh%*256 " * Szomszam
601 Printso
610 c1%=sl#;ch%=sh% " * Szomszam elmentese
612 f%=58 " * 'text' vagra terminator kapcsolo
613 h4#="" :h5#="" :h6#="" :h7#="" " * Se9edsztrin9ek
616 .
620 Print#4,chr$(10);chr$(10);chr$(s1#);chr$(sh#);
621 rem " * Szomszam lemezre irasa
622 .
630 Forl=ck+3to cv " * Munka $ olvasasa karakterenkent
640 c#=peek(l) " * Aktualis karakter / token
650 if c%=58 and l=cv then 1030 " * Sorvege (-) -tal
660 if c%=24 and mk%=0 then 290 " * 'Macskakorom' kezdete
670 if c%=f% then 950 " * ' ' vagra ' '
680 if c%=34 then f%=34:goto1020 " * ' '
690 if c%>203 and c%<213 then 1060 " * BASIC 3.5-os fu999venyek
710 if c%=131 then:m2%=1:goto2310 " * DATA = 2.menet szukseges
720 if c%=143 then 2920 " * REM = torles a sor vegei9
730 if c%=129 then 1100 " * IF
740 if c%=215 then 1220 " * TRAP
750 if c%=217 then 1230 " * TROFF
760 if c%=214 then 1240 " * RESUME
770 if c%=216 then 1250 " * TRON
780 if c%=235 then:m2%=1:goto1420 " * DO
790 if c%=236 then 1900 " * LOOP
810 if c%=222 then 2260 " * GRAPHIC
920 if c%=213 then 2510 " * ELSE
930 if c%=161 then 2660 " * GET
940 if c%=251 then 2770 " * USING
950 if c%=140 then:m2%=1 " * RESTORE
960 .
970 goto 1020
973 .
975 rem--- 'text' ---
980 i%=1
991 mk%=1

```

```

882 h4#=h4#+chr$(34)
883 :
890 c%=peek(l+i%)          :rem" * Kovetkezo karakter
891 i%=i%+1
900 h4#=h4#+chr$(c%)      :rem" * Generalt utasitas 99ujtese
910 if c%>34 and l+i%-1 < cv then 890
920 l=l+i%-1              :rem" * Szamlalo visszaallitas
921 mk%=0                  :rem" * "Macskakorom" mutato torlese
930 goto1040              :rem" * Vissza a ciklusba
940 :
945 rem--- / / ---
950 if c%=34 then f%=52:goto1030
960 if f1%=1 then h6#=h4#;h7#="" :goto 1030 :rem"* IF van a sorban
970 if h4#="" then h4#=""
975 :
980 Print#4,h4#,chr$(0);   :rem" * Generalt sor lemezre irasa
981 h4#=""
990 sl%=sl%+1             :rem" * Kovetkezo utasitas uj sorba
991 if sl%>255 then:sl%=sl%-255:sh%=sh%+1
1000 Print#4,chr$(10);chr$(10);chr$(sl%);chr$(sh%);
1010 goto1040
1020 :
1025 rem--- 99ujtes ---
1030 h4#=h4#+chr$(c%)
1031 if f1%=1 then h7#=h7#+chr$(c%):rem" * Ha IF se9ed $-be 99ujtes
1040 nextl
1045 :
1050 if d1%=1 then h4#=""+"h4# :rem" * Direktiva kell az utasitas ele
1060 :
1070 Print#4,h4#,chr$(0); :rem" * Sor ve9e
1080 return
1090 :
1095 rem--- if ----
1100 i%=1
1101 f1%=1
1110 h5#=chr$(c%)
1120 c%=peek(l+i%)
1121 i%=i%+1
1122 if c%>203 and c%<251 then h5#=""+"h5#;d1%=1:rem"* 3.5 direktiva kell
1130 if c%>167 then h5#=h5#+chr$(c%):goto 1120 :rem"* THEN
1140 h5#=h5#+chr$(c%)
1150 c%=peek(l+i%)
1151 i%=i%+1
1152 if c%=32 or c%=137 then 1150 :rem" * SPACE vagy GOTO
1160 if c%>47 and c%<56 then h5#=h5#+" "+chr$(137)+chr$(c%):l=l+i%-1:th%=1:goto1200
1161 rem " * THEN sorsszam => THEN:GOTO sorsszam
1170 l=l+i%-1
1180 Print#4,h4#+h5#;chr$(52);
1181 h4#=""
1182 goto670
1190 :
1200 Print#4,h4#+h5#;
1201 h4#=""
1202 goto1040
1210 :
1215 rem--- trap/troff/tron/resume ---
1220 h#="" trap": goto 1260
1230 h#="" troff": goto 1260
1240 h#="" resume":goto 1260
1250 h#="" tron"
1251 :
1260 hu%=0:gosub 2960      :rem" * Hibauzenet kiadasa
1262 :
1270 i%=1
1280 c%=peek(l+i%)
1291 i%=i%+1
1290 if c%>58 and l+i%-1<cv then 1290:rem" * Olvasas a kovetkezo utasitasig
1300 l=l+i%-1
1310 if f1%=0 then h4#=h4#+chr$(52)
1320 if l=cv then 1050
1330 if c%=52 and f1%=0 then 970
1340 goto 1040
1350 :
1360 rem--- basic 3.5-os fu9avenyek ---
1361 hu%=0:h4#="" fu9aveny":gosub 2960
1370 if f1%=0 then h4#=""+"h4#":goto 1030
1380 h4#=h6#+" "+h7#
1391 goto1030
1390 :

```

```

10 rem * if-then *
20 if a=0 then a=a+1:goto 40
30 b=100:c=200:if d$="1" then 50
40 d$=""
50 :
60 rem * tron troff trap resume *
70 tron:trap 90:for i=1 to 10:Printi:next
80 troff
90 resume next
100 :
110 rem * basic 3.5 függvények *
120 a=instr("1234"),"3"):b=100:c=joy(1)
130 if a <> 0 then a=r9r(0):Printa
140 Print rclr(0)
150 :

0 poke129,128:rem>>b
10 :
20 if a=0 then a=a+1:goto 40
30 b=100
31 c=200
32 if d$="1" then:goto50
40 d$=""
50 :
60 :
70 :
71 :
72 for i=1 to 10
73 Printi
74 next
80 :
90 :
100 :
110 :
120 :a=instr("1234"),"3")
121 b=100
122 :c=joy(1)
130 if a <> 0 then::a=r9r(0):Printa
140 :Print rclr(0)
150 :
    
```

4. lista

Korlátok

Az előfordító:

- 500 = DATA sorszámot
- 500 = DO—LOOP ciklust
- 500 = hibajzenetet

képes kezelni. Az egyes utasításoknál be-tartandó Precomp szintakszist az utasítá-soknál ismertetem. Soronként maximálisan tíz utasítás használható.

Generált utasítások

IF... THEN szerkezet

Ha ebben a szerkezetben 3.5-ös függvény stb. van, a GENERATOR fordítási direktí-vát helyez el. Ha THEN-nel a vezérlést ad-ja át GOTO nélkül, GOTO generálódik a THEN után. Ez annak a következménye, hogy minden THEN után egy utasításhatá-roló lesz beírva, hogy a Plus-Comp 3.5-ös utasítás esetén is értelmezni tudja.

TRON TROFF TRAP RESUME

Ezek az utasítások az előfordítás után — a REM-hez hasonlóan — a generált prog-ramban nincsenek meg.

CBM BASIC 3.5-ös függvények

A függvények direktívába helyezését az 1360—1390-es sorok tartalmazzák. A H4\$-be generált utasítás elé direktíva ke-rül (::).

Az utasítások használatát a 4. lista mu-tatja. **BÁRÓ CSABA**

A magas szintű programozási nyelvek közül jó néhány előnyös tulajdonsága miatt kiemelkedik a C nyelv. Szinte minden nagy szofverház elkészítette a maga C fordítóját, kiépítette az egyre több szolgál-tatást nyújtó könyvtárrendszerét. A C nyelv kezd népszerűvé válni, egyre több program készül ezen a nyelven.

A C64 a hazánkban is egyre jobban elter-jedő IBM PC kompatibilis számítógép mel-lett kis gépnek tekinthető, azonban az egyé-

ni tanuláshoz, kis programok készítéséhez hasznos társ is lehet.

A Data Becker cég 1985-ben készített er-re a gépre C rendszert, amely képszerkesz-tőből, lemezkezelőből, fordítóból és szer-kesztőből áll. A gép korlátait is figyelembe véve, ez a kis rendszer alkalmas a C nyelv megtanulásához.

A továbbiakban néhány hasznos függ-vényt mutatok be.

MÉSZÁROS LÁSZLÓ

Képernyő törlése:

```

clr :
( putchar( CLR ); /* CLR az STDIO.C-ben definiált érték
)
    
```

Ezt és még számos hasznos függvényt makróként is megírhatunk:

```

#define clr() putchar( CLR )
#define home() putchar( HOME )
#define rvsort() putchar( REVERSEOFF )
#define rsort() putchar( REVERSEOFF )
    
```

Pozicionált kiírás a képernyőre:

```

kiir( sor, oszlop, form, string ) /* a sor, oszlop pozíciótól *
int sor, oszlop /* kezdve a form leíró speriint *
char *form, *string /* kiírja a string tartalmát *
( cursor( sor, oszlop );
  printf( form, string );
)
    
```

Fájlkezelés, szekvenciális fájl megnyitása:

```

sopen( nev, mod ) /* a mod 'r' vagy 'w' vagy 'a' *
char *nev, mod /*
( char fnev(25); /*
  int i; /*
  strcpy( fnev, nev ); /* nev másolása fnev-be *
  strcat( fnev, ".s." ); /* .s. ruzése fnev-hez *
  fstrlen( fnev ); /* f-be fnev hossza *
  fnev[i]=mod; /* megnyitási mód fnev-be *
  fnev[i+1]='n'; /* sorvég jel fnev-be *
  fnev[i+2]='0'; /* fnev végének jelölése **
  fopen( egység, csatorna, fnev ); /* file megnyitása *
  return i; /*
)
    
```

Célszerű az egység (lemezmeghajtó egység száma) és a csatorna nevű egész típusú változókat külső változóként definiálni, így értékük bárhol változtatható.

Parancs csatorna:

```

eopen() /* megnyitjuk az egység-en *
( int hibá; /* a 15. csatornát *
  hibá=open( egység, 15, EMFT ); /* az EMFT üres "" *
  return hibá; /*
)
    
```

Parancs küldése:

```

parancs(p) /*
char *p; /*
( int hibá; /*
  egység=B; /* a 8-as egységen megnyitjuk */
  hibá=eopen(); /* a parancs csatornát */
  printf( hibá, "%s", p ); /* es kiírjuk a parancsot */
  close( hibá ); /*
)
    
```

Relatív fájl készítése:

```

rcreat( nev, rekh ) /* a nev nevü *
char *nev; /* rekh rekordhosszu *
int rekh; /* relatív file létrehozása *
( char fnev(25); /* a 8-as egységen *
  int i; /* a 8-as csatornán *
  egység=B; /*
  csatorna=0; /*
  strcpy( fnev, nev ); /* nev másolása fnev-be *
  strcat( fnev, ".l." ); /* .l. másolása fnev-hez *
  fstrlen( fnev ); /* f az fnev hossza *
  fnev[i]=rekh; /* rekh fnev-be *
  fnev[i+1]='n'; /* sorvég *
  fnev[i+2]='0'; /* fnev vége *
  fopen( egység, csatorna, fnev ); /* file megnyitása *
  return i; /*
)
    
```

Pozicionális relatív fájlban:

```

rseek( rekp, adatp ) /* a rekp. ik rekord *
int rekp, adatp; /* adatp pozíciójára *
( char k(7); /* pozícionálunk *
  int hibá; /*
  hibá=eopen(); /* parancs csatorna nyitása *
  k(0)='F'; /*
  k(1)=csatorna; /*
  k(2)=rek(4,56); /* parancs összeállítás *
  k(3)=exp(56); /*
  k(4)=ostp; /*
  k(5)='n'; /*
  k(6)='0'; /*
  printf( k, k, hibá ); /* parancs elfordése *
  close( hibá ); /*
)
    
```

Fordítás vagy értelmezés?

Programok írásához — egyéb kellékek mellett — nem nélkülözhető valamilyen programozási nyelv használata. A nyelvek bábeli zűrzavaráról már könyvtárnyi irodalom szól, amelyet mi is gyarapítunk a jelenleg kurrens nyelvek folyamatos bemutatása során. Most nem nyelvekről, hanem azok számítógépes „megvalósításáról” (implementációjáról) lesz szó. A téma a számítástudomány területéhez tartozik. Van azonban néhány olyan vonatkozása is, amely minden számítástechnikus és szimpatizáns számára érdekes lehet.

Az alapprobléma onnan ered, hogy az emberek nem szívesen tanulják meg a gépek bináris „nyelvét”, a gépek pedig — érdelemnyilvánítás nélkül ugyan, de — maguktól nem értik az emberek nyelvét. Ezért az élő nyelvek helyett egyszerűsített, pontos és betartandó szabályokkal definiált nyelvek jöttek (és jönnek) létre a gépekkel való érintkezés (programozás) megvalósítására. Sajnos, a gépek egyedül még ezeket a nyelveket sem értették, a programozók pedig már nem akartak a nyelv emberi vonásából tovább engedni.

A konfliktus kétféle módon oldható fel:
— minden programozási nyelvhez meg kell építeni az őt közvetlenül értő speciális gépet, vagy

— programozás útján kell a gépeket „megtanítani” az egyes nyelvek megértésére.

Az első megoldás nagyon munka- és költségigényes, és csak akkor jöhetne szóba, ha sikerülne egy — vagy néhány — általánosan elfogadott programozási nyelvet bevezetni, amely hosszabb ideig uralná a szakmát. Mivel ennek az elmúlt 40 év tapasztalatai határozottan ellentmondanak, maradt a második, a „szoftver” út. Ha egy gépet megtanítunk egy programozási nyelv megértésére, azt mondjuk, hogy a kérdéses nyelvet az adott gépen „megvalósítottuk”. A nyelvek gépi megvalósítása általában bonyolult programok segítségével történik, és a számítástechnikán belül egy külön területet alkot. E terület részletei nem tartanak közérdeklődésre számot, de a programozási nyelvek feldolgozásának két alapvető módszerével célszerű megismerkedni.

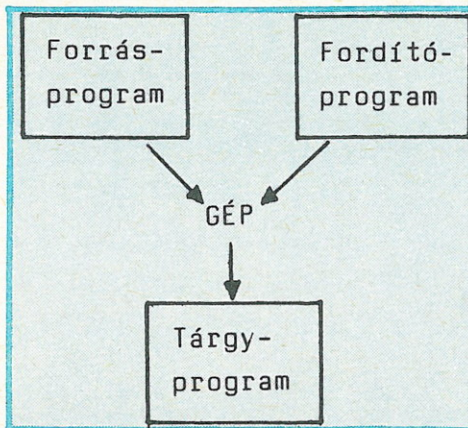
Bevezetésül — a közös szóhasználat érdekében — néhány alapvető fogalmat a táblázatban foglaltunk össze.

Annyit sokan tudnak, hogy egy nem gépi kódban írt programot a gépen vagy fordítás vagy értelmezés segítségével lehet csak futtatni. Mivel a két technika előnyök és hátrányok szempontjából sok tekintetben

kiegészíti egymást, érdemes megvizsgálni, hogy a programfejlesztés során mikor melyiket jó preferálni.

A fordítás folyamatát az 1. és a 2. ábra együttesen illusztrálja. Először a fordítóprogram létrehozza a tárgyprogramot, melyet a gépbe betöltve végrehajthatunk. Előfordul, hogy az 1. ábrán kapott tárgyprogram még nem gépi kódú, ezért a 2. ábra szerinti „futtatásra” csak újabb fordítás után kerülhet sor. (A C nyelv fordítói például első lépésben általában assembly nyelvre fordítanak.) A logikai teljesség kedvéért el kell mondanunk azt is, hogy a 2. ábra szerinti programvégrehajtás tulajdonképpen egy értelmezési folyamat, amely abban különbözik a 3. ábrán láthatótól, hogy az értelmező „programot” a gép áramkörei testesítik meg.

Maga az értelmezés (3. ábra) úgy zajlik le, hogy az értelmező program — melyet különösen BASIC esetén gyakran a gép ROM memóriájában tárolnak — utasításonként végighalad a forrásprogramon, közben elemzi annak utasításait, és behívja a rendszer azon alprogramjait, amelyek a gépből a kérdéses utasításnak megfelelő tevékenységet váltják ki.



1. ábra

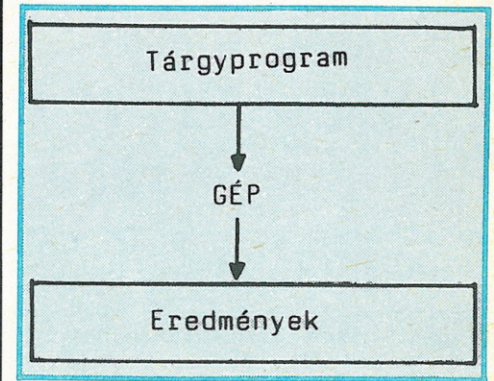
Ha mindkét technika előnyös és hátrányos vonásait át akarjuk tekinteni, négy alcímre van szükség.

Az értelmezés előnyei

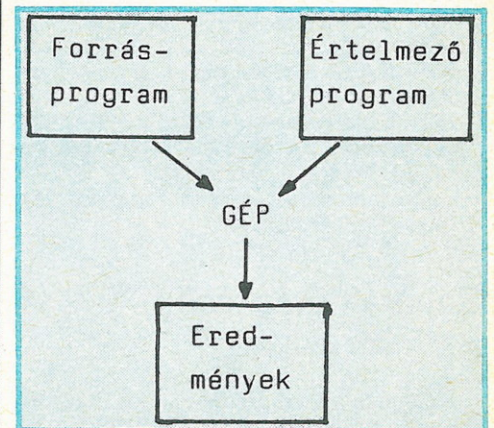
Mivel a forrásprogram — az értelmező automatikus működését nem tekintve — „közvetlenül” hajtódik végre, a programozó egyszerűen és hamar lát eredményt vagy hibát, hiszen a géppel, úgy tűnik, egy nyelvet beszél. Ennek megfelelően igen barátságos kapcsolat alakítható ki a rendszer és a programozó között. A programot egy célszerűen a nyelvre orientált szerkesztőprog-

ram segítségével „inkrementálisan” írhatjuk, ami azt jelenti, hogy akár az egyes utasítások vagy próbarészek után sem igényel különösebb erőfeszítést egy próba futás. A szerkesztő utasításonként ellenőrizheti programunk formai helyességét, és ezzel sok utólagos fáradságot takaríthat meg.

Az értelmezés technikai megvalósításából következik a dinamikus tárkezelés (a változó számára lefoglalt tárterület mérete csak a program futásakor válik ismertté) viszonylag egyszerű megoldása. A tárkezelő



2. ábra



3. ábra

rutinokat nem kell a tárgyprogramba beépíteni, azok az értelmező részei lehetnek.

A program hibáinak felderítésekor nem közömbös, hogy az értelmező rendszerek jó nyomkövetést tesznek lehetővé, hiszen a forrásprogram futás közben folyamatosan rendelkezésre áll.

Az értelmezés hátrányai

Mivel az értelmező program minden utasítást minden végrehajtáskor kénytelen újra és újra „megérteni”, a programunk futása számottevően lassul. Az értelmező nemcsak időben, de tárban is csökkenti a hasz-

nos kapacitást. Az utasítások végrehajtási sorrendje előre nem jósolható meg, ezért az értelmező rendszereknél az operatív tárban kell tartani az értelmező program mellett a teljes forrásprogramot is. Az előző pontban említett előnyökért tehát komoly árat kell fizetni.

A fordítás előnyei

Itt a tárgyprogram létrehozása és végrehajtása szétválasztható, ezért a tárgyigénye leve kisebb lehet. Mivel a fordítóprogram csak egyszer (vagy néhányszor) olvassa végig a forrásprogramot, az háttértárban is tárolható, ahova a generált tárgyprogram akár részletekben is kivihető. A fordítást két futtatás között csak akkor kell ismételni, ha a forráson változtattunk, ezért a fordítás időigénye — különösen a fejlesztés vége felé, az alapvető hibák kiszűrése után — nem kritikus. Így lehetőség van arra, hogy a fordító tömörség vagy futási idő szempontjából optimális tárgyprogramot hozzon létre. Ez a program későbbi gyakori

hanem csak az operációs rendszeren keresztül aktivizálhatók. A forrásprogram hibáinak jelzése a fordítás közben azért nehézkes, mert minden hibához igen sok „következmény” hiba is csatlakozhat, ami áttekinthetatlenné teszi a hibajelzéseket.

A dinamikus tárkezelés csak úgy oldható meg, hogy a fordító a tárgyprogramba beépíti a tárkezelő rutinokat. Ezért sok fordítással megvalósított nyelvben a dinamikus tárkezelést nem engedik meg, vagy csupán mutatók (indirekt címzés) segítségével realizálják.

A tárgyprogram futása során fellépő hibák jelzése is bonyolultabb, hiszen ilyenkor a forrásprogram már nem áll rendelkezésre, a programozó pedig a forrásprogram környezetében várna a hibajelzést.

Látjuk, hogy mindkét technika mellett és ellen nyomós érvek szólnak. Ilyen esetben a választás előtt további feltételeket kell elemezni, és akkor kiderül, hogy a programok fejlesztési szakaszában, amikor sokat módosítunk, csak tesztadatokkal, egy-egy ágat próbálunk futtatni, az értelmezés a hatékonyabb. Üzemszerű, ismétlődő futtatás esetén viszont, amikor a program már töb-

Forrásprogram:	valamilyen programozási nyelven írt program
Fordítóprogram:	a forrásprogramot egyik nyelvről egy másikra fordítja. A célnyelv nagyon gyakran a gép saját nyelve, az ún. gépi kód
Tárgyprogram:	a fordítóprogram eredményének általános elnevezése
Közbenső forma:	a tárgyprogram egy speciális esete, amikor a fordítóprogram nem közvetlenül állítja elő a gépi kódú programot, hanem először egy közbenső nyelvre fordít
Értelmező program:	a forrásprogramot "végrehajtja" a gépen úgy, hogy azt a gépet szimulálja, amelynek gépi kódja a forrásprogram nyelve

használata során halmozott időmegtakarítást eredményez.

Az értelmezésnél említett inkrementális fejlesztés itt ún. moduláris programozással pótolható, amelyben ugyan előre meghatározott szerkezettel, de kisebb egységekből lehet a végső programot összeállítani.

A fordítás hátrányai

A fejlesztői ciklus — amíg egy programból eredményt látunk — óhatatlanul meghosszabbodik. Különösen igaz ez akkor, ha a szövegszerkesztő, a fordító és a kapcsolatszerkesztő (linkage) programok nem egyetlen integrált rendszerben dolgoznak,

bé-kevésbé változatlan, a fordítás kétségtelenül célszerűbb.

Az ily módon való fejlesztés egyetlen akadálya jelenleg, hogy a nyelvekhez általában vagy csak értelmező (BASIC, PROLOG, LIST, LOGO stb.) vagy csak fordítóprogram (Pascal, C, COBOL, FORTRAN, PL/I stb.) tartozik. A fejlődés azonban kétségtelenül az ún. negyedik generációs nyelvek irányába mutat, ahol egy integrált fejlesztői rendszerben mind az értelmezés, mind a fordítás megoldható (dBASE, BASIC), illetve a fordítás és értelmezés egy speciális keveréke használható (FORTH), vagy egy fordító rendszeren belül az értelmezéshez közel álló üzemmód is megengedett (Turbo Pascal). **BAKOS TAMÁS**



szünetmentes áramforrás

Az MD Vállalkozási és Kereskedelmi Kft. a GUTOR GmbH. nyugatnémet céggel világszínvonalú szünetmentes áramforrások gyártását és magyarországi forgalmazását kezdte el.

A berendezések zárt konstrukciójuk révén bármely irodai helyiségben elhelyezhetőek. A gépekhez jól szervezett szervizhálózatot és folyamatos alkatrészellátást biztosítunk.

Felhasználási területek

Adatfeldolgozó, biztonságtechnikai, adó-vevő berendezések, valamint szükségvilágítás stb.

Teljesítményhatárok:

500 VA—10 000 VA

Szállítás:

1500 VA határig azonnal, raktárról

Információ kérhető:

MD Vállalkozási és Kereskedelmi Kft.

Budapest Pf. 217. 1536

Telefon: 666-617

Telex: 22-7676 mdh

Örökmozgó szellemek

Az előző cikkekben a szellemgrafikák, a sprite-ok forgatásáról esett szó. Még mielőtt azonban belevágnánk a nagyfelbontású képernyő forgatásának ismertetésébe, szenteljünk időt a C64-es szellemgrafikák felfelé-lefelé forgatásának. Érdeemes, hiszen ebben az esetben a bal, illetve a jobb felé forgatással ellentétben nem kell a bájtokon bit szintű műveleteket végeznünk, hanem elég teljes bájtokat mozgatunk.

Ahhoz, hogy felfelé forgathassuk a 21×3 bájtól álló szellemgrafikát, az első sorban lévő három bájtot kell először kimentenünk valahová — ez a hely természetesen a veremtároló lesz —, ezután már az egyes sorokban lévő bájtokat egy sorral előrébb másolhatjuk át. A legvégén pedig gondoskodnunk kell arról is, hogy az első sorból kimentett bájtokat a legutolsó sorba írjuk vissza a veremtárolóból.

Az elmondottakat az ábra szemlélteti. A sprite-ot a veremtároló felhasználásával forgatjuk felfelé. Az első sor három bájtját írjuk bele a veremtárolóba, majd onnan az utolsó sorba. AKT és KOV a programban alkalmazott mutatók. Az ábrán a számok a bájtok sorszámát jelölik.

A bájtok visszairásával óvatosan kell eljárunk, mert a veremtárolóból csak azt az információt olvashatjuk ki először, amit legutoljára írtunk be. Ezért is hívják a veremtárolót LIFO memóriának, az angol *Last In First Out* megnevezés rövidítésévé, amit magyarra fordítva *UBEK*-nek lehetne nevezni: az *Utolsóként Be*, *Elsőként Ki* jelentés miatt. A veremtárolóra jó példa egy olyan tányéroszlop, amire a pincérek időnként tiszta tányért raknak, de le is vesznek róla tányérokat. Erről a tányéroszlopról mindig csak a legfelső tiszta tányért lehet elsőként levenni.

Így, ha három bájtot helyezünk el a veremben — a szellemgrafika első sorának bájtjait —, akkor azokat onnan a későbbiekben csak fordított sorrendben vehetjük vissza, és erre a programozáskor figyelniünk kell. Amint ezt az 1. lista 490—530-as, illetve 740—790-es sorai mutatják.

A listán látható a gépi kódú program, amely a tizenegyedik 64 bájtnyi memóriaterületen levő szellemgrafikát forgatja felfelé. Ám tovább is fejlesztettük a programot azaz, hogy nemcsak forgat, hanem mozgat is. Így a szellemgrafika mintegy magától forog és mozog is a képernyőn, mivel ez a program a korábbiakhoz hasonlóan minden egyes IRQ megszakítás ideje alatt — azaz minden hatvanad másodpercben — végrehajtódik.

Az 1. listán látható gépi kódú program soraiban magyarázata:

240—340 Itt indítható a szellemgrafikát felfelé forgató és mozgó program. A belső megszakítás letiltása után az IRQ vektor korábbi értékét kimentjük egy időleges tárolóterületre (TEMP), majd az IRQ vektornak új értéket adunk.

360-420 A korábbi IRQ vektor értékét visszairjuk az időleges tárolóterületről. Ez egyúttal leállítja a forgató és mozgó rutinokat is.

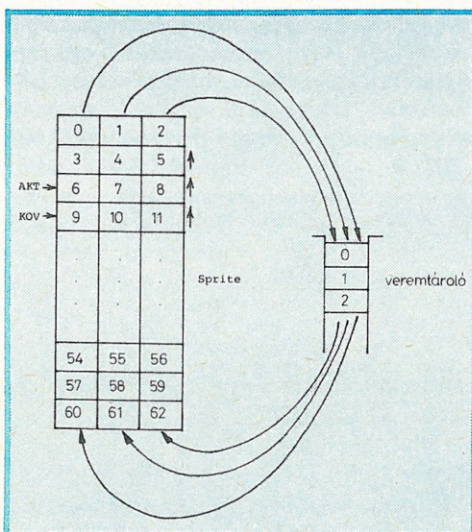
440-470 A saját IRQ rutinunk kezdete: kezdőértéket adunk az egyik mu-

adunk: AKT=KOV és KOV=KOV+3.

660—690 A tulajdonképpeni forgatás itt megy végbe: a következő sorban lévő bájtokat egy sorral előbbre másoljuk át, az aktuális sor megfelelő helyeire.

710—720 Végére értünk-e a szellemgrafikának, azaz minden sorát egy sorral előrébb másoltuk-e?

740—790 Itt vesszük elő a veremtárolóból — fordított sorrendben — az oda kimentett első sor bájtjait, és írjuk vissza az utolsó sorba, amin a KOV mutató áll.



tatónak, a szellemgrafika kezdőcímének értékét felveszi a KOV (a következő sort kijelölő) mutató.

490-530 A szellemgrafika első sorában álló három bájtot kimentjük, azaz ezt a három bájtot beírjuk a veremtárolóba.

550 Itt állítjuk be a szellemgrafika magasságát, a sorok számát, amit az X tárolóba teszünk. Ez azért kevesebb 21-nél, mert egyrészt két mutatót alkalmazunk és az egyik egy sorral előrébb áll, mint a másik, másrészt pedig előbb végezzük el a megfelelő műveleteket — a mutatók átállítását és a forgatást —, és csak azután vizsgáljuk meg, hogy véget értek-e a szellemgrafika sorai.

560—640 Két mutatónak, az AKT (aktuális) és a KOV (következő) sorok mutatóinak új kezdőértéket

1. lista

```

120: CD00          .OPT P1,00
          :
          : SPRITE-OK (MOB) FORGATASA
          : AZ IRQ ALATT FELFELE.
          : FORGATAS BE - SYS(52400)
          : FORGATAS KI - SYS(52505)
          :
190: CD00          *# = #CD00
200: CD00          IRQVEK = #0314
210: CD00          KOV    = #FB
220: CD00          AKT    = #FD
223: CD00          MOBOX  = #D000
225: CD00          MOBOY  = #D001
          :
240: CD00 78      FÖRGEBE SEI
250: CD01 AD 14 03 LDA  IRQVEK+1
260: CD04 AC 15 03 LDY  IRQVEK+1
270: CD07 8D 6E CD STA  TEMP
280: CD0A 8C 6F CD STY  TEMP+1
290: CD0D A9 2B LDA  #< UJIRO
300: CD0F A0 CD LDY  #> UJIRO
310: CD11 8D 14 03 STA  IRQVEK
320: CD14 8C 15 03 STY  IRQVEK+1
330: CD17 58 CLY
340: CD18 60 RTS
          :
360: CD19 78      FRGKI SEI
370: CD1A AD 6E CD LDA  TEMP
380: CD1D AC 6F CD LDY  TEMP+1
390: CD20 8D 14 03 STA  IRQVEK
400: CD23 8C 15 03 STY  IRQVEK+1
410: CD26 58 CLY
420: CD27 60 RTS
          :
440: CD28 AD 6C CD UJIRO LDA  MOB
450: CD2B 85 FB STA  KOV
460: CD2D AD 6D CD LDA  MOB+1
470: CD30 85 FC STA  KOV+1
          :
490: CD32 A0 02          LDY  #2
500: CD34 B1 FB          ELSO  LDA  (KOV),Y
510: CD36 48          PHA
520: CD37 88          DEY
530: CD38 10 FA          BPL  ELSO
540: CD3A A2 14          LDX  #20
560: CD3C 18          FLFELE CLC
570: CD3D A5 FB          LDA  KOV
580: CD3F 85 FD          STA  AKT
590: CD41 69 03          AOC  #3
600: CD43 85 FB          STA  KOV
610: CD45 A5 FC          LDA  KOV+1
620: CD47 85 FE          STA  AKT+1
630: CD49 90 02          BCC  TOV
640: CD4B E6 FC          INC  KOV+1
650: CD4D A0 02          LDY  #2
660: CD4F B1 FB          FEL  LDA  (KOV),Y
670: CD51 91 FD          STA  (AKT),Y
680: CD53 88          DEY
690: CD54 10 F9          BPL  FEL
          :
710: CD56 CA          DEX
720: CD57 D0 E3          BNE  FLFELE
740: CD59 A0 00          LDY  #0
750: CD5B 5B          UTOLSO FLA
760: CD5C 91 FB          STA  (KOV),Y
770: CD5E C8          INY
780: CD5F C0 03          CPY  #3
790: CD61 D0 FB          BNE  UTOLSO
          :
803: CD63 EE 00 D0          INC  MOBOX
805: CD66 EE 01 D0          INC  MOBOY
810: CD69 6C 6E CD          JMP  (TEMP)
          :
830: CD6C 00 02          MOB  .BYTE 192,2
840: CD6E 31 EA          TEMP .BYTE #31,FEA

```

803—805 A \emptyset . számú szellemgrafikát mozgatja X és Y irányban. Amennyiben a \emptyset . szellemgrafika adatait a tizenegyedik sprite-területről ($11 \times 64 = 704$ tárcíműl kezdődően) vesszük, akkor ez a sprite a képernyőn állandóan mozog majd a felfelé forgással együtt. A szellemgrafika mozgatása és forgatása szétválasztható, elérhető, hogy egy sprite forog, egy másik pedig mozog a képernyőn.

810 A korábbi megszakító rutinokra ugrik vissza a program vezérlése. Mivel a korábbi megszakító vektor címét kimentettük az ideiglenes tárolóterületre, a TEMP-re, emiatt ez a program hajlandó lesz minden programmal együttműködni, azokkal is, amelyek

```
1000 FOR I= 52480 TO 52592
1010 READ X:POKE I,X:S=S+X:NEXT
1020 DATA 120,173, 20, 3,172, 21, 3,141
1030 DATA 110,205,140,111,205,169, 40,160
1040 DATA 205,141, 20, 3,140, 21, 3, 88
1050 DATA 96,120,173,110,205,172,111,205
1060 DATA 141, 20, 3,140, 21, 3, 88, 96
1070 DATA 173,108,205,133,251,173,109,205
1080 DATA 133,252,160, 2,177,251, 72,136
1090 DATA 16,250,162, 20, 24,165,251,133
1100 DATA 253,105, 3,133,251,165,252,133
1110 DATA 254,144, 2,230,252,160, 2,177
1120 DATA 251,145,253,136, 16,249,202,208
1130 DATA 227,160, 0,104,145,251,200,192
1140 DATA 3,208,248,238, 0,208,238, 1
1150 DATA 208,108,110,205,192, 2, 49,234
1160 DATA 0
1170 IF S<> 15185 THEN PRINT"HIBAS ADAT !":END
1180 PRINT"OK !!"
```

2. lista

már átírták a megszakító vektor kezdőcímét \$EA31-ről valamilyen más címre.

830 A felfelé forgatandó szellemgrafika memóriaterületének kezdőcíme: $2 \times 256 + 192 = 704$. Ez megfelel a tizenegyedik sprite-területnek: 11×64 . Ezt a két bájtot módosítva áthelyezhetjük a forgatandó sprite-ot egy másik memóriaterületre.

A 2. listán látható ennek a gépi kódú programnak a BASIC betöltő programja. A szellemgrafikák lefelé forgatásának programját itt nem közöljük, csak könnyítésül megjegyezzük, hogy a lefelé forgatás is az ittenihez hasonló módon oldható meg, azzal a különbséggel, hogy ott a szellemgrafika utolsó sorát mentjük ki először a veremtárolóba, majd a szellemgrafika sorait másoljuk át. Az utolsó előtti sort kell először átmásolni az utolsó sorba, és így visszafelé haladva másoljuk át a bájtokat egészen addig, amíg el nem értük az első sort is. Ha már az első sort is átmásoltuk a második sorba, akkor vehetjük elő a veremtárolóból az oda kimentett utolsó sor bájtjait és írjuk be az első sor bájtjai helyére, az ittenihez hasonlóan fordított sorrendben.

A mutatókat az SBC utasítás felhasználásával mozgatjuk visszafelé. A lefelé forgatás programját ezek alapján mindenki saját maga is kidolgozhatja.

A gépi kódú programot a SYS(52480)-nal hozhatjuk működésbe és a SYS(52505) utasítással állíthatjuk le. Itt is érvényes, amit az előző cikkünkben leírtunk, hogy a SYS(52480) utasítást nem szabad megismételni, mert ilyenkor a program állandóan saját magát fogja hívni. Leállítani már akár többször is lehet a programot.

Vegyük még szemügyre a gépi kódú program módosítási lehetőségeit:

1. POKE 52579,206. Ezzel az utasítással megváltoztathatjuk a \emptyset . szellemgrafika mozgatásának X irányát. Ez az utasítás ugyanis az 1. lista 803-as sorában lévő INC MOB \emptyset X-et változtatja meg DEC MOB \emptyset X-re. Az eredeti állapotot a PO-

```
10 REM *** SZABO JUDIT ILDIKO ***
11 REM *** SZELEMGRAFIKAI. ***
12 :
13 DATA 5,0,5,0,0,10,128,0,24,64,0
14 DATA 5,32,0,120,16,0,248,8,0,240
15 DATA 196,0,240,2,0,224,1,0,192,0
16 DATA 128,192,0,64,128,0,32,144,0
17 DATA 144,8,192,16,7,48,16,8,14,32
18 DATA 16,1,192,32,0,0,64,0,0,128,0
19 DATA 0
20 REM *** LOFEJ ADATOK BEOLVASASA:
21 REM *** EZ A 0-IK SZELEMGRAFIKA A
22 REM *** 11-IK SPRITE-TERULETEN.
23 FOR I=0TO62:READA:POKE704+I,A:NEXT
24 V=53248:POKE2040,11:POKEV+21,1
25 POKEV,255:POKEV+1,50
26 POKEV+23,1:POKEV+29,1
27 :
28 REM *** EGER SZELEMGRAFIKA ADATAI
29 DATA 0,0,0,0,99,0,0,148,128,1,8,64
30 DATA 1,6,64,0,144,64,0,80,128,0,241
31 DATA 0,1,6,0,2,2,0,66,130,0,62,2,0
32 DATA 32,2,0,24,5,192,31,8,0,40,128,0
33 DATA 72,128,0,136,64,0,0,64,0,0,64
34 DATA 0,0,0,0
35 REM *** EGER: 1-SO SZELEMGRAFIKA
36 REM *** A 13-IK SPRITE TERULETEN.
37 FOR I=0TO62:READA:POKE832+I,A:NEXT
38 V=53248:POKEV+23,3:POKEV+29,3
39 POKE2041,13:POKEV+21,PEEK(V+21)OR2
40 POKEV+2,255:POKEV+3,100
41 :
42 REM *** BERNATHEGYI ADATAI ***
43 DATA 0,0,0,0,0,15,128,0,48,112,0,64,8
44 DATA 0,128,4,0,128,4,1,4,18,127,36,18
45 DATA 192,4,17,128,4,17,128,4,16,128
46 DATA 4,16,128,4,16,128,196,16,71,130
47 DATA 32,63,193,193,24,96,6,0,48,24,0
48 DATA 16,96,0,25,128,0,14,0
49 REM *** BERNATHEGYI: 2-IK SPRITE,
50 REM *** A 15-IK SPRITE TERULETEN.
51 FOR I=0TO62:READA:POKE15*64+I,A:NEXT
52 V=53248:POKEV+23,7:POKEV+29,7
53 POKE2042,15:POKEV+21,PEEK(V+21)OR4
54 POKEV+4,255:POKEV+5,150
```

3. lista

KE 52579,238-cal állíthatjuk vissza. Figyelem: a POKE utasítású módosítások ideje alatt a programnak nem szabad működni, előzőleg a SYS(52505)-tel a programot le kell állítani!

2. POKE 52582,206. Az előzőhöz hasonlóan megváltoztatja a \emptyset . sprite Y irányú mozgásirányát. Az eredeti állapot: POKE 52582,238.

3. POKE 52580, 5 \times 2. Ennek az utasításnak a hatására nem a \emptyset . szellemgrafika

fog az X irányban mozogni, hanem az ötödik.

4. POKE 52583, 3 \times 2 + 1. Ez az utasítás az előzőhöz teljesen hasonló módon elveszi a \emptyset . sprite-tól az Y irányú mozgást és átadja a 3. sprite-nak.

5. FOR I=52579 TO 52584:POKE I,234:NEXT. Kiiktatja a szellemgrafika mozgatását, mivel az 1. listán a 803—805-ös sorokban lévő bájtokat a NOP-pal helyettesíti.

6. POKE 52588,64:POKE 52589,3. A 13. sprite-területre: $13 \times 64 = 3 \times 256 + 64$ helyezi át a forgatandó sprite-ot. Ezután már az a sprite forog majd, amelynek adatai ezen a memóriaterületen találhatók.

7. POKE 52539,X — ahol X a szellemgrafika sorainak számát jelenti: $1 \leq X \leq 20$. Ezzel az utasítással igen érdekes hatást érhetünk el: félbevághatjuk a szellemgrafikát: annak a felső X+1 sora forogni fog, míg a visszamaradó rész változatlan marad. Az eredeti állapotot a POKE 52539,20 állítja vissza.

8. FOR I=52480 TO 52519:POKE I,234:NEXT:POKE 52579,96. Ez az utasítás sor megszünteti a program kapcsolatát az IRQ rutinnal. A továbbiakban már nekünk kell a programot többször meghívni BASIC-ből ahhoz, hogy a szellemgrafika felfelé forgatását létrehozzuk. Ha 21 alkalommal adjuk ki a SYS52480 utasítást, akkor a szellemgrafika egyszer körbefordul. Így tetszőleges sebességű forgatást tudunk egy BASIC programból megvalósítani. Itt a szellemgrafika X és Y irányú mozgatása is kiiktatózott.

Végezetül pedig megadunk három szellemgrafikát a 3. listán. Ezeket az előbb elmondottak kipróbálhatók. Ezt a programot úgy írtuk meg, hogy az egyes sprite-okat külön is lehessen használni. Ha a 3. lista 13—26-os sorait gépeljük csak be, akkor a lófej szellemgrafika jelenik meg a képernyő jobb oldalán legfelül. A 28—40-es sorok az egér, a 42—54-es sorok a bernáthegyi kutya megjelenítésére szolgálnak. Ha valamelyik szellemgrafikát ki akarjuk kapcsolni, akkor a következőképpen járunk el:

POKE 53248 + 21,PEEK(53248 + 21)

AND(255 - 2↑I). Ez az utasítás kikapcsolja az I-edik sprite-ot.

Következő alkalommal a nagyfelbontású képernyő forgatásáról lesz szó. Ez azért nehezebb, mint a szellemgrafikák forgatása, mert a nagyfelbontású képernyő egyes soraiban lévő bájtok nem sorfolytonos számításúak, tehát nem olyanok, mint a szellemgrafika soraiban lévő bájtok (lásd az ábrát).

SZABÓ PÉTER PÁL—

SZABÓ JUDIT ILDIKÓ

Hardver

Az előzőekben röviden összefoglaltuk és bemutatuk a digitális technikában alkalmazott áramköröket. Ebben a részben az eddigi ismeretek felhasználásával megkezdjük a mikroprocesszoros rendszerek funkcionális egységeinek a bemutatását.

Mikroprocesszoros rendszerek áramköri felépítése

E rendszerek digitális áramkörökből épülnek fel, így a be- és kimenőjeleik is digitális jelek. Egy ilyen rendszer központi eleme a mikroprocesszor, amely utasításokat értelmez, és ezeket az utasításokat a memóriából, azaz a tárból olvassa ki. Kapcsolatát a külső környezettel a be- és kimeneti egység teszi lehetővé. Az így felépített rendszert mikroszámítógépeknek, röviden mikroprocesszoroknak nevezünk. A mikroprocesszor típusától függetlenül egy mikrogépnek a következő — az 1. ábrán bemutatott — főbb részei vannak:

- a központi egység (ez a mikroprocesszor), amelynek feladata a többi egység vezérlése és irányítása,
- a programtár és adattár (ez a memória), az egyikben a működtető program található, a má-

A sorozat alap gondolata — azon a régi felismerésen túl, hogy az elektronika és a számítástechnika elválaszthatatlan egymástól — a következő tapasztalatot summázza. A szoftver — a programok — jelentősége egyre nő, de az is tény, hogy az igazán jó (az adott számítógép nyújtotta lehetőségeket maximálisan kihasználó) programok megírásához a programozónak rendelkeznie kell alapfokú áramköri hardverismerettel is. Megerősíti ezt, hogy szaporodik az olyan berendezések, mikroprocesszort alkalmazó rendszerek száma, amelyek programvezérelten működnek. Az ilyen rendszerek tervezőinek és fejlesztőinek is szükségük van integrált hardver- és szoftverismeretekre.

sikban a program működése során létrejövő, vagy a külvilág felől érkező adatok tárolhatók, — a be- és kiviteli egység, amely a külvilággal tartja a kapcsolatot, — az egységek közötti információ- és adatáramlást szolgáló vonalak.

Ezek az egységek sok ezer tranzisztorból álló integrált áramkörök, amelyek vezetékekkel vannak összekötve, és a vezetékeken nagy sebességgel digitális jelek (H és L szintek) áramlanak.

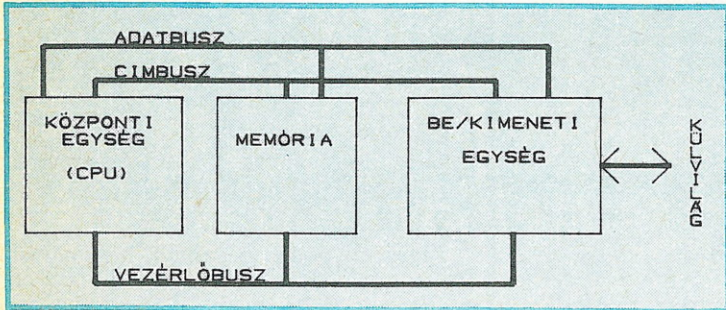
A mikroprocesszor és a funkcionális egységek elektromos úton kommunikálnak egymással. Ez a kommunikáció az úgynevezett sineszen vagy más néven buszvonalon jön létre. Mi is az a sín? A sín az azonos funkciójú vezetékek csoportja. Erre a vezetékkötegre minden egység párhuzamosan kapcsolódik azért, hogy egymáshoz információt továbbíthasson. Így többek között a mikroprocesszor, a memória, valamint a be- és kimeneti áramkörök, az egymás (és a külvilág) közötti adatcsere miatt ugyanarra a nyolc vezetékre kapcsolódik, ez az adatsín.

A kommunikációhoz az adatsín kívül még két sín: a cím- és a vezérlősin is nélkülözhetetlen.

A mikroprocesszor a címbuszon adja ki az információt aszerint, hogy melyik egységgel kíván kommunikálni, vagyis hogy melyik címről kíván olvasni, illetve, hogy melyik címre kíván információt küldeni.

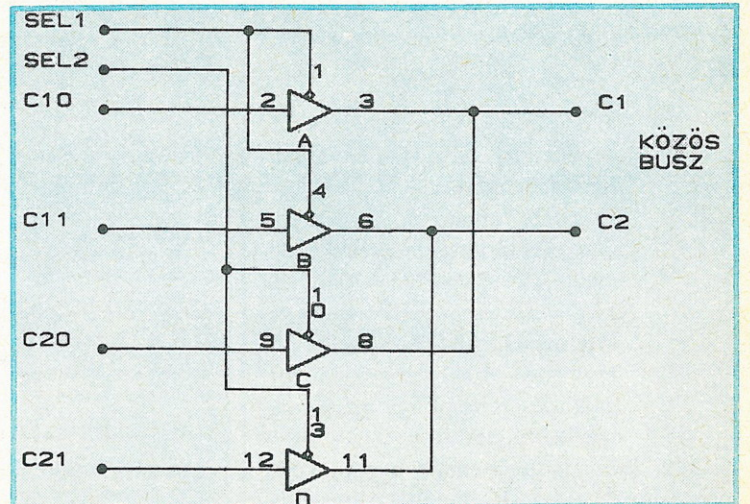
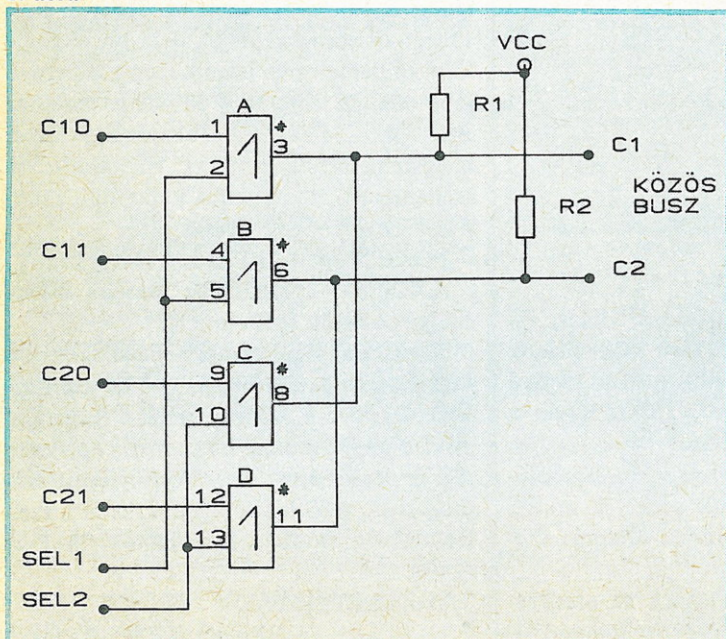
A vezérlőbusz vezetékein adja ki a mikroprocesszor az elvégzendő műveletre vonatkozó információt. (Például amikor egy megcímzett egységre adatot küld.) A legtöbb mikroprocesszor-típus azt is megkülönbözteti (egy másik vezérlőjellel), hogy a tárral vagy a be-kimeneti egységgel kíván-e kommunikálni.

A kommunikáció módjától függően a buszokon a jeláramlás lehet egyirányú, mint történetesen a címsínen, hiszen a címek a processzortól erednek, és a sín ezeket a memóriához, valamint a be- és kimeneti egységhez juttatja el. Lehet kétirányú is: az adatbuszon például az adatok mindkét irányba — a processzorból és a procesz-



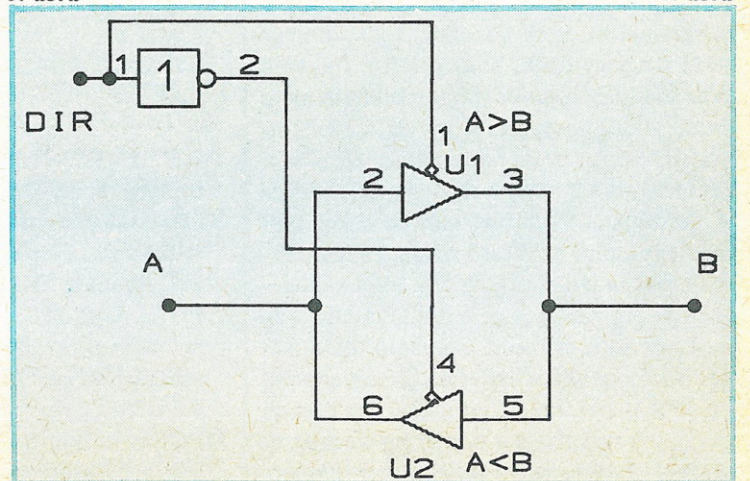
1. ábra

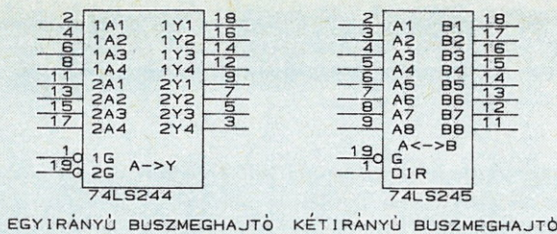
2. ábra



3. ábra

4. ábra





EGY IRÁNYÚ BUSZMEGHAJTÓ KÉT IRÁNYÚ BUSZMEGHAJTÓ

5. ábra

szorba — áramlanak. A sínekre csatlakozó egységek közül egy időben csak az ADÓ küld információt az adatsínre. A többi egység vagy nem vesz részt a működésben, vagy a kiküldött adatokat veszi, azaz vevőként működik.

Az egységek buszra kapcsolódó részét buszmeghajtóknak nevezzük. Az ilyen jellegű síncsatlakozás megoldható logikai kapuhálózattal is, de a gyakorlatban két megoldás, a nyitott kollektoros és a háromállapotú kimenetek használata terjedt el.

A nyitott kollektoros tranzisztorkimenetek — ahogy ezt már a sorozat 1. részében megemlítettük — lehetővé teszik, hogy a síncsatlakozásokat összekössük, és közösített pontként egy közös, úgynevezett felhúzó ellenállást alkalmazzunk. A közös ponton akkor lesz H szint, ha minden rákapcsolódó kimenet H szintű. Ha bármelyik kimenet L szintűvé válik, a közös pont is ilyen állapotú lesz. Ha úgy alakítjuk, hogy azok a buszra csatlakozó kimenetek, amelyeknek éppen nem szabad befolyásolniuk a buszt, H szinten legyenek, akkor a sínen adóként működő egység az L szintjeivel a közös szint L szintre képes állítani. A megoldást kétvezetékes, két egységet tartalmazó busz esetére a 2. ábrán mutatjuk be.

Az ábrán a „*” a nyitott kollektoros VAGY kapu kimeneteit jelöli. A SEL1 és SEL2 jelek L szintje választja ki, hogy melyik egység kapcsolódjon a közös buszra. Természetesen azt a vezérlő áramkörnek kell lehetővé tennie, hogy egyszerre a két egység ne legyen aktív (SEL1 és SEL2 egyidőben nem lehet L szintű).

Az igazi — és ma már általánosan használt — sínrendszerekben az úgynevezett háromállapotú (tri-state, ejtsd: triszét) kimenetű egységeket alkalmazzák. Az elnevezést a kimenet működési módja magyarázza: a háromállapotú kimenet tiltáskor a kimeneti pont nagy ellenállású (úgy mondjuk, hogy lebeg), olyan, mintha lekapcsoltuk volna a buszról. Ilyenkor egyáltalán nem befolyásolja a busz állapotát. Engedélyezve ugyanúgy működik, mint egy kétállapotú (H és L szintű) kimenet. Ez a megoldás TTL áramköröknel is megtalálható. Az ilyen jellegű busznál a nem aktív egységek kimeneteit tiltott állapotba kell állítani. A 3. ábrán az így kialakított sínmegoldást mutatjuk be. Az ilyen buszon csak egyirányú adatáramlás lehetséges, azaz a fenti esetekben az egységek csak ADÓ-ként kapcsolódhatnak a buszra. Mint az előbb már említettük, az adatvonalak kétirányú meghajtást igényelnek. Az ilyen megoldáshoz két egymással azonos, de ellentétes irányú szint alakítanak ki. A háromállapotú áramkörök felhasználásával azonban ez könnyebben megvalósítható. A 4. ábrán látható ennek egy buszvezetékes megoldása.

A DIR jelű bemenet logikai állapota határozza meg azt, hogy az adat A oldalról jut a B oldalra, vagy fordítva. Ha DIR=H, akkor az U1-es meghajtó engedélyezett (A>B irány), és U2 tiltott. A DIR jel L szintjénél a másik irány engedélyezett.

Az egy- és kétirányú buszmeghajtást minden buszra kapcsolódó áramkörben meg kell valósítani, hogy az egységek összekapcsolhatók legyenek. Mivel ezekben az áramkörökben az áramköri funkcionális működést megvalósító áramkörök is igen sok beintegrált tranzisztort használnak fel, ezért az áramkörök buszra kapcsolódó háromállapotú kivezetéseinek terhelhetősége kicsi, általában 1-2 TTL egységterhelés (fan in=2, fan out=2). Nagyobb terhelhetőségű kivezetések kialakítása nagyobb integrált áramköri

felületet igényelne, ami nem gazdaságos megoldás. Ezért nagyobb rendszerekben általában külön buszmeghajtó áramköröket használnak a processzor jeleinek meghajtására. Kisebb rendszerekben nem mindig szükséges külön buszmeghajtók használata, mivel ezt a feladatot a processzor kimeneti áramkörei is el tudják végezni. Az 5. ábrán konkrét példaként a két leggyakrabban használt buszmeghajtó áramkör funkcionális vázlatát mutatjuk be.

A G bemenet alacsony szintjénél engedélyezték a buszmeghajtók.

Tápfeszültségek a rendszerben

A mikrogepekben felhasznált TTL áramkörök miatt a mikroprocesszoros rendszerekben használt leggyakoribb tápfeszültségérték +5 V. Egyéb — a későbbiekben ismertetendő — egységek ezen kívül még használják a +12 V, -5 V és -12 V-os feszültségeket is. Egy tápfeszültség előállítására általában a 6. ábrán kialakított kap-

tápfeszültség változására, a feszültségstabilizátorral tesszük lehetővé, hogy a tápfeszültség a tápegység kimeneti terhelésétől függetlenül állandó legyen. Ezek a stabilizátorok fix feszültségre stabilizálnak, így a +5 V-os kimenő feszültséget adó leggyakrabban alkalmazott típus a 7805 típusjelű. A +12 V, -5 V, -12 V feszültségek előállítására rendre a 7812, 7905, 7912 típusjelű áramköröket használjuk. Ilyen formában a tápegységek felépítése egységes, csak a stabilizátorok típusában és negatív feszültségnél a feszültség polaritásában különböznek.

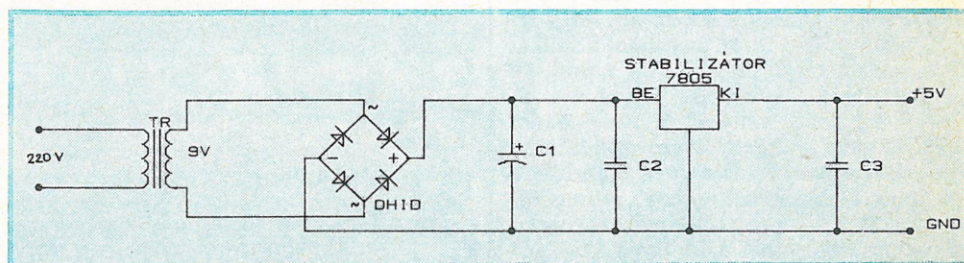
A rendszerekben nagyon fontos a jól és megbízhatóan működő tápegységek használata. A hálózat felől érkező elektromos zavarokat is a tápegységnek kell kiszűrnie. Erre a célra szolgálnak a C2, C3 kondenzátorok.

A mikrogepekben vannak belső zavarforrások is. Minden digitális áramkör jelszintváltása a tápfeszültségen zavarjelet hoz létre, és a mindenhová elvezetett tápfeszültség vezetékén ez a zavar szétterjed. Ezek ellen hatásos védelmet nyújtanak az integrált áramkörök tápfeszültség-csatlakozásai közelében elhelyezett szűrőkondenzátorok.

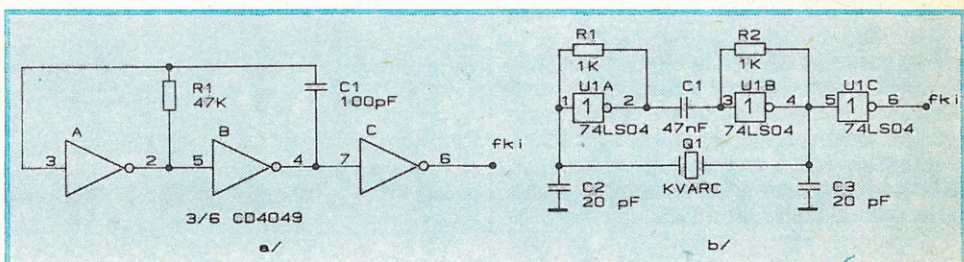
Mivel a tápvezetéseken a jeláramokhoz képest lényegesen nagyobb áramok folynak, ezért ezeket vastagabb (szélesebb) vezetékekből kell kialakítani.

A rendszer órajelének előállítása

A mikroprocesszoros rendszer szinkronműködésű logikai hálózatokból épül fel. Ehhez a rendszerben órajel generálnak. A nagy sebességű



6. ábra



7. ábra

csolást használják. Maga a tápegység lényegében négy jól elkülöníthető részre bontható:

1. A transzformátor feladata kettős: egyrészt a hálózati nagy váltakozó feszültséget kisebb feszültség szintre csökkenti, másrészt életvédelmi okokból a hálózati feszültséget a berendezéstől fémesen leválasztja.

2. Az egyenirányító a szinuszosan váltakozó feszültségből még hullámzó, de már egyenfeszültséget állít elő. Általában erre a célra nagy diódából kialakított úgynevezett hid-egyenirányítót használnak. Ez jelenleg egy tömbbe öntött, négykivezetésű egységként kapható.

3. A szűrőkör a hullámzó egyenfeszültség ingadozásának megszüntetésére szolgál. Erre a célra nagy kapacitású kondenzátorokat használnak.

4. Mivel a mikrogep áramkörei érzékenyek a

gű, pontos időzítésen alapuló működés tiszta, lengésektől mentes pontos órajelet igényel. Általában a kvarckristállyal épített órajel-generátorokat alkalmazzák. Ezek pontos és tiszta órajelet állítanak elő, de igénytelenebb esetekben RC oszcillátorokat is alkalmaznak.

Az órajel frekvenciája határozza meg a mikrogep művelet-végrehajtási sebességét, ezért értéke tetszőlegesen nem választható meg. A 7. ábrán egy RC és egy kvarcoszcillátor kapcsolást mutatunk be.

Mind a tápegység, mind az oszcillátor megfelelő funkcionálása elengedhetetlen a mikrogep működéséhez. Hibakereséskor is ezek azok az egységek, amelyek működéséről elsőként kell meggyőződni!

A CHIP nyugatnémet számítástechnikai magazin ez évi első számának 101. oldalán kellemes meglepetéssel találkozhattak az olvasók: ez egy hajlékonylemezzel vitt, Volkswriter de luxe névre hallgató szövegszerkesztő program volt. A címbeli közmondás intelmét felrúgva megnéztük, hogy mit tud ez a program. A képernyőképeket — bár a program németül „beszél” — a jobb érthetőség kedvéért magyar fordításban közöljük.

Mit várhatunk egy szövegszerkesztőtől?

A szövegszerkesztő program segítségével szövegeket hozhatunk létre.

Mit várunk még — a teljesség igénye nélkül — egy szövegszerkesztőtől?

— legyen a szöveg oldalakra, fejezetekre, bekezdésekre tagolható

— a készülő szövegben tudjunk előre, hátra mozogni, keresni, beszúrni

— „lapszintű” formai lehetőségeket: futócím, lánbjegyzet, több hasábra bontás, lapszámzás

— „bekezdés szintű” formai lehetőségeket: balra, jobbra ütköztetés, széttolás, középrehelyezés, fejezetszámzás, tabulálás.

Ennyit a legtöbb szövegszerkesztő tud. A különbség közöttük abban áll, hogy milyen könnyedén tudjuk használni a funkciókat, mennyit kell fejből tudni a program parancsairól.

Van azonban néhány sajátos feladat, amelyet nem mindegyik szövegszerkesztő tud végrehajtani. Például ilyen a szérialevelek írása. Ehhez két fájlt kell készítenünk: az egyikbe írjuk a szétküldendő szöveget, a másikban felsoroljuk a különböző megszólításokat. A két állományt a program összefűli és a levelet annyi példányban nyomtatja ki, ahány megszólítást megadtunk.

Az előbbieken leírt funkció neve többnyire MERGE, TEXT MERGE, esetleg PRINT MERGE.

Egy másik példa. Valamilyen termékhez használati utasítást kell írunk, és szeretnénk, ha bizonyos szakkifejezések vagy akár az egész szöveg egy adott szókészletből kerülne ki. A szöveg elkészülte után, esetleg közben a program előkeresi a szavakat a szótárból. Ha nem talál egy szót, akkor hasonlót keres, és kérhetjük például egy hibásan gépelt szó cseréjét is. A szótár

gyakran készen kapható, de lehet sajátot is gyártani.

És mit nyújt a Volkswriter de luxe?

A program menüvezérelt. Elindításkor egy kezelői menüt kapunk (1. ábra), és van egy segítő parancsmenü, amely bármikor előhívható. A parancsok kiadásakor az F1—F10 gombokat kell lenyomni egyedül vagy valamelyik shift típusú (ALT, SHIFT,

CONTROL) billentyűvel együtt (lásd a táblázatot).

Válasszuk a kezelői menü E parancsát! (Új fájl megnyitás.) A fájl nevének megadása után kapunk egy üres munkaterületet és kezdődhet a szöveg begépelése. Ha az A parancsot választjuk, a munkaterület nem üres, hanem a megadott nevű (létező) fájl töltődik bele.

A munkaterületen a nyilakkal lehet mozogni. A navigálást segítik a PgDn, PgUp, Home, End gombok, sőt az F3 lenyomására a kurzor az adott sor elejére (végére) ugrik. A tabulálás egyszerű: be kell hívni a ve-

Volkswriter Deluxe - parancsmenü			
E - új fájl megnyitás	V - tartalomjegyzék	D - nyomtatás	X - kilépés
A - fájl megnyitás	W - folytatjuk a munkát	T - textmerge	U - átnevezés
S - fájl kimentés	F - formátum beállítás	H - help	L - törlés

Tartalomjegyzék: E:\VX

Kérem válasszon :

E - egy új fájl megnyitása	A - létező fájl újramegyítése
W - folytatjuk a megnyitott fájl szerkesztését	S - a munkaállomány tárolása azonos vagy más néven
F - a margók, lapméret, betűtípus, sortávolság, végtelenített vagy egyenkénti papír beállítás	H - beépített írásos segédlet hívása a képernyőre
D - nyomtatási menü	V - tartalomjegyzék váltás vagy az aktuális tartalomjegyzék kiírás
T - textmerge, a címek összefűlése a szöveggel	
E - visszatérés a DOS-hoz (exit)	U - fájl átnevezés L - fájl kitörlés

Szerkesztés közben HELP: az F1 lenyomására

1. ábra

Formátum menü		
V - formátumhozzárendelés	A - másik formátum megnyitás	
Kérem válasszon:U	U - formátummódosítás	S - formátum kimentés

Aktuális formátum: E:VW	.FMT	Formátummódosítások
Nyomtató típusa (3=tetszőleges)	: 3	A kurzor a formátumparaméterek mellé áll.
Nyomtató újrabéállítás (I/N)	: N	Új paraméter megadása: írjuk be az új értéket és üssünk <ENTER>-t.
Lapszámzás (I/N)	: 1	<ENTER> leütése változatlanul hagyja az értéket.
Lapméret (2-255)	: 72	Ha megfelelnek az értékek: üssünk F10-et.
kezdősor	: 6	A formátumparaméterek lemezzre tárolhatók.
zárósor	: 66	
Bal margó	: 1	
Sortávolság	: 1	
Végtelenített papír (I/N)	: 1	
Kétszeres erősségű nyomtatás (I/N)	: N	
Rejtett szövegnyomtatás (I/N)	: N	
Blokkok nyomtatása (I/N)	: N	

2. ábra

A funkciógombok hatása

GOMB:	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Egyedül:	help	tárolás	<---->	-szó	kijelölés eleje	kijelölés vége	keresés	átformálás	margók	alap-állapot
ALT	+:nyomtatás	beszurás	+sor	törlés sorvégig	átvitel	másolás	helyettesítés	törlés	középre vitel	menü
CTRL	+: formázás	megjegyzés	jelek kiírása	-sor	blokk eleje	blokk vége	ugrás	-	-	-
SHIFT	+: alsó index	felső index	Shift1	Shift2	Shift3	Shift4	vastag betű	áthuzás	aláhuzás	rejtett szöveg

ne) nézd a fogát!

zetőlécet (F9) és ki kell jelölni a bal, jobb margót a /, \ karakterekkel. A sorban elhelyezhetünk tabulátorstop-pozíciókat is: jele a + karakter. Ha gépeléskor egy bekezdésen belül a jobb margót átlépjük, az éppen beírt szó egyszerűen átkerül a következő sor elejére. A kezelhető szöveg mérete 1 Mbájt lehet, ez kb. 500 gépelt A4-es formátumú oldalnak felel meg. Egy sorba legfeljebb 250 karaktert írhatunk; a képernyőn természetesen minden sorból 80 karakter látszik. ENTER leütésre új bekezdés kezdődik a sor elején. Címek, megszólítások középre helyezése az ALT/F9 hatására történik. Ne felejtünk el újra „centírozni”, ha utólag megváltoztattuk a cím hosszát, mert ezt a program nem végzi el automatikusan.

Beszúráshoz a programot insert üzemmódba állíthatjuk. Ilyenkor a kurzor alakja másrmilyen. Egy karakter törlése akár a DEL, akár a szürke ← gomb segítségével történhet. Ha a sor elején áll a kurzor, nem megy a további karakterenkénti törlés. Ilyenkor a CTRL/F4 segít: ez a sortörli parancs. Nagyobb összefüggő rész eltávolításához a kívánt szövegrészt a > és a < határolójelek közé kell venni, majd ALT/F8. Nincs azonban lehetőség egy törölt szövegrész visszaállítására.

Egy IBM PC klaviatúrán nincsenek ékezetes betűk. A szövegszerkesztők általában egyes ritkán használt billentyűket kódolnak át úgy, hogy lenyomásukkor ékezetes betűk jelenjenek meg a képernyőn. A VX

megoldása: az ALT gomb és egy betűbillentyű lenyomása ér egy ékezetes betűt. Hogy melyiket, azt egy táblázatból veszi a program. A táblázat egy szöveges formájú fájl, a neve KEYBD.INT. A fájl két sorból áll: az elsőben az ALT-tal lenyomandó gomb eredeti kódjának megfelelő karakter áll, a másodikban, pontosan alatta pedig a megjelenítendő ékezetes betű. A fájlt magunk is változtathatjuk, ha nem tartjuk megfelelőnek az ALT karakterek és az ékezetes betűk hozzárendelését. Hasonló a helyzet a nyomtatást vezérlő PRINT.INT fájl esetében is.

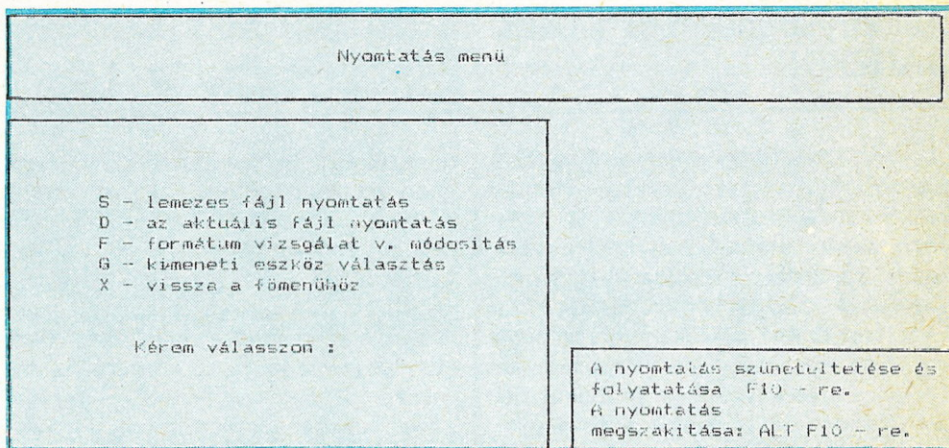
A kinyomtatott szövegre vonatkozhatnak formai szabályok. Például megadható a

nyomtatók többnyire a karakterformák széles választékát nyújtják. A VX alapcsomagban levő printervezérő (a neve VXPRTI.TBL) ezek közül közvetlenül csak a normál, aláhúzott, áthúzott és a kétszer nyomtatott karaktert ismeri. Ki lehet adni azonban a ..KMD parancsot: az utána írt karaktereket a VX printervezérő utasításként nyomtatja, azaz nem vált utána új sort.

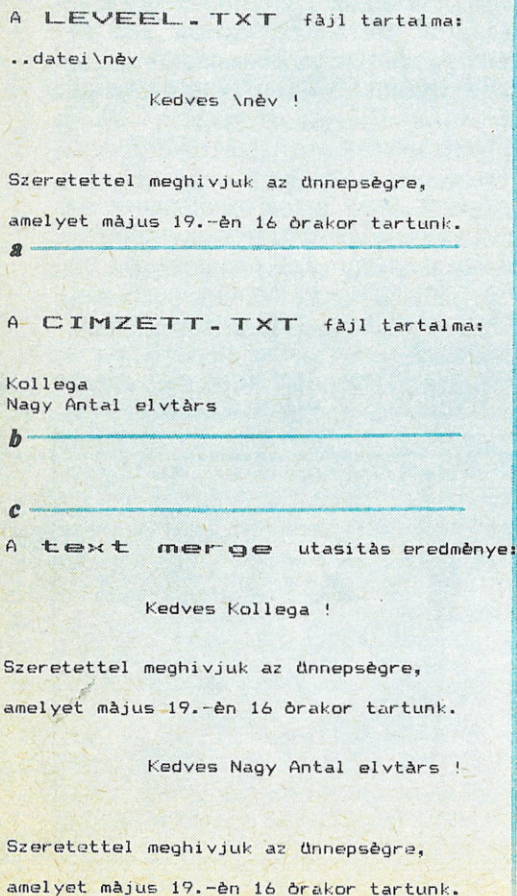
A VX végrehajtja a bevezetőben említett TEXT MERGE funkciót is. Nézzünk egy példát arra, hogyan kell használni. Az egyik fájlba írjuk le az elküldendő levél szövegét (3a. ábra).

A ..DATEI utasítás után álljanak azok a mezők, amelyekbe be kell helyettesíteni a paraméterfájl tartalmát. A paraméterfájl tartalma legyen a 3b. ábra szerinti. Ha ez is el van mentve lemezre, kezdődhet a nyomtatás.

Válasszuk a kezelői menü T parancsát. Erre a VX megkérdezi a szereplő fájlok nevét. A behelyettesítés így zajlik le: a para-



3. ábra



4. ábra

lapméret, ezen belül az is, hány sor maradjon üresen a lap tetején és az alján, mennyi legyen a sorköz stb. Ezeket a szabályokat a VX egy formátumfájlban tárolja, binárisan kódolva, azaz csak a VX-en keresztül módosíthatjuk, a kezelői menü F parancsa segítségével (2. ábra). Egy munkaállományhoz hozzárendelhetjük bármelyik létező formátumfájlt. Ha nem tesszük meg, a program bizonyos fájlnev és kiterjesztés azonosságok alapján elvégzi helyettünk.

A VX másik jellegzetessége, hogy a kinyomtatandó szövegbe elhelyezhetünk néhány — a szövegben változást okozó — utasítást. Ezeket kötelező az 1. pozíción kezdeni, két ponttal. Például:

..KOPF02zBevezetés

E parancs jelentése: mostantól minden lap második sorában (02) legyen fölírva a „Bevezetés” szöveg, még hozzá középre (z). A szabályos utasításokat a VX nem nyomtatja ki, hanem végrehajtja, kivéve, ha egy .. VERB utasítást talál, mert ez az utasítás azt jelenti, hogy mostantól kezdve mindent — még a két ponttal kezdődő sorokat is — ki kell nyomtatni. Ellentéte a ..NORM utasítás. Egy másik példa:

..PAUSE—új_lap_következik—

Hatása: a nyomtatás félbeszakad és addig szünetel, amíg egy tetszőleges billentyűt le nem nyomunk. Közben a képernyő utolsó sorában megjelenik az

—új_lap_következik—

szöveg.

Az IBM PC gépekhez csatlakoztatható

méterfájl egy sorában egymástól vesszővel elválasztott mezőértékek vannak, ezeket a nekik megfelelő — a levélben szereplő — mezőnév helyett írja ki a program. A paraméterfájl minden sorához egy teljes levél készül.

Az eredmény a 3c. ábrán látható.

A VX tartalmaz néhány, nyomtatással kapcsolatos további lehetőséget, amelyekkel a munkánkat segíti: a munkaállomány részeit is ki lehet nyomtatni, a szövegben el lehet helyezni „árnyékban levő” karakteret, amelyekre a program nem nyomtat ki. A kezelői menü D parancsára a 4. ábrán bemutatott képet láthatjuk.

Nyomtatás előtt választhatunk, hogy képernyőn, nyomtatón vagy lemezes fájlban kérjük-e az eredményt. Lemezre akkor nyomtatunk, amikor a szerkesztést nyomtató nélküli gépen végeztük. A lemezre írt állományt egy másik — nyomtatóval ellátott — gépen VX nélkül is kirathatjuk, például a DOS PRINT parancssal.

A kézikönyvből bizonyára sokkal többet megtudhatunk a VX-ről. Annyit azonban megállapíthatunk, hogy a program a többi IBM PC gépen futó szövegszerkesztőhöz viszonyítva mind a kezelés, mind a tudás szempontjából a középmezőnyben helyezkedik el.

A PRINTFOX ÉS A PAGEFOX

Desktop Publishing a Commodore 64-en

Az NSZK-beli Zorneding városkában működő SCANNTRONIK vállalat Desktop Publishing (DTP) programjaival is beírta magát a Commodore 64-esek történelmébe. Ezek a szoftverek a hazánkban már sokak által ismert PRINTFOX és legújabb testvére a néhány hónap óta modul formájában forgalmazott PAGEFOX.

Az utóbbi időben egyre gyakrabban hallhatunk a Desktop Publishing-ről illetve rövidítve a DTP-ről, amely mindenkinek elérhetővé teszi a különböző kiadványok, nyomtatványok saját tervezését és készítését. Mit is takar voltaképpen ez a nemzetközivé vált amerikai eredetű kifejezés? A szó szerinti fordításban írásztali kiadót jelentő szoftvercsomagok különböző számítógépekre készültek. Ezek lehetővé teszik alkalmazóknak, hogy egy szoba sarkában szövegek, címek, képek és grafikai elemek felhasználásával sajátmaguk megtervezhessenek és megjeleníthessenek különböző kiadványokat, nyomtatványokat a sokszorosítás céljára. Hogy mik az alapvető követelményei a DTP programoknak? Tervezés szempontjából elsősorban a szöveghasások tetszőlegesen választható száma és nagysága. A sorkizárt, vagyis a hasábon belül az egyenlő

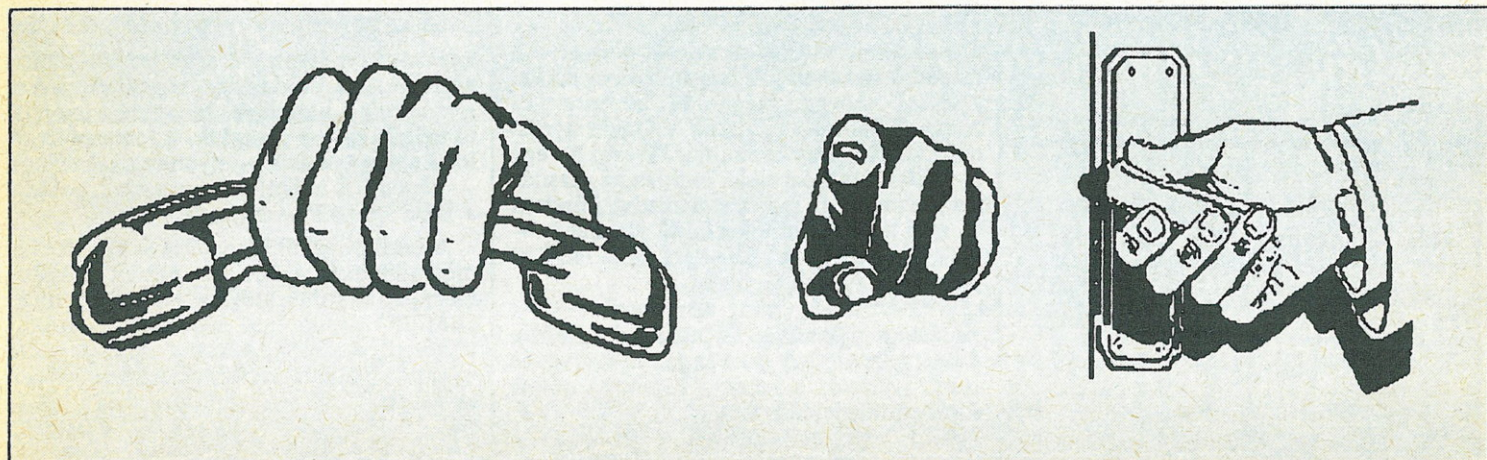
hosszúságú sorok biztosítása is nagyon hasznos dolog. Fontos funkció a képek, grafikák beillesztési lehetősége, illetve méretük változtathatósága. A pontos tervezést segítheti az elektronikus „tervezőpapírra” hívható pontháló, vagy raszter, valamint a koordináta kijelzés. A DTP programok általában saját szövegszerkesztővel is rendelkeznek, de fel kell tudniuk használni más szövegszerkesztők által létrehozott szövegfájlokat is. Mind ebből akkor készíthető változatos oldal, ha különböző betűcsaládok és betűváltozatok is rendelkezésre állnak. Jó ha van egy „beépített” rajzoló program is készletben. Ma már ezeknél a szoftvercsomagoknál ugyancsak alapvető követelmény, hogy minden ami majdan kinyomtatásra kerül, az a tervezés fázisában ugyanolyan formában a képernyőn is látható, ellenőrizhető legyen. Ez az úgynevezett WYSIWYG

elv, ami annak az angol kifejezésnek a kezdőbetűiből alakult ami magyarul úgy hangzik, hogy „amit látsz (a képernyőn), azt kapod (eredetiként)”.

A PRINTFOX és a PAGEFOX eleget tesz a DTP programok alapvető követelményeinek, sőt az utóbbinak némely funkciója még a professzionális nagyobb gépek hasonló szoftverlehetőségeit is túlszárnyalja. Mindez Hans Haberlnak a munkáját dicséri. És, hogy milyen eredményt kapunk végtermékként például egy Printfox-tól, azt ez a két oldal is tükrözi, amelyet most tart kezében az olvasó. (Ez az oldalpár olló és ragasztó nélkül, 1:1 méretben egy CITIZEN 120D típusú nyomtatóval készült.)

Az első FOX

A Printfox nemcsak egy minden igényt kielégítő kitűnő szövegszerkesztőt tartalmaz, hanem egy Hi-Eddi alapú, ugyancsak kiváló rajzolóprogramot is. A négyképernyős grafikus szerkesztőbe, - amely egyúttal a megjelenítendő oldal tervezési felülete is - behívhatók a Superscanner II-vel leolvasott vonalas képek mellett más grafikai programokkal készített képeink is. Így például az Art Studio, a Doodle és a Profi Painter rajzai is használhatók.



Néhány rajz a PRINTFOX BASAR több mint kétszáz kész, azonnal használható grafikájából

A kiegészítésként kapható PRINTFOX BASAR három lemezoldali tömörített feldolgozású kész ábraanyagot tartalmaz. Az összeállítás a legkülönbözőbb területekről ad válogatást. A másik kiegészítő programcsomag a CHARACTER FOX, amely az alap betűkészletet újabb húsz típussal, díszkeretekkel, iniciálékkal egészíti ki. Több hasznos programocskát mellé szintén ezen a lemezen található egy karakter-szerkesztő. Ezzel módosíthatjuk meglévő betűkészleteinket (pl. magyar ékezetesre!, kontúrosra, árnyékoltra stb.), vagy tetszőleges új betűcsaládokat tervezhetünk, de a grafikai szerkesztőbe behívott bármilyen betűállomány átmásolható és a továbbiakban saját karakterként felhasználható.

A szöveghasábok elhelyezését, betűtípusait, a sorok előre, hátra zárását, a betűk követési távolságát, vagy a sorkizárást a szövegszerkesztőbe beírt utasítással határozza meg. Kicsit nehézkes a koordináták pontos beállítása, a különböző betűtípusok behívása (akár soronként is) lelassítja a munkát. Az egyenletes szövegtextúra kialakítása érdekében ajánlatos a szavak sorvégi elválasztása, ami szintén megoldott, de az elválasztási javaslatokat a Printfoxnál magunknak kell elhelyezni.

A PAGEFOX az igazi...

A Scantronik fejlesztőinek határokat szabott a C 64 memóriája, a DTP programjuk új változatát a PAGEFOXOT már nem tudták „belegyömöszölni”. Megtalálták a megoldást és létrehoztak egy modult, amit csak a bővítő csatlakozóhelyre kell dugaszolni és ezzel egy 160 Kilobájtos számítógéppé alakul 64-esünk. A modul magát a programot is tartalmazza és ez már a Printfox összes kényelmetlenségét kiküszöböli. Így amellet, hogy kompatibilis a kistestvérrel, használata leegyszerűsödött, gyorsabb lett és újabb professzionális funkciókat kapott.

A szövegszerkesztő maga meg egyezik a Printfoxéval, megtartva annak az összes jó tulajdonságát. A más programokból származó adatokat automatikusan saját formátumára konvertálja. A szöveg formázását jól áttekinthető, egérrel, vagy botkormánnyal kezelhető menü könnyíti. Írásunk az 1, 2 és 3 hasábos tördelésen kívül a kiválasztott betűváltozattal tetszőleges „gumi keretbe” tölthető. A megfelelő elhelyezésről szabadon választható nullaponttal pixel, vagy milli-

A Pagefox lehetővé teszi azt, hogy rajzaink köré szedjünk szöveget. Ezt úgynevezett kontúrszedésnek hívják, amit még a nagyobb gépek is csak ritkán biztosítanak. A program ezt az utasítást, akárcsak a szövegválasztást automatikusan végzi.



AaBbCcDd 12
 AaBbCcDd 12
 AaBbCcDd 12
 AaBbCcDd 12
 AaBbCcDd 12
 AaBbCcDd 12
 AaBbCcDd 12
 AaBbCcDd 12
 AaBbCcDd 12
 AaBbCcDd 12
 AaBbCcDd 12

Betűminták a Karakter Fox kínálatából

méter koordináta-kijelző gondoskodik. Amennyiben az első keret „telefolyik” szöveggel, úgy a maradék automatikusan a következőkbe töltődik. Itt olyan új funkcióval találkozhatunk, ami még a nagy PC-k DTP programjaiban is csak ritkán van meg. Ez a kontúrszedés, ami azt jelenti, hogy egy grafika, vagy tetszőleges szabálytalan forma mellé is szedhetjük a szöveget. További újdonság az automatikus - sajnos német nyelvtan szerinti - szóelválasztás. A modulban lévő 12 betűtípust mintegy 3000 változattá bővíti a kombinációk lehetősége. A további betűcsaládok utántölthetők.

Jelentősen felgyorsult a kész oldalkép kialakítása is annak következtében, hogy a betűkészleteket nem kell a programnak állandóan utánatölteni és az egész A/4-es oldal is állandóan a memóriában van. Az ellenőrzést segíti a gyorsformázás amikor az oldal kicsinyített, olvashatatlan, de pontos képe öt(!) másodperc alatt jelentkezik. A végleges oldal megjelenítése a betűkészletektől és a szövegmennyiségtől függően 30-60 másodpercet vesz igénybe.

A Pagefox grafikai szerkesztője is több újdonsággal szolgál. Itt is megjelent a menülista. A 640 * 800 pontból álló teljes A/4 oldal folyamatosan készületben van. A geometriai formák kialakítását a „gumivonalak” segítik. Különlegesség, hogy a nyomtatónak megfelelően beállítható a pontos körrajzolás. A változtatható mintaállománnyal nemcsak a zárt formákat lehet kitölteni, hanem a különböző rajzolófunkciókban is használhatók. A kiválasztott másolandó rajzrészlet mozgatható közben is látható, nagy segítséget adva a pontos munkához. A rajzfelület áttekinthetőségét a 25 és az 50%-os kicsinyítés biztosítja. Az is újdonság, hogy az oldalról tetszőleges részlet lemezre menthető, behívható és ki-nyomtatható.

Aki kipróbálta a PAGEFOX-ot, bátran állíthatja, hogy az a C 64-re kapható Desktop Publishing szoftvercsomagok között nem talál versenytársra. (A gyártók egyébként további olyan programokat terveznek amelyek kihasználják a megnövelt tárkapacitást.) A Scantronik Printfoxa és Pagefoxa ideális eszköz a különböző iskolai, egyesületi újságok, meghívók, diplomadolgozatok és más nyomtatványok esztétikus, gyors házi előállítására.

Huppán Béla

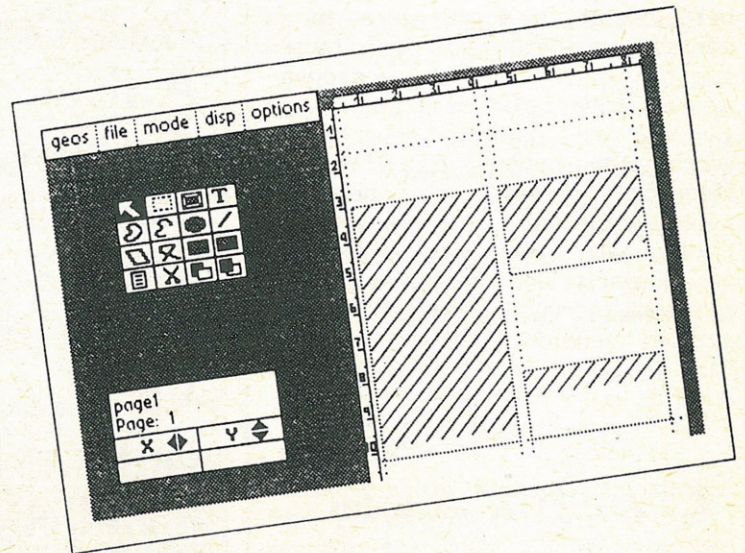
Geopublish, avagy kiadvány készítése az új Geos programmal

A C64-gyel szállított rajzoló- és szövegszerkesztő programot tartalmazó Geos felhasználói szoftvercsomaghoz az utóbbi időben több hasznos kiegészítő, illetve önálló program jelent meg. Ezek mind a már sokak által ismert, igen komfortos, felhasználóbarát tulajdonságokkal rendelkeznek. Az egyik legújabb ilyen program a Geopublish, amely Geos-környezetben teszi lehetővé az elektronikus asztali kiadványszerkesztést, a Desktop Publishinget. (A DTP-ről az előző oldalakon már írtuk.) A Geopublish nehéz közvetlenül összehasonlítani a PAGEFOX-szal, mert alapvetően mások a sajátosságai.

Először is arra szeretném felhívni a figyelmet, hogy aki a Geopublishtól gyors munkát vár, az szerezzen be egy RAM-bővítőt. Ugyanis nagyon hosszú maga a főprogram, tehát a töltés ideje, de használat közben is az egyes programrészeket újból kell behívni. A gyakori lemezműveletek nagyon lelassítják a munkát.

A programindítás után megjelenik a képernyőn (képünk) a bal oldalon az ismert legördülő menüablakok alatt egy tizenhat piktoqramból álló eszköztár, amelyből szükség szerint választhatjuk ki a megfelelő „szerszámot”. A koordinátakijelző sem hiányzik, és az oldalszámjelző már sejteti, hogy több oldalt is feldolgozhatunk egyszerre. A bejelentkező képernyő jobb oldalán a tervezendő A/4-es oldal kicsinyített képe látható. Ezen lehet meghatározni az oldaltükröt, vagyis azokat a segédvonalakat, illetve hasábokat, amelyekbe majd a szöveg, a képek és a címek kerülnek. Az A/4-es oldalt célszerű csak két szöveghasábra osztani, mert nincs szóelválasztási lehetőség. Több, vagyis keskenyebb hasábnál — tekintettel arra, hogy csak teljes szó lehet egy sorban — és ha a szavakból egyenlő hosszú sorokat képezve sorkizárt, azaz tömbhöz hasonló szöveget akarunk kapni, akkor nagyon egyenetlenekké válnak a szóközök. Egérrel vagy botkormánnyal válthatunk az úgynevezett layout (tervező) oldalra és kijelölhetjük a hasábokon belül a képek és a szöveg helyét. Sajnos, csak a Geopainttel előállított képeket tervezhetjük be. Akinek rendelkezésére áll a Desktop 1 kiegészítés, annak lehetősége van még az ott található „Graphics Grabber”-re a Print Shopból vagy a Print Masterből képeket konvertálni Geos formátumra. Mindenesetre a Geopublish képfeldolgozó megoldásának az az előnye megvan, hogy a Geos fotóalbumából kiválasztott kép eredeti méretét automatikusan az oldalban kijelölt hely méretéhez igazítja. Az aránytorzulást viszont nem lehet kikerülni. Az átvett képet aztán önálló objektumként lehet kicsinyíteni és nagyítani, illetve a kívánt helyre tolni. Csupán rá kell kattintani a képre, és máris aktívá válik a két kis négyzet által behatárolt terület. Egyszerűbb grafikákat, javításokat az eszköztárból kiválasztott rajzolófunkciókkal készíthetünk.

A címbetűk kiválasztásához rendelkezésre áll a Geos minden



eddig betűtípusa. A Geopublish erőssége, hogy a méret itt is változtatható. Ha a menüből a szövegbeírást választjuk, megjelenik egy külön ablak, ahol a betűnagyságot lépcsőzetesen akár néhány centiméteres méretig növelhetjük. A beírt címet aztán önállóan mozgathatjuk, a képekhez hasonlóan, a végleges helyére. Ha elhelyeztük a címet és a képeket az oldalon, következhet az előkészített helyekre a szöveg behívása. Ezt a szöveget előzőleg a Geos Geowrite szövegszerkesztőjével vagy a Writer's Workshoppal kell megírni. Az utóbbival azért jobb, mert azzal ellenőrizhetjük a sorkizárt szöveget is. Ha a folyó szöveg átkerült a Geopublish oldalakra, már csak kisebb javításokra van lehetőség a kinagyított olvasható területen. A megfelelő betűtípusoknak és természetesen a szövegnek is a Geopublish munkalemezen kell lenniük.

A nyomtatáshoz szintén rendelkezésre kell állni a megfelelő adatoknak. Az összeállított oldalakat a Geos által támogatott nyomtatókkal lehet megjeleníteni.

A Geopublish mint DTP-program segíthet abban, hogy akár nagyobb terjedelmű kiadványokat is gond nélkül készíthessünk. Ami a program jó tulajdonságait negatívan befolyásolja, az a lassú munka, ha RAM-bővítő nélkül kell dolgozni. A másik hátrány az, hogy az ismert és jó rajzolóprogramok által készített képeket nem lehet felhasználni. (Lapzárta után kaptuk az információt, hogy létezik már ilyen képátalakító programocská. Sőt, időközben az NSZk-beli Markt und Technik kiadó forgalomba hozta a Giga Paint nevű, új szoftverét, amely már tud olyan képeket produkálni, amelyeket a Geopublish is elfogad.)

há—bé—

A CENTRUM ÁRUHÁZAK Vállalat egy évvel e cikk megírása előtt jelentette be: boltjaiban forgalomba hozza az Enterprise 128 számítógépet. Azóta lapunk többször foglalkozott a hazai piacra hirtelenjében betoppant géppel. Termékismertetőt közölt például a 87/9. és a 87/11. számban. Az egy esztendő alatt jó néhány Enterprise-os olvasónktól kaptunk e gépről észrevételeit, információt. Most megpróbáljuk összefoglalni szerkesztőségünk és kedves olvasóink tapasztalatait e kissé sokat vitatott gépről.

A kezdet...

A NOVOTRADE Rt. és a CENTRUM ÁRUHÁZAK Vállalat jól szervezett reklámhadjáratával vezette be az új gép honi értékesítését. Az olcsó számítógépre „kiéhezett” tábor érdeklődése szinte határtalan volt, ami a mai napig sem csökkent. Ennek köszönhetően a magazinban megjelent felhívásnak is: „Szívesen fogadnánk a géppel kapcsolatos további szakmai információkat”, nagy fogantja volt. Nézzük hát, mi fán terem az a tapasztalat! — Az újdonsült tulajdonosok már a gép beszerzésekor sajtószerű helyzetbe kerülnek, amolyan beugratósdíba. Ugyanis két alapvetően (ezeken belül több eltérés létezik) különböző gépet hoztak forgalomba. Hogy ez baj avagy sem, nem a szerkesztőség feladata elbírálni. Az azonban, hogy erről a kedves vásárló a Felhasználói kézikönyv — amelyet a megvásárlás után kapunk kézhez — tizenharmadik oldalának tájékoztatásán kívül mit sem tudott, a kevésbé alapos pedig még most sem, már kereskedelemetikai kérdés. Amint az is, hogy a „programkazettát nem cserélünk” elv akkor is érvényesül, amikor a felhasználó teljesen tájékozatlan. Meggyőződésünk, hogy ha egy programkazettát — nem is olcsón — Enterprise 128 gépre eladnak, annak működni kell! Kivéve, ha az eladó felhívta a vásárló figyelmét, hogy az azonos nevű és típusú, de egyébként csak egy pontosan meghatározott gépen működik.

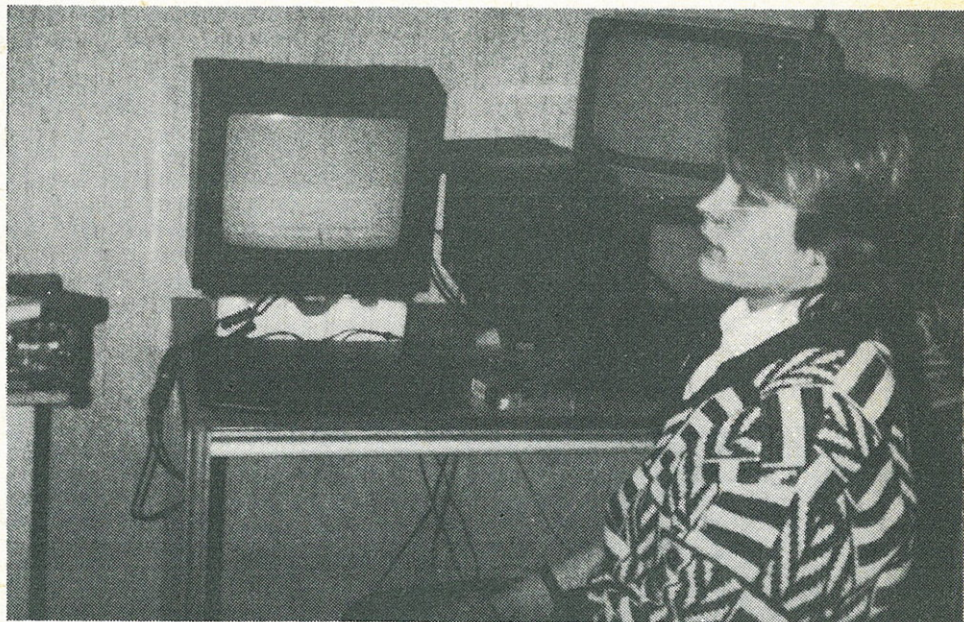
Félreértés ne essék! Nem az Enterprise gép ellen lépünk fel, hanem annak sokszor „homályos” értékesítését bíráljuk. A cikk írója nem tud azonosulni az „Ismerkedés az Enterprise számítógéppel” c. könyv szerzőinek azon ajánlásával, hogy „programszerző útjainkra legalább gépünk külső tárát vigyük magunkkal”. (13. old.)

Az Enterprise-zal hosszú távon számolhatunk

A szerzők (Benedikti István és Huszka Béla) jóindulatú és gyakorlatias tanácsa szerint, ha ne adj isten gyári videokazettát akarunk vásárolni, érdemes magunkkal vinnünk a készülékünket, de legalábbis a televíziókat. Igaz, más vélemény is van, amely úgy tartja, a felhasználóknak ilyen bonyolult gép esetében már illik felniőttké válniuk. Ma ugyanis az már senkit nem lep meg, hogy a PC-ken nem minden gyári szoftver fut, ezt a vásárlásnál kell tisztázni!

gép felépítéséből adódik. Az EXDOS felhasználói kézikönyv a lemez meghajtó egységek tizenhét típusát sorolja fel, amelyekkel tesztelték a gépet, azaz amikkel működik. Csak egy dolog hiányzott! Az interfész hardver és szoftver kiegészítése.

A vásárlók többsége nem barkácsoló típus. Aki azonban rászánta magát a különböző alkatrészek beszerzésére és még nyomtatott áramkört is csinált, beforrasztotta az alkatrészeket, hozzájuthatott az



Új szoftver készül az „A” Stúdióban

Más kérdés az, hogy az ilyen programok leírásában feltüntetik a lehetséges géptípusokat, illetve a kívánalmakat.

Ígéret és valóság

A beharangozás szerint az új géphez folyamatosan gyári programokat fejlesztenek ki, de ha ez késne is, a Spectrum-emulátor segítségével a választék szinte korlátlan. A perifériákról csak annyit, hogy — hála a fejlesztők előrelátásának — majdnem minden szabványos eszköz felhasználható. Ezek egyébként igen praktikus és könnyen kezelhető módon csatolhatók, ami a

„ígéret földjéhez”, és az egyik legmodernebb periféria kezelési rendszert használhatja. Vagy másik megoldásként megvehetjük azt egy, az igényt és a hiányt jól kihasználó gmk-tól. Amennyiben ezt a kis momentumot is figyelembe vesszük, úgy a kedves felhasználó szinkronban lehetett az idővel és aszinkronban a pénztárcájával. Sőt még gépe garanciáját is elveszítette.

Az ígéretetek megvalósulnak?

A Spectrum-emulátor valóban létezett,

igaz, csak prototípusban. A sorozatgyártás alkatrészhiány miatt nem indult meg. A többször áttervezett emulátorhoz olyan alkatrészek kellettek, amelyek vagy CO-COM-listán voltak, vagy beszerzésüket a magyar kereskedelem több mint egyéves határidővel vállalta. Közben elterjedt a hír, hogy nem is oldható meg a Spectrum-emulálás. Ebből annyi az igazság, hogy csak szoftverrel valóban nem. Az emulátor tervezője, az „A” Stúdió megmutatta az elkészült konstrukciót. Meggyőződhettem arról, hogy valóban működik. Ötszáz Spectrum programon próbálták ki, amelyeknek 85—90%-a futott. Nem tudja jelenleg kezelni a flash és a különleges turbo loadert.

Szintén ez alkalommal láthattam a sokak által hiányolt lemezmeghajtó illesztőt, a kontrollert. Mit mondanak? A „félmaszek” és a gyári között nemcsak árban, hanem minőségben is óriási a különbség. Szintén elterjedt rémhír, hogy a géphez nem lesznek további programok. Ebből az az igazság, hogy külföldről már valóban nem várhatóak, de csak az „A” Stúdió jelenleg nyolc szoftver fejlesztésén dolgozik. (Valószínűleg a cikk megjelenésekor már több kapható is.) A lemezmeghajtóhoz kapcsolt controller és az IS-DOS operációs rendszer lehetővé teszi a CP/M operációs rendszerek többségének a használatát, amiről szintén meggyőződhettem.



Enterprise teljes konfigurációban

Valóság és távlatok

A levélírók egyértelműen dicsérték az Enterprise 128 számítógépet, de elmarasztalták a hozzá kapható szegényes irodalmat. A felhasználói kézikönyvön kívül eddig megjelent az EXOS 2.1, az EXDOS 1.3 és az IS-DOS 1.0, valamint a már említett „Ismerkedés az Enterprise számítógéppel” és „A hetedhét Enterprise” című könyv.

Akik lemaradtak valamelyiknek az első kiadásáról, azoknak jó hír, hogy hamarosan megjelenik a második kiadás is. Remélhetően most már elég példányszámban. Ezenkívül szerkesztés alatt van több más kiadvány is. Mivel a könyvek megjelenését eddigi tapasztalatunk alapján nem követi különösebb hírverés, ezért a felhasználóknak csak azt tudjuk ajánlani, hogy gyakran érdeklődjenek a CENTRUM műszaki osztályain.

Korábban gond volt a NYÁK-csatlakozóval, de ez a múlté. Igaz, a gyártó KON-TAKTA hőre lágyuló műanyagból készíti a foglalatot, ami forrasztáskor sok kellemetlenséget okozhat. Végre-valahára most már az összes perifériakábel összeállítható. Kapható a külső botkormányhoz interfész, illetve adapter, így csupán a vevő ízlésén múlik, hogy milyen típust választ.

A VIDEOTON Elektronikai Leányvállalata a tavaszi BNV idejére ígéri az emulátort. Reméljük, betartja szavát. (Az olvasó a cikk megjelenésekor már tudja.) Jelenleg is több típusú, megfelelő lemezmeghajtó kapható, amely az „interfészkérdés” megoldódása után jól használható. Mi lesz a szoftverrel? Az éppen vásárolható külföldi programok utánpótlása nem várható. Folyamatos azonban a magyar fejlesztés. Aki komolyabban bele akarja magát vetni a számítástechnikába, annak szinte korlátlan távlatokat nyújt a CP/M rendszer alatt futó programok választéka. Nem tartjuk és tarthatjuk lebecsülendőnek a már több mint tízezer Enterprise-tulajdonos programkészítő hajlamát sem.

Néhány tanács

A gép billentyűzete egyszerű konstrukció, de azt tudja, amit ígér, viszonylag kitartó. Eddig intenzív használatnál sem találkoztunk fóliahibával. Azonban a koszt nem szereti. Ezért használaton kívül célszerű letakarni a gépet és a billentyűzetet alkalmanként tiszta szilikonolajjal lefújni. Ha a gépünk már nem garanciális, akkor nyugodtan leszedhetjük a fedőlappját, és a billentyűzet alatt levő gumit kitisztíthatjuk. A NYÁK-csatlakozó megvezetése a gép műanyag dobozán van, és ez nem minden esetben találkozik a dugón lévő horonnyal. Így a periféria interfésze nem érintkezik tökéletesen. Előfordult, hogy csak többszöri próbálkozásra sikerült a kellő érintkeztetés. A gyári programokat szövegszerkesztő üzemmódban érdemes betölteni, mert majdnem mindegyik gépi kódban készült. Így a legtöbb működik a tesztelések szerint. A SNOOKER (biliárd) program nem fut a kétnyelvű gépen. A hiba oka a túl kicsi ALLOCATE terület, mert a német BASIC is helyet foglal. Kristóf Attila olvasónk küldött erre megoldást. Mivel a 20-as sorban a betöltést csak az első 9 bájtt végzi, a többi nyugodtan kitörölhetjük, legfeljebb nem lesz megszakításletiltás. Az utolsó számot, a C9-et a BASIC-visszatérés miatt meg kell hagyni. A 10-es sor-

ban a 11570-t írjuk át 11555-re. Az átírt program a következő:

```
10 ALLOCATE 11555
20 CODE LOADER=HEX $ ("3e, 8, e,
0, 59, 55, 44, f7, 6, C9")
```

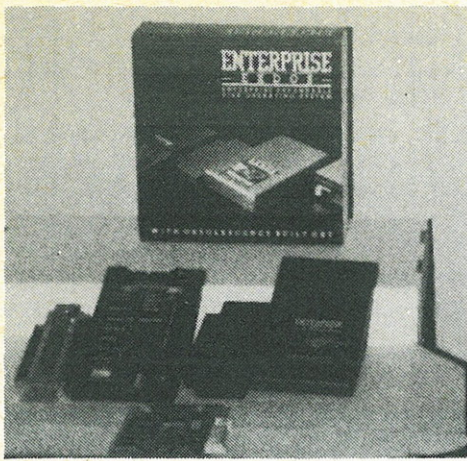
Elindulás előtt vegyük fel a programot.

Néhány jó tanácsot adott Szabó István a programok kazettán való tárolásáról. A kazettára mentés előtt nem kell megnyomnunk a SHIFT+F4 billentyűket. Ha ezt tesszük, akkor nekünk kell a magnót kezelni, míg alaphelyzetben a távvezérlés be van kapcsolva. A SAVE kiadása után — ellentétben a C64-gyel — nem vizsgálja, hogy lenyomtuk-e a magnó REC gombját, hanem azonnal elkezd a felvételt. Ezért a Commodorehoz szokottak erre ügyeljenek. Ha a gép keres egy programot a magnón, akkor a státuszsorba kiírja a megtalált program nevét, de előtte nem törli a sort. Amennyiben a másik programunk neve rövidebb, akkor záró szóképek láthatók.

Az Enterprise számítógéppel kifejezetten jól lehet magnóra rögzíteni és olvasni a programokat. Hosszú idő alatt sem találtunk betölthetetlen programot, ellentétben más gépekkel. Ezért a magnót streamelésre is jól lehet használni.

Az évforduló margójára

Általános vélemény és tapasztalat, hogy az Enterprise 128 a kategóriájában és árfekvésében igen jó számítógép. A cég megszűnésének nem a konstrukció volt az



Előtérben a „maszek”, háttérben az „új” kontroller

oka, hanem az elhibázott kereskedelemplolitika.

Magyarországon az eladások mennyisége alapján a gép lassan meghatározónak mondható e kategóriában. A becslések szerint a nyolcvanezer Commodore 64 és az ötvenezer Spectrum után az Enterprise a harmadik helyre tör. Ez olyan elterjedtségre utal, hogy hosszú távon is számolni kell vele.

Amennyiben tárgyilagosan ítéljük meg a gép kezdeti értékesítési bonyodalmaikat, akkor az az igazság, hogy kicsit türelmetlenek voltunk. A szervizelés szinte az első pillanattól megoldott volt.

Ha szűkösen is, de kevesebb mint egy éven belül hozzájuthatunk az alapvető kiegészítőkhöz. Habár ez nem lehet mentesség, mégis emlékezzünk a más gépek értékesítése körül kialakult állapotokra. A Commodore gépekhez például évekig csak turista-forgalomban érkezett be periféria, és a töb-

bivel sem (volt) jobb a helyzet. Úgy tűnik, hogy Magyarországon egy új géptípus helyzetének viszonylagos „stabilizálásához” minimum egy évre van szükség. Ezért aztán természetesen érhető a géptulajdonosok adandó jogos türelmetlensége.

PINKE GYÖRGY

Mi kapható az Enterprise 128-hoz?

Az alábbi listát cikkünk megírásakor állítottuk össze, ezért az 1988. áprilisi helyzetet tükrözi. Az Enterprise géphez folyamatosan érkeznek kiegészítők és szoftverek. Azért ez a gépre jellemző összeállítás jól tájékoztatja az olvasót.

Enterprise egér 4 000 Ft
Enterprise SPEAKEASY beszédszintetizátor 3 750 Ft
Botkormány: Gan SHot 1 170 Ft
VT 94104 998 Ft
Medalist 790 Ft
(A botkormányhoz átalakító kell)
Botkormány-átalakító 456 Ft
Soros kábel (szabadvégű) 680 Ft
Párhuzamos interfész kábel (szabadvégű) 680 Ft
Monitorkábel 1 298 Ft
(A PROFESSIONAL Szerviz a gép videocsatlakozását díjtalanul átköti.)
Botkormányfogantyú 171 Ft
VT dupl. lemez meghajtó (illesztő nélkül) 46 700 Ft

EPSON RX80 nyomtató 63 600 Ft
Szoftver: Commodore Multi File
Transzfer 2 360 Ft
(A Commodore nyomtató és lemez meghajtó csatlakoztatását teszi lehetővé.)
Entervideo
segédprogram 529 Ft
(A német nyelvű gépben megtalálható függvényeket az angol nyelvűbe beleteszi.)
Fine Pen rajzolóprogram (sprite kezelő) 411 Ft
BASIC-bővítő 411 Ft
Egyszerűbb játékok (kb. 5-6 fajta) 390—470 Ft
Összetettebb játékok (kb. 6-7 fajta) 470—600 Ft
Irodalom: EXDOS (Novotrade kiadó) 300 Ft
Enterprise Controller (komplett) 10 550 Ft

Az Enterprise gépeket országosan javítja: a PROFESSIONAL Szerviz.

ÁTALÁNYDÍJAS JAVÍTÁSI ÉS KARBANTARTÁSI = ÖRÖK SZERZŐDÉS GARANCIA



**COMMODORE, ATARI, TVC stb. személyi számítógépek
valamint IBM PC/XT/AT professzionális PC számítógépek
és perifériák (floppy, printer)
garanciális és fizető JAVÍTÁSA, KARBANTARTÁSA.**

SZERVIZEINK:

1053 Budapest V., Magyar u. 12-14.

1083 Budapest VIII., Szigony u. 9.

1191 Budapest XIX., Gábor A. sétány 3.

3100 Salgótarján, Arany János u. 3.

3525 Miskolc, Fazekas u. 1-3.

T.: 173-551 Tx: 7621

T.: 343-153

T.: 32-14-007

T.: 46-17-011

4034 Debrecen, Holló László u. 14.

5600 Békéscsaba, Bartók Béla u. 37. T.: 66-27-195

6726 Szeged, Székelysor 13. T.: 62-13-377

7400 Kaposvár, Füredi u. 24. T.: 82-16-307

7624 Pécs, Jurisics M. u. 17. T.: 72-11-812

9700 Szombathely, Szalonok u. 31. T.: 94-14-519

1053 Budapest, Henszlmann I. u. 9. Telefon: 174-144 Telex: 22-7621

VEVŐSZOLGÁLAT — FOTOELEKTRONIK — NOVOTRADE — ALFA G.T.

VÉTEL — ELADÁS 1077 Budapest, Dohány u. 16. Tel.: 428-936

DIGITAL
szelvény
Mikroszámítógép Magazin
1988 július.

Egy sarokkal olcsóbb!!

A DIGITAL

Számítástechnikai Szaküzlet (Budapest, Szilágyi Erzsébet fasor 35. 1026) Sinclair-termékekre szakosodott: elsősorban a ZX81-es és Spectrum gépekhez használatos eszközöket, programokat árul. De kaphatók itt más gépekhez való tartozékok (például botkormányok), számítástechnikai alkatrészek (integrált áramkörök stb.) és zsebszámológépek is.

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

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

Az e havi kedvezmény

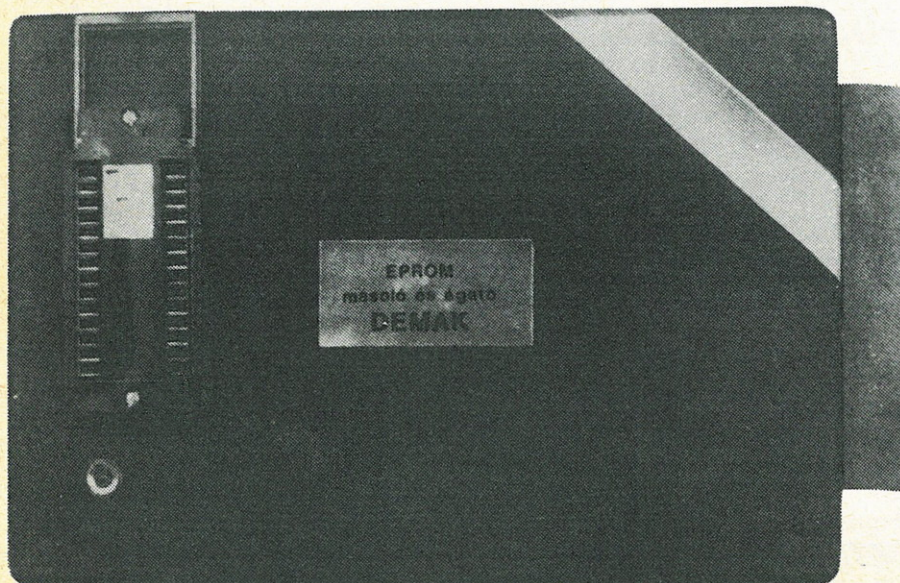
Intelligens EPROM-programozó. Ára 7500 Ft, engedmény 8%.

A DEMAK EPROM-égető 2716–27256 (2–32 k) típusjelű vagy ezekkel kompatibilis NMOS és CMOS EPROM-ok égetésére alkalmas. Az égetés normál vagy gyorsított Intel-algoritmus szerint dolgozhat. Az eszköz külső tápegységet nem igényel.

A működtető programot tartozékként kapja a vásárló. Ennek betöltése után menüből választhatjuk ki az égetendő típust. Ha netán téved valaki, visszaléphet. Ezután egy újabb menüre áttérve ki-

választhatjuk a műveletet. Olvasáskor az általunk kiválasztott területre másolja az EPROM tartalmát. Ha listázunk, az EPROM kezdő- és végcímével megadott tartományát hexadecimálisan kiírja a képernyőre. Íráskor az EPROM-ba beégeti a tár kezdő- és végcímével megadott tartományát. A művelet végrehajtása előtt ellenőrzi, hogy az EPROM üres-e. Ha nem, üzenetet ad. A művelet közben az esetleg nem égethető bájtnál leáll és hibáüzenetet ad. Az ellenőrző műveletnél egy adott tárterületet hasonlít össze az EPROM tartalmával. Az ürességet vizsgálat ellenőrzi.

S.E.



ISKOLASZÁMÍTÓGÉP- SZERVIZ

1077 Budapest VII.,
Baross tér 19.
Telefon: 428-999

Vállalja:

**IBM PC/AT,
IBM PC/XT**

és

**Commodore
típusú**

**(C16, C Plus/4,
C64, C128)**

**gépek
javítását,
átalánydíjas
szervizét,**

**egyedi
programok,
program-
csomagok
készítését.**

Minden kedden 17-től 20 óráig
HCC ENTERPRISE klub
a VSZM
Közösségi Házban
(Bp. XI. Fehérvári út 120.)
Klubvezető: Romvári Gábor
Telefon: 810-950/473

Olvásóink egyre gyakrabban küldenek be olyan hasznos programokat, amelyeket terjedelmi okok miatt nem áll módunkban közölni. Mivel azonban úgy véljük, hogy közérdeklődésre tarthatnak számot, ezeket röviden ismertetjük. Akik használni szeretnék a programokat, a részletes leírást, a programlistákat vagy — ahol erre mód van — a programot adathordozón, levélben megrendelhetik szerkesztőségünk-től. A listák, ill. adathordozók másolását a KASKAD Kiszövetkezet óbudai PÓLUS Szakcsoportja végzi (lásd lapunk 1988/2. számát). A másolási díj listák, leírások esetén oldalanként 8,— Ft, kazetta esetén 150,— Ft, floppy esetén 300,— Ft. Az adathordozóra történő másolás díja az adathordozó árát is magában foglalja.

Közületektől cégszerű megrendelést kérünk, a másolási díjat az MNB 208—42518—7014 számlára kell befizetni. Magánszemélyek a díjat a KKVMF PÓLUS Szakcsoport, Bp., Bécsi út 94—96. 1034 címre fizethetik be.

A megrendelés az alábbi formában lehetséges:

MIKROSZÁMÍTÓGÉP MAGAZIN SZERKESZTŐSÉGE 1371 Budapest, Pf. 433.

Megrendelem a Mikroszámítógép Magazin 1988/... számában szereplő közprogramok közül az alábbiakat:

A program neve
Csak programlistát és leírást kérek

Példányszám

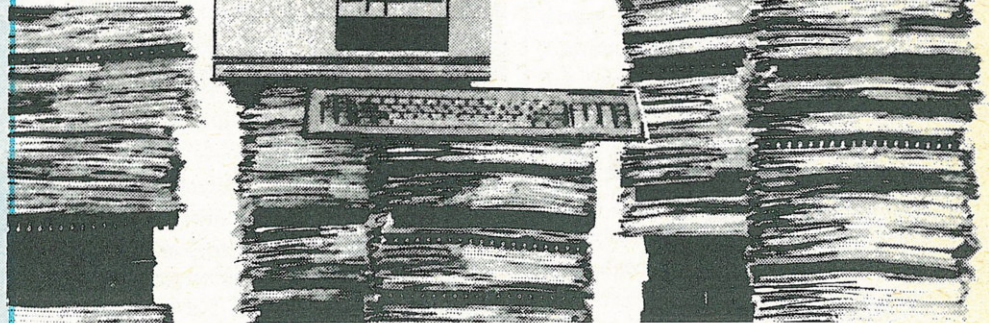
Csak adathordozót kérek
Mindkettőt kérem

A megrendeléshez csatolom a szolgáltatási díj befizetését igazoló csekkszelvényt. Dátum, név, pontos cím.

A programok ellenőrzése nem áll módunkban, ezért az esetleges programhibákért mi nem vállalhatjuk a felelősséget.

Kérjük, akik saját készítésű programjaikat — díjtalanul — felajánlják a köz javára, a részletes programleírást, a listát és az alábbiakhoz hasonló rövidített ismertetőt juttassák el címünkre. A megrendelők számára nevük jelenti a garanciát.

Közprogramok



Háttérzene

Programnév: Zene megszakításban
Géptípus: C64
Konfiguráció: alapgép
Adathordozó: kazetta
Terjedelem: 177 soros BASIC program
A készítő neve: Debre Attila
Megjegyzés: —
A leírás oldalszáma: 1
A program oldalszáma: 6

A program a C64 számítógép megszakító rutinját használja fel zene lejátszására. Lehetővé teszi, hogy a BASIC program beírása és futtatása közben egy állandóan ismétlődő háromszólamú dallamot játsszon a gép.

Karakterkészlet-készítő

Programnév: Saját karakterkészlet
Géptípus: Plus/4
Konfiguráció: alapgép
Adathordozó: kazetta
Terjedelem: 27 soros BASIC bemutatón és ismertető program, a \$4000—\$4707 területű (kb. 2 k) gépi kódú program, a \$1100—\$1227 (kb. 300 bájtt) területű gépi kódú program

A készítő neve: Lőrentei János
Megjegyzés: —
A leírás oldalszáma: 4

A program oldalszáma: 8

Kétféle — karakteres és grafikus — üzemmódban dolgozhatunk saját készítésű karakterkészlettel. A grafikus készlet érdekessége, hogy a képernyőn egyidejűleg több karakterkészlet is megjeleníthető. Példaként bemutatunk egyet, kinyomtatott formában.

Ékezetes karakterkészlet játékprogramokhoz

Programnév: Ékezetes ABC 3.0
Géptípus: C64
Konfiguráció: alapgép kazetta- vagy lemezegységgel
Adathordozó: kazetta, lemez
Terjedelem: 194 BASIC-sor
A készítő neve: Kristin Péter
Megjegyzés: létezik egy Graphic BASIC változat is, ami nyomtatni is tudja ezeket a karaktereket

A leírás oldalszáma: 1
A program oldalszáma: 3

A program eltolt BASIC START címet használva olyan BASIC-ben vagy gépi kódban írt programra építhető rá, amely a \$0800—\$106F tárterületet nem használja. A programmal magyar nyelvű üzenetekkel ellátott programok írhatók.

S. E.

[SHIFT + SZÁM]	Á	É	Ő	Ö	Ű	Ú	Ū	Ű	Í		
	1	2	3	4	5	6	7	8	9	0	
\$4000— \$41FF	Q	W	E	R	T	Y	U	I	O	P	+ -
	A	S	D	F	G	H	J	K	L	: ;	=
	Z	X	C	V	B	N	M	.	/	! "	'
\$4200—\$43FF COMMODORE	AZ EREDETI GRAFIKUS JELEK GRAFIKUS KEPERNYŐ										
	1	2	3	4	5	6	7	8	9	0	
BEÍRÁSI MÓD:	1	W	E	R	T	Y	U	I	O	P	" + -
NORMÁL											: ;
FELSŐRÉSZ		A	S	D	F	G	H	J	K	L	! " *
SHIFT											: ; ..
ALSÓRÉSZ											! " *
\$4400— \$46FF	Z	X	C	V	B	N	M	~	<	>	£ =

DUPLA MAGASSÁGÚ BETŰK

Sorozatunkban azokat az új hardver- és szoftvertermékeket ismertetjük, amelyek várhatóan általánosan elterjednek, és meghatározó szerepük lesz a fejlődés irányainak kialakításában.

Merre tart a világ?

gépek

Kártyaújdonságok,

További kártyák

Atari PromiseLAN

Az Atari cég találékonyan igyekszik növelni vásárlóinak táborát. Egyrészt az ilyen gépek tulajdonosainak lehetőséget ad arra, hogy más gépekkel összekapcsolódhassanak, másrészt arra, hogy összeköthessék az IBM PC-ket és a velük kompatibilis számítógépeket, valamint az Apple Macintosh család tagjait az Atarikkal — és egymással — közös hálózatba.

A rendszer tizenhét gép csillaghálózatba kapcsolására alkalmas. Nem igényel speciális kábelezést, a telefonvonalakat használja. A Novell cég NETBIOS (Network Basic Input and Output System) szabványával is és az AppleTalk előírásokkal is kompatibilis. A kártya használatával a felsorolt géptípusok bármelyike közvetlenül összekapcsolhatóvá válik a korábbi cikkünkben említett Atari lézernyomtatóval.

IBM OS/2 LAN Server V 1.0

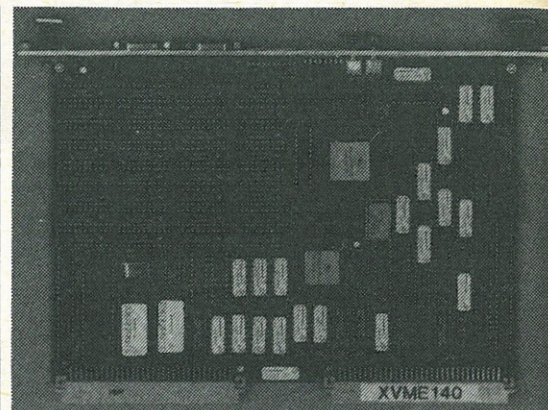
Ez az első hálózatvezérlő kártya az IBM új családjához. Lehetővé teszi, hogy egy hálózatban működtessék az új operációs rendszerben dolgozó gépeket a PC DOS operációs rendszert használó gépekkel. A hálózat akár az IBM Token-Ring, akár a PC Networks lehet.

Motorola VMEmodule™

A cég által használt VMEbus rendszerhez most már a 68030 típusú mikroprocesszorral készült központiegység-kártya is beszerezhető. Az XVME140 (1. kép) jellemzője nemcsak a mikroprocesszor, hanem az új MC 68882-es típusú lebegőpontos másodprocesszor is. Az első változat még csak 20 MHz frekvenciájú, de hamarosan elkészülnek a 25 és 33,3 MHz-es változatok is. A kártyán 256 k statikus RAM van, egyciklusos hozzáférési idővel, továbbá két ROM foglalat, két soros csatoló, számláló/időzi-

tő, különféle VMEbus illesztők, indikátorok, kapcsolók találhatók rajta. A program egyelőre egy Debug monitor, mivel a kártya forgalmazásának célja a 68030 alapú rendszerek fejlesztése.

A lebegőpontos segédprocesszor regisztereit a 68030-as mint sajátjait használhatja. A processzor kielégíti az IEEE 754 szabványt. A VMEbus Master Interface lehetővé teszi 16, 24, 32 bites cím-, 16, 32 bites adatvonalak csatolását. Buszprioritást kezelő egysége, hét megszakítási szintű megszakítási rendszere van. Az újraindító jel



1. kép

2. kép



jöhet a buszról, a RESET gombról és a mikroprocesszortól. Az MC68681 DUART (Dual Asynchronous Receiver/Transmitter) egy kettős soros csatornát és számláló/időzítő kört tartalmaz, amely nyolcbites csatornán keresztül érhető el a mikroprocesszor felől. A soros csatornák egymástól függetlenül működtethetők, maximálisan 9600 baud sebességgel. A csatornák kezelését az IC-be égetett program végzi. Ugyanez vonatkozik a számláló/időzítő funkciók kezelésére is.

Számítógépek

Amstrad gépek

Sorozatbeli szokásunkhoz hiven az itthon nem vagy alig ismert cégek termékein kívül a céget is bemutatjuk. Ez a cég ilyen,

bár termékei közül jó néhány darab van már nálunk.

A cég tulajdonosa huszonegy évesen alapította a vállalatot 1968-ban. Első számítógépéből, a CPC 464-ből 1984-ben több mint 200 ezret adtak el, és az „Év számítógépe” címet kapta az európai újságíróktól. Ugyanebben az évben bevezették az első szövegszerkesztő rendszerüket (nyomtatóval), a PCW 8256-os típust. 1986-ban felvásárolták a Sinclair céget, és ezzel a legnagyobb mikroszámítógép-gyártóvá nőttek ki magukat az angol piacon. A PC 1512 típusú gépük 1987-ben az angol piacon „Az év terméke” lett — megelőzve például a Jaguar autómárkákat. A COMDEX SPRING volt a helye az egyik ismertető gép, a PC 1640 bemutatásának, a PPC 640-esé pedig a COMDEX FALL.

A PC 1640 PC/XT kompatibilis gépet 8

Software is. A rendszer (2. kép) igen tömör felépítésű.

A PPC 640-es könnyű, teljesen hordozható gép (3. kép). Mindössze 5 kilót nyom. A képernyőforátum a PC-szokvány (80×25, illetve 640×200). A kijelző egy döntött LCD. Egy vagy két 3,5” méretű lemezegysége és 640 k RAM tára van. Belepítettek egy 300—2400 baud sebességű modemet is. Billentyűzete teljes méretű AT-kompatibilis. A gép tartozéka az MS DOS 3.3 telekommunikációs program. A kommunikációhoz soros és párhuzamos csatornával látták el. Monitorcsatlakozó segítségével normál méretű monitorral is használható. Ami szintén szokatlan, hogy gépkocsi-akkumulátorról vagy villanófénytápegységről is üzemeltethető.

Ennek a gépnek elkészítették egy egyszerűsített, olcsóbb változatát, a PPC 512-es típust is. Ennél nincs modem és a tárméret 512 k. Egyebekben azonos az előzővel.

Atari PC kompatibilis gépek

A PC1 egy 512 k tárkapacitású gép; mind önálló, mind helyi hálózatban munkaállomásként használható.

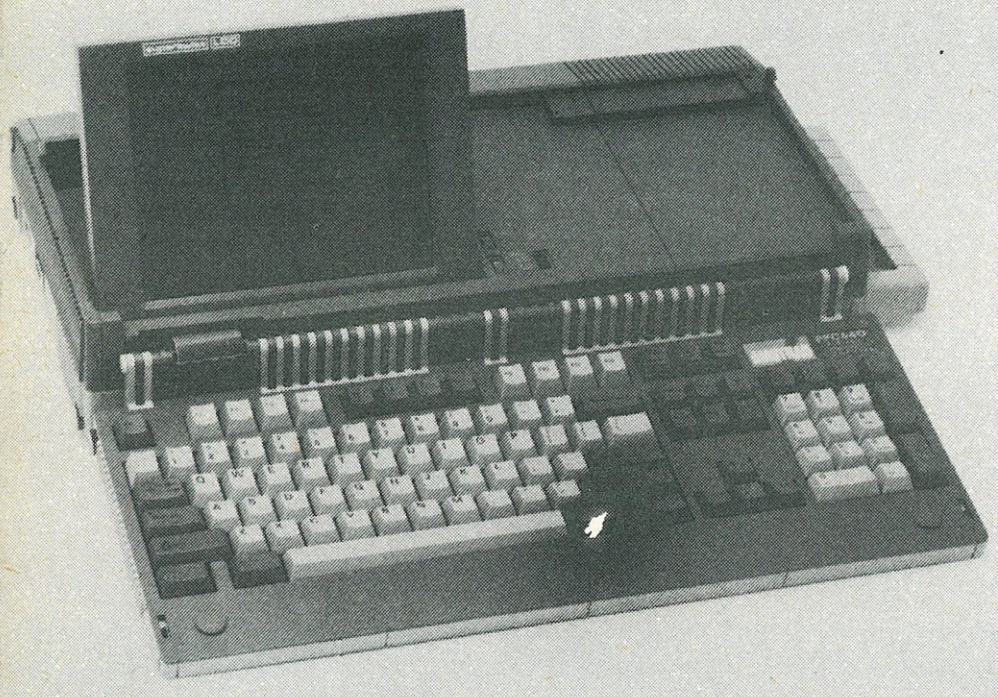
A PC2 normál és gyorsított órasebességgel dolgozik. A sebesség programból kapcsolható, mégpedig a rendszer újratöltése nélkül. A billentyűzet AT típusú. Egér, párhuzamos csatorna, soros csatorna és nagy felbontású monitor tartozik az alapszerkezethez. A beépíthető lemezegységek 5,25 és 3,5” méretűek lehetnek. A tárméret 256 k a kijelző számára, és további 512 k a felhasználó részére. A tár 640 k-ra bővíthető. Öt XT stílusú bővítőkártyának alakítottak ki helyet az alapkártyán.

A PC4 AT-kompatibilis 8, 12 MHz órafrekvencián dolgozó gép. Mindazt tartalmazza, mint az előző. Annyiban mégis eltér attól, hogy a tármérete 1 M, szalagegysége is lehet, és a bővítőkártyái (4 db) AT típusúak. A Hercules, CGA, EGA, MDA típusú grafikon kívül VGA-ja is van.

IBM PS/2 sorozat

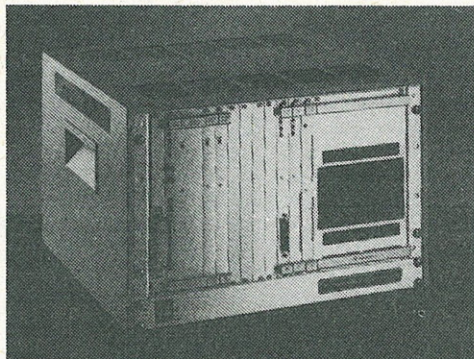
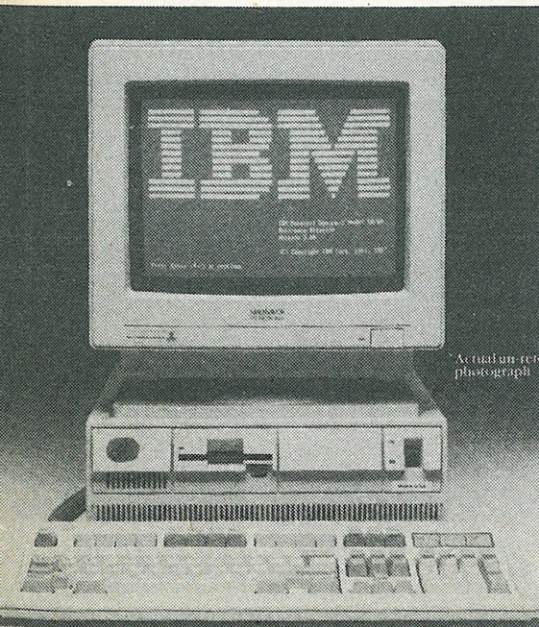
A piac meghatározó cégének új sorozata éles vitát váltott ki: vajon eltűnik-e az eddigi PC és kompatibilis gépek által uralt világ, és hódít-e a PS/2 – és az esetleg majd megjelenő másolatai —, vagy sem. A tavaly tavasszal megjelent rendszerből (4. kép) az IBM mindenesetre fél év alatt eladott egymillió darabot. A család választéka bővül, és igaznak látszik a gyártó azon állítása, hogy ez az eddigi legnagyobb kínálata. A rendszer néhány új vonását már a sorozat korábbi számában megemlítettük, az összes újdonságának ismertetése azonban sokkal nagyobb terjedelmet igényelne.

A leginkább hangoztatott nézetek szerint marad a „rég” világ, kiegészítve az „új” világgal néhány jellemzőjével. Az eltérés az



3. kép

4. kép



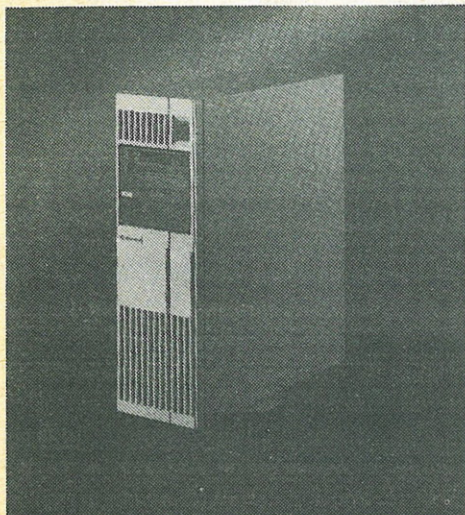
5. kép

MHz órafrekvenciájú 8086-os mikroprocesszorral, Hercules kompatibilis monokrom és CGA, valamint EGA kompatibilis monitorillesztő kártyával, 640 k RAM tárral, egérrel, soros/párhuzamos csatornával, hajlékony- vagy merevlemez egységgel szállítják. A készülék tartozéka az MS DOS 3.2, és ami szokatlan, a GEM Desktop

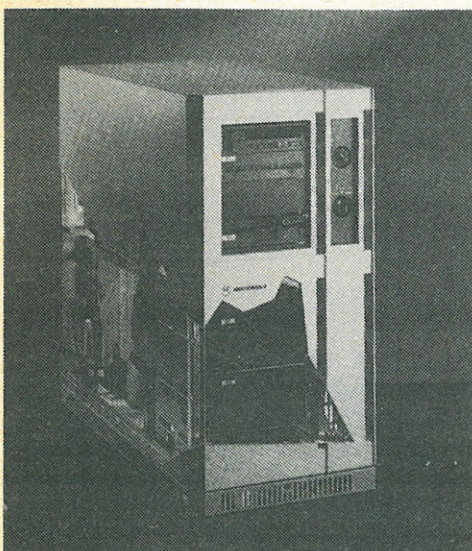
egyes vélemények között elsősorban abban van, hogy melyek lesznek az elterjedő „új” jellemzők. A csökkent méretű lemezegek máris széles körben elterjedtek, mint ahogy a nagyobb felbontású grafika (VGA) is.

Motorola számítógépek

A cég a VME Delta sorozatát folytatja. A Unix és Ethernet bázisú rendszert egyrészt az eddigi 68020-as típusú gépek lehe-



6. kép



7. kép

tőségeinek kibővítésével, másrészt új 68030 típusú gépek bevezetésével igyekeznek vonzóbbá tenni.

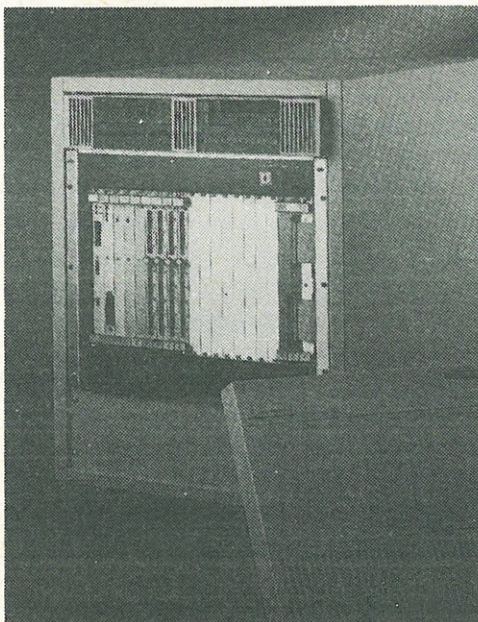
A fiókrendszerű Model 1132 (5. kép) 12 fiókú, ipari folyamatszabályozások, adatgyűjtések alkalmazásához készült, 68020-as alapú, 16 k gyorstárú, 4–16 M RAM kapacitású, 2–34 soros csatornájú, 67–161 M merevlemezegységű, 60 M szalagegységű eszköz.

A műszaki és hálózatos megoldásokra a Model 2334 (6. kép) 3 MIPS sebességű egységet ajánlják. A rendszer tíz felhasználót fogadhat, kevesebb a tárkapacitása és kisebb a soros csatornaszáma.

A Model 3641 (7. kép) 68030-as típusú gép, 68882-es segédprocesszorral. 64 k

gyorstár, 12 fiók, maximálisan 32 M RAM, 50 soros csatorna, 1,3 G (gigabájt) lemezegeység tartozhat hozzá. Hasonló célra szolgál, mint az előző, csak nagyobb rendszerekhez ajánlható (például 50 felhasználó a maximum).

A csúcs a Model 3841 (8. kép), amelynek 48 M tára, 66 soros csatornája, 1,6 G merevlemezegysége, 120 M szalagegysége van.



8. kép

9. kép

Az utóbbi két egység sebessége 6 MIPS (millió utasítás másodpercenként).

SAIC SIGMA Neurocomputer Workstation

A sorozat első számában már szó volt a cég párhuzamos processzoráról. Ezek a számítógépek ennek a használatán alapulnak.

A Sigma egy 80386/80387 PC/AT-kompatibilis gép, 16 MHz órajelektívával, 1 M RAM-mal, 80 M merevlemezegységgel, 1,2 M hajlékonylemez-egységgel, 640 × 350 pontos grafikával, párhuzamos és soros csatornával, optikai érzékelős egerrel (9. kép). A gép az ablaktechnikát alkalmazó ANSim szoftverfejlesztő környezettel képes tizenegy felhasználó által konfigurálható idegrendszeri hálózatmodellel maximálisan 100 ezer kapcsolat és 25 ezer folyamatlelem működésének szimulálására; másodpercenkénti sebessége 25 ezer kapcsolat.

A Sigma 1 hasonló az előzőhöz, csak kiegészítve a már hivatkozott Delta Floating Point Processor kártyával. Ez a szuperszámítógép a már említett szoftverkönyezettel és az azt kiegészítő Delta C, Delta Macro szoftverrel képes egymillió kapcsolat, egymillió folyamatlelem szimulálására, és a sebesség 10 millió kapcsolat másodpercenként (!).

SIMONYI ENDRE



Helyreigazítás

Lapunk 1988/6. számában a Közprogramok rovatban a Paróczay Gábor által beküldött Számkonverter program nem TVC-re, hanem ZX-Spectrum gépre íródott. A hibáért elnézésüket kérjük.

A szerkesztőség

Fájlok feldolgozása magnóval

A program C Plus/4 gépen futtatható. Működéséhez elengedhetetlenül szükséges a turbo cartridge használata. A program lehetővé teszi valós és sztringtömbök gyors kimentését, illetve betöltését datasette használatával. Ezáltal a magnó lassú működése adatfeldolgozás esetén részben kiküszöbölhető.

A program két részből áll. Az egyik a 10–29-es sorig terjed és tömbök kimentését hajtja végre. A 20-as sorban ellenőrizzük a tömb fajtáját. Amennyiben valós tömbről van szó, a 22-es sor meghatározza a tömb hosszát, a 23-as a kimentés végcímét, a 24-es pedig kimenteti a tömböt. A kimentésnél igénybe vesszük a cartridge ama lehetőségét, hogy visszatöltéskor nem állítódik be a program kezdete, illetve a program vége mutató. A 27-es sorban, amennyiben a tömb valós volt, a program megáll.

Ha a 20-as sorban kiderül, hogy sztring típusú tömbről van szó, a végrehajtás a 40-es sorral folytatódik. A KE, illetve VE változóba bekerül annak a memóriaterületnek a kezdő, illetve végső értéke, ahol a

sztringtömb elemeinek értékei találhatók. Ezután a program visszatér a 23-as sorhoz. Ezúttal a 24-es sor kimenteti a sztringtömböt, amely az elemek értékeinek kezdőcímét és hosszát tartalmazza. Mivel SAVE után a gép direkt üzemmódba lép be, ezért a 24-es sorban a SAVE parancsot PRINT-tel adjuk ki. A két következő sor biztosítja, hogy a program ne szakadjon félbe. A 28-as sor kimenteti a sztringtömb értékeit tartalmazó területet és a program megáll.

Betöltéshez a program belépési pontja a 30-as sorban van (oda GOSUB 30-cal megyünk). Fontos megjegyezni, hogy amennyiben a 36-os sorban kiderül, hogy a betöltött tömb sztring típusú, a program betölti a sztring karaktereket, beállítja a „Karakterfüzér kezdete” mutatót, a „Felhasználói füzér” mutatót pedig a sztringértékek elé állítja, hogy adatfeldolgozás közben a sztringértékek el ne vesszenek.

A program szabályos működésének feltételei:

1. A felhasználói programba ne deklaráljuk a tömböt.
 2. A tömböt csak létrehozáskor kell deklarálni direkt üzemmódban.
 3. Csak egydimenziós tömböt használjunk.
 4. A tömb neve csak egy karakterből álljon.
 5. A program a sorrendben első tömböt menti ki. Visszatöltéskor a mutatók átalításával minden egyéb adat törődik és a tömb eredeti helyére kerül vissza.
 6. A program a felhasználói programok szerves részét képezi (együtt futtatandók).
 7. Saját programunk futtatása mindig az előzőleg kimentett tömb betöltésével kezdődjön.
 8. Amennyiben sztring típusú tömbevel dolgoztunk, kimentés előtt mindenképpen elemrendezést kell végrehajtani: FOR I=0 TO N: D\$(I)=D\$(I)+"" :NEXT Ezáltal a tömbelemek növekvő sorrendben, egymás után következnek.
- Ha tehát elkészült a programunk, akkor RUN/STOP-pal leállítjuk, direkt módban deklaráljuk a tömböt és GOTO-val folytatjuk. Az adatfeldolgozás befejeztével GOTO 10 utasítással a tömböt szalagra mentjük.

SZIMEON PETKOV

```

1 REM VALOS ES STRING TOMBOK GYORS KIMENTESE ILL. BETOLTESE
5 REM ***** SZIMEON PETKOV  C-1988,BULGARIA *****
6 REM PRINTER JELEI
7 REM ***@-CLEAR HOME*****Q-KURZOR LE***
10 POKE45,PEEK(47):POKE46,PEEK(48)
19 NC=PEEK(47)+256*PEEK(48)
20 IF PEEK(NC+1)=128 THEN NC=PEEK(NC+6)+256*PEEK(NC+5):NC=NC*3+7:GOSUB40:GOTO23
21 NC=PEEK(NC+6)+256*PEEK(NC+5)
22 NC=NC*5+7
23 NC=PEEK(47)+256*PEEK(48)+NC:VE=VE+PEEK(PEEK(47)+256*PEEK(48)+7)
24 PRINT"@";"SAVE"+CHR$(34)+"FILE"+CHR$(34);",7,7,";PEEK(47)+256*PEEK(48);",",",
NC
25 PRINT"QQQGGOTO 27"
26 POKE239,3:POKE1319,19:POKE1320,13:POKE1321,13:STOP
27 IF PEEK(PEEK(47)+256*PEEK(48)+1) 128 THEN END
28 SAVE"STRING",7,7,KE,VE
29 END
30 REM
31 LOAD"",7
32 POKE47,PEEK(819):POKE48,PEEK(820):POKE49,PEEK(821):POKE50,PEEK(822)
33 REM ELEMENK SZAMA NC-BE
34 NC=PEEK(47)+256*PEEK(48)
35 NC=PEEK(NC+6)+256*PEEK(NC+5)-1
36 IF PEEK(PEEK(47)+256*PEEK(48)+1) 128 THEN RETURN
37 LOAD"",7
38 FU=PEEK(819)+256*PEEK(820)-2:POKE 54,INT(FU/256):POKE53,FU-256*PEEK(54)
39 POKE 51,PEEK(819):POKE52,PEEK(820):RETURN
40 REM STRING TOMB KEZDETE ES VEGE
41 KE=PEEK(47)+256*PEEK(48)+NC-1
42 KE=PEEK(KE-1)+256*PEEK(KE)
43 VE=PEEK(47)+256*PEEK(48)+8
44 VE=PEEK(VE)+256*PEEK(VE+1)
45 RETURN

```

ADOM A MAGYARÁZATOT

A Magazin ez évi 5. számában feltett első kérdés az volt, hogy a hatványok kifejezésének egyik átalakítása:

$$e^x = (e^{x/n})^n, \quad (1)$$

ami a középiskolai matematikaanyag szerint helyes, miért nem ad helyes eredményt számítógéppel ellenőrizve.

A számítógépek az exponenciális függvényt csak közelítő pontossággal számolják. Erre vagy hatványok összegét

$$\sum_{i=0}^m a_i \cdot x^i$$

vagy ilyenek hányadosát

$$\left(\sum_{i=0}^{m_1} a_i \cdot x^i \right) / \left(\sum_{j=0}^{m_2} b_j \cdot x^j \right)$$

használgják. A példában $x=1$ eset szerepelt. Így $e = (e^{1/n})^n$;

$$\sum_{i=0}^m a_i; \left(\sum_{i=0}^{m_1} a_i \right) / \left(\sum_{j=0}^{m_2} b_j \right)$$

„e” irracionális szám, ami ezért sem véges számú racionális szám összegeként, sem összegük hányadosaként nem állítható elő teljes pontossággal, viszont tetszés szerinti pontossággal igen. Ha például az engedélyezett hiba h , és $h < 1$, akkor lehet $m=0$, $a_0=2$; ha $h < 10^{-9}$, akkor lehet $m=0$, $a_0=2,71828183$.

A függvényt közelítő összefüggésnek természetesen nem egyetlen érték (például $x=1$) esetében kell megfelelő pontosságúnak lennie, hanem abban a tartományban, amit a függvényre engedélyeztek. Itt már nem elégséges egyetlen állandó használata. (Az egyik 8 jegyre pontos hatványsorozat például 9 állandót tartalmaz.)

A matematikában elterjedten használják az ún. Taylor-polinomot a közelítésre. Előnye, hogy egyszerű szabállyal állítható elő. Az exponenciális függvényre a szabály

$$a_i = a_{i-1} \cdot x/i \text{ és}$$

$$T_m = \sum_{i=0}^m \frac{x^i}{i!}$$

(Az egyszerű szabály egyszerű előállító programot eredményez. A sort mégsem használják a BASIC-fordítókban, mert aránylag sok tag használata szükséges, ami lassítja a számítást.)

A polinom minden határon túl növelve „m” értékét, az exponenciális függvényt adja. Így

$$e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$$

Amennyiben valamilyen véges „m” értékig összegzünk, úgy

$$e^x = h + \sum_{i=0}^m \frac{x^i}{i!} = h + T_m \quad (2)$$

illetve

$$h = e^x - T_m = \sum_{i=m+1}^{\infty} \frac{x^i}{i!}$$

Ha x elég kicsi, akkor $m=2$; $e^x \sim 1 + x + \frac{x^2}{2}$ elég, és ilyenkor

$$h \sim \frac{x^{m+1}}{(m+1)!} = \frac{x^3}{6}$$

Legyen adott esetben BASIC-fordítónk maximális pontossága 10^{-10} . Így

$$\frac{e^x - h}{e^x} = 10^{-10}, \text{ akkor } \frac{x^3}{6} \leq 10^{-10}, \text{ vagyis } x \sim 8 \cdot 10^{-4}.$$

(2)-t (1)-be helyettesítve:

$$e^x = [T_m(x/n) + h(x/n)]^n = T_m^n(x/n) + n \cdot T_m^{n-1}(x/n) \cdot h(x/n) + \dots \quad (3)$$

ahol x/n a függvények új független változója. A kérdéseskor bemutatott hibátáblázatból látható volt, hogy a hiba sokkal kisebb, mint a közelítő érték: legfeljebb százszázalék. A (3) jobb oldalának további tagjaiban a közelítő érték mind kisebb, míg a hiba mind nagyobb hatványai szerepelnek, ahogy a következő képletben a hiba négyzete, ami a közelítő értékhez képest már csak billiomod százaléka. A hatványok szorzói is nőnek, de nem olyan gyorsan, mint ahogy a hibahatványok értéke csökken. Jó közelítéssel a további tagok elhanyagolhatók. Így

$$e^x \sim T_m^n(x/n) + n \cdot T_m^{n-1}(x/n) \cdot h(x/n) = T_m^{n-1}(x/n) \cdot [T_m + n \cdot h(x/n)] \quad (4)$$

A számítógépnek a számkerekítés (h_k) miatti hibája (4) szerint a második tag

$$h_k \sim n \cdot h(x/n) = n \cdot \frac{x^3}{6}. \text{ A feladatban}$$

$$n = 2 - 4096 \text{ értékű volt.}$$

Legyen például $n=2048$. Így $h_k \sim 5,4 \cdot 10^{-7}$, vagyis sokszorosa a BASIC-fordító alpműveletek végzésénél elkövetett hibájának. Miért? Az $n=2048$ esetén mivel $n=2^k$ volt, $k=11$, vagyis ennyiszor szoroztuk meg önmagukkal a számítás során a részeredményeket. Az első szorzandó számításánál fellép a kerekítési hiba, ami (4) szerint többszöröződik minden szorzásnál. Mivel a feladatban $x=1$ eset volt, ezért a (2) egyenletben csak $m=12$ esetén kapunk megfelelően pontos közelítést, és a hiba nem számítható olyan egyszerűen. (Így például a 14. tag $1/13!$, a 15. tag $1/14!$, vagyis a 15. tag még a hiba mintegy 7 százalékát adja. Elhanyagolása tehát túl nagy hibával jár.) Hasonló a helyzet a (4) egyenletnél is.

Ugyanez a gondolatmenet – sokkal bonyolultabb számítással – igaz a 2. feladatnál is.

Összefoglalva: nem a vizsgált matematikai tételek hibásak, hanem az az elgondolás, hogy a tételek helyességének ellenőrzésére ez a számítógépes módszer megfelelő.

SIMONYI ENDRE

KI AD MAGYARÁZATOT?

1 ≠ 1

Az ábrán egy GW-BASIC programnak IBM PC kompatibilis gépekre írt változata látható. A program ún. kétszeres pontosságú változónak adja meg az a, b, c változókat, majd két módszerrel – az EXP függvény használatával és a Taylor-polinommal – kiszámítja az értékét. Az EXP függvény független változóját behelyettesítéssel adtuk meg. Látható, hogy eltérés csak a 17. jegyben van.

GW-BASIC 2.02

(C) Copyright Microsoft 1983,1984

Compatibility Software GW-BASIC V2.02

Copyright (c) 1984 by Phoenix Software Associates Ltd. All Rights Reserved.

62194 Bytes free

OK

10defdbl a,b,c:a=1:b=exp(a):print b

20c=1:for i=1 to 30:a=a/i:c=c+a:next i:print c,abs(c-b)/c

run

2.718281828459045

4.084282258747710-17

OK

list 10

10 DEFDBL A,B,C:A=1:B=EXP(1):PRINT B

run

2.718281745910645

3.036785956954907D-08

OK

11LIST 2RUN 2LOAD" 4SAVE" 5CONT 6,"LPT"

7TRON 8TROFF 9KEY 0SCREEN

A 10-es sort úgy módosítottuk, hogy az EXP függvény független változójának értékét közvetlenül számmal (1) adjuk meg. Az eltérés sokkal nagyobb lett. Most már a 8. jegytől eltér a két érték. Miért? Vajon igaz a cím?

S. E.

Közületek figyelem!
Mikroszámítógépet
akarnak vásárolni?
Tájékozódjanak
a naprakész piaci helyzetről!
Díjtalan ismertető!
MESZ
Számítástechnika
1368 Budapest, Pf. 193

MINFORM

Rovatunkban az Apple, Atari, Commodore és Sinclair mikrók tulajdonosait feltehetően érdeklő, angol és német nyelvű cikkekről informáljuk olvasóinkat egy tartalomleíró szőlánc segítségével.

A forráshely karaktersorozatát nyíll vezeti be, ezt a / jelleg a folyóirat kódja követi (lásd táblázat). A két / jel között a megjelenési adatokat (év, hó), illetve a cikk kezdő oldalszámát szerepeltetjük. A második / jel után pedig - az esetleges másolatkérés megkönnyítendő - a cikk teljes oldalterjedelmét közöljük.

A tartalomleíró szőláncok permutálósával és alfabetikus rendezésével szerkesztett teljes anyagot az OMIKK "APACS" c. kiadványsorozata tartalmazza (vevőszolgálati telefon: 341-765). Lapunk ebből csak a "programlista" címszóval kezdődő részletet teszi közzé.

A folyóiratok megtekinthetők a SZÁMALK (Bp. XI. Szakasits Á. u. 68.), illetve - a x-gal jelzettek az OMIKK (Bp. VIII. Múzeum u. 17.) szakkönyvtárában. Másolatok a SZÁMALK-tól csekszelvény, az OMIKK-tól megrendelőlevél bekielődésével, vagy személyesen rendelhetők.

PROGRAMLISTA
c64 nyomtatás színes indigoKeszlet t
obbmenetes. hasznalata
->64er/87.04-82/1

PROGRAMLISTA
illesztes master-text nyomtatás alka
lmazasi utmutato 3.resz
->64er/87.04-84/2

PROGRAMLISTA
jatekprogram Keszites tanacsok pelda
program ->happ/87.04-112/3

PROGRAMLISTA
proterm tavadatvitel c64
->64er/87.04-52/8

PROGRAMLISTA
radiozas barkacsolas c64 rttv/cw Kon
verter alkatreszlista kapcsolasi raj
z ->64er/87.04-60/3

PROGRAMLISTA
spectrum beszedkimenet mikrofonnal k
azettara beszeditt beszed digitaliza
lasa ->happ/87.04-72/2

PROGRAMLISTA
spectrum masodik monitorKep generala
sa ->happ/87.04-71/1

PROGRAMLISTA
speedscript 80 szovegfeldolgozas app
le ii alkalmazasi utmutato gepi Kodu
80-oszlopos valtozat
->cute/87.04-41/14

PROGRAMLISTA
strukturalt basic c64 15 uj/modosito
tt utasitas ->cute/87.04-92/4

PROGRAMLISTA
szovegKezeles amiga programbol vezer
elt betutipus valtas
->cute/87.04-91/2

PROGRAMLISTA
szovegKezeles atari st fileKezeles r
utingyujtemeny ->anti/87.04-56/8

PROGRAMLISTA
tablazatKeszites c64 fileKezeles sza
molo-editalo modul a <datafile>hez (87.02 es 03) ->run/87.04-52/15

PROGRAMLISTA
terKep c64 grafika oktatás (foldrajz)
<fractale 9.0> ->64er/87.04-69/4

PROGRAMLISTA
veletlenszam atari x1/xernd general
as retry count modositás
->cute/87.04-78/2

PROGRAMLISTA
virus crackerek mailbox anti-hardvi
rus-filter ->happ/87.04-76/1

PROGRAMLISTA
vizamrite c64 alkalmazasi tippek 12.
resz naptar szerkesztes
->64er/87.04-86/3

PROGRAMLISTA
zene atari x1/xer jatekprogram Keszit
es hanghathas/hatterzene beepites lis
tazas-lassito rutin
->anti/87.04-16/6

PROGRAMLISTA
vezerles barkacsolas c64 jatek lanc
alpas jarmu-modell iranyitasa
->run2/87.02-44/6

PROGRAMLISTA
video barkacsolas c64 illesztőaranko
r vevo es editorprogram a <videodat>
-hoz tv-kejelbe illesztett adatinfo
rmaciok ->mc/87.03-72/7

PROGRAMLISTA
visszamentes c64 fileKezeles adatmen
tes lezaratlan filebol unsplat
->cute/87.01-83/2

PROGRAMLISTA
visszamentes c64 program visszanyere
s formazas utan disk-retter
->happ/87.03-48/2

PROGRAMLISTA
wordstar nyomtatás Kiegészites epon
tipusu modellekhez
->happ/87.02-108/3

PROGRAMLISTA
zene c16 <sound ii>
->chip/87.02-134/2

PROGRAMLISTA
zene c64 alkalmazasi utmutato <music
maker 64> ->cute/87.01-63/3

PROGRAMLISTA
zene c64 filetomorites <sound-crunch
er> ->64er/87.03-91/1

PROGRAMLISTA
zene c64 illesztés midi csatornaveze
ries hangszer-billentyuzet hozzarend
eles ->etor/87.03-31/3

PROGRAMLISTA
amiga deluxe paint fileKezeles grafi
kaKepfile Konvertalas basic-hez <if
f translator> ->cute/87.04-104/2

PROGRAMLISTA
apple ii lemezKezeles szabad Kapacit
ast nyivantarto rutin
->chip/87.04-295/1

PROGRAMLISTA
atari x1/xer autorun.sys-file general
as a lemezen ->cute/87.04-81/1

PROGRAMLISTA
atari x1/xer break-kiiktatas hidegsta
rt ->anti/87.04-20/3

PROGRAMLISTA
atari x1/xer grafika KepernyoKezeles
basic programozas 13.resz
->anti/87.04-23/4

PROGRAMLISTA
atari x1/xer jatek <taxman>
->anti/87.04-35/4

PROGRAMLISTA
atari x1/xer nyomtatás print shop raj
z/karakter atvitel ragaszthato cimke
re ->anti/87.04-9/6

PROGRAMLISTA
atari x1/xer szamoutput formatalas
->happ/87.04-75/1

A folyóirat neve	Kódja
x 64'er Magazin	64er
Antics	anti
x Chip Magazin	chip
x Compu!u!	cutu
x Dr. Dobb's Journal	dobb
Elektr Electronics	etor
Happy Computer	happ
x mc - Zeitschrift	mc
Run (USA)	run
Run (NSZK)	run2
x Your Computer	your
x ZX Computing Monthly	ZXCM

PROGRAMLISTA
atari begepelési utmutato segedprogr
am ->anti/87.04-74/2

PROGRAMLISTA
b-help c64 tarolorezidens segedlet b
asic programozashoz
->run2/87.04-102/10

PROGRAMLISTA
c128 fileKezeles programok/szubrutin
ok osszeolvasztasa merge 128
->cute/87.04-96/2

PROGRAMLISTA
c16 grafika plus/4 <graffito> 2 Kbyt
e tarigeny ->run2/87.04-120/2

PROGRAMLISTA
c16 jatek <miner 410er>
->run2/87.04-135/3

PROGRAMLISTA
c64 cimdefinialo segedlet goto/gosub
-hoz utolagos sorszamozas
->run2/87.04-112/8

PROGRAMLISTA
c64 grafika hyperscan mandelbrot Kep
ek eloallitasa/nyomtatasa
->cute/87.04-82/6

PROGRAMLISTA
c64 grafika KepernyoKezeles szegely-
terulet felhasznalasa split-screen t
echnika ->cute/87.04-100/2

PROGRAMLISTA
c64 grafika Kepmezo nagyitas Kicsiny
ites ram(0-\$dfff) athelyezés lemezre
->chip/87.04-293/3

PROGRAMLISTA
c64 jatek <asteroidenfeld>
->run2/87.04-128/2

PROGRAMLISTA
c64 jatek <Kiosk>
->run2/87.04-130/5

PROGRAMLISTA
c64 jatek <mister tooth>
->run2/87.04-144/5

PROGRAMLISTA
c64 jatek <weltendaemmerung>
->happ/87.04-47/5

PROGRAMLISTA
c64 KepernyoKezeles nyomtatás screen
-dump/freeze szubrutin
->run/87.04-14/2

PROGRAMLISTA
c64 KepernyoKezeles szovegKiirtas
type' utasitással
->run2/87.04-122/2

PROGRAMLISTA
c64 menutechnika programba illeszthe
to rutinok ->run/87.04-72/2

PROGRAMLISTA
c64 nyomtatás max. 30 karakter nagy
tott abrazolasa ->happ/87.04-56/1

Szoftverminősítő szoftver

A szoftverfejlesztők és a felhasználók régi álma egy olyan szoftver, amellyel a programok egységesen dokumentálhatók, viszonylag objektív módon minősíthetők és minimális ráfordítással teljeskörűen tesztelhetők. Nos, az SZKI SCI-L egyik sikerterméke, a QUALIGRAPH minőségmérő és dokumentáló program évekkel ezelőtt éppen ilyen célra készült. 1987 elejéig azonban kizárólag a nyugati országokba exportálták, Amerikától Ausztráliáig. Magyarországon ugyanis korábban nem érdeklődtek iránta; igaz, szoftverfejlesztéssel és -kereskedelemmel is csak néhány nagyvállalat foglalkozott. Ez az oka annak, hogy a QUALIGRAPH programkimenetének és dokumentációjának nyelve máig is az angol. (A cikkünk illusztrálására bemutatott ábrákon is ezért angol nyelvűek a feliratok.)

A szoftverminőség és mérése

A program minőségmérésének még nem alakult ki a konkrét mérőszámrendszere. Ezért — mint ilyen helyzetekben szokásos — a szakemberek más tudományághoz fordulnak, és analógiákat keresnek. A matematikában a gráfelmélet, az automata-, az információelmélet, a matematikai statisztika stb. mérőszámrendszerével találtak megfeleltetést; a QUALIGRAPH is ezeket használja.

A gráfelmélet szerinti megfeleltetésnél a programot irányított gráfként fogják fel. A gráf élei a program elágazásait, a vezérlésátadásokat, szögpontjait az utasítászekvenciákat, a hurkok a visszacsatolást jelölik. Az utasítászekvenciának a két vezérlésátadó utasítás között levő programsorokat tekintjük. Így például a ciklomatikus szám: $C = e - n + 2$, ahol e az élek száma, n a szögpontok száma. Az e értéke növeli, n csökkenti C értékét. Ez azt jelenti, hogy két program közül az a bonyolultabb, amelyikben az elágazások száma nagyobb. A mutató javasolt értéke 10-15 modulonként.

A vezérlési sűrűség mérőszáma szintén a szögpontok és az élek számát vizsgálja, de arányszámként, mint az egy szögpontra eső élek számát.

Az információelmélet fogalomrendszerében a programot közvetítő csatornaként fogják fel, amelyen keresztül az információ áramlik. Így alkalmazható a programra mérőszámként az entrópia.

A matematikai statisztikát nemcsak a mérőszámok kialakítására, hanem értékelésükre is használják. Statisztikai mérőszámoknak tekinthető például a hierarchikus és a strukturális komplexitás. Az előbbinél az összes modulok, komponensek számát osztják a program szintjeinek számával; a tapasztalati értéke hat és hét között van. Ennél több szintű hívás (modul hív meg modult) ugyanis már olyan mértékű figyelemmegosztást kíván, ami szinte óhatatlanul hibához vezet. A másik, a strukturális komplexitás az egy modulra, komponensre eső hívások számát méri. Értékének p' gramok összehasonlításánál van szerepe.

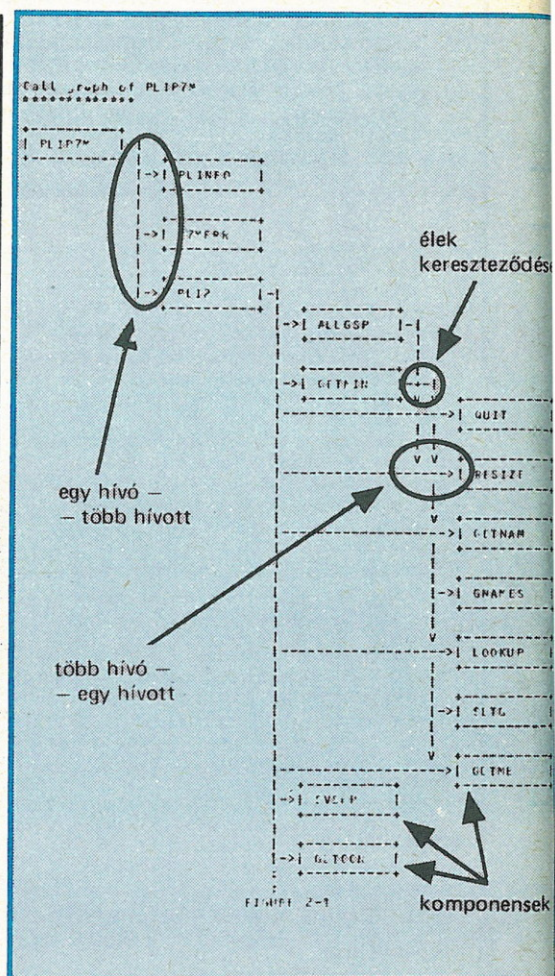
A minőségmérő számok többségének kialakítása, értékük megállapítása során közrejátszik a szakemberek szubjektivitása és tapasztalata. Ezért a mérőszámokat olyan szempontból is vizsgálják, hogy mennyire alkalmasak a program minőségmérésére. Egy egyszerű ilyen vizsgálat, amikor korrelációt számolnak a vizsgált mérőszám értékei és a program javítása, karbantartása által igényelt ráfordítás között. Nyilvánvalóan azok a mutatók jobbak, amelyek szorosabb kapcsolatot fejeznek ki a ráfordításokkal. Azoknak az olvasóinknak, akik jobban el akarnak mélyedni ebben a témában, ajánljuk dr. Szentés János *Szoftverminőség és mérése* c. könyvét (SZÁMALK, 1985.).

A következőkben a QUALIGRAPH által számított mérőszámokat egy-egy konkrét programfutáson keresztül mutatjuk be.

A QUALIGRAPH működési elve

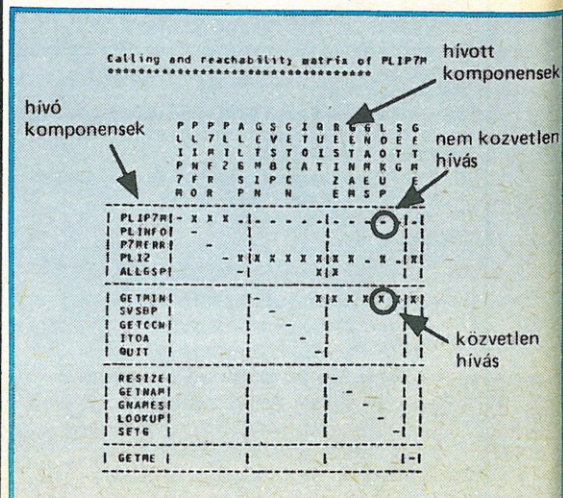
A szoftver a következő programozási nyelveket és azok dialektusait érti: C, COBOL, FORTRAN, Pascal, PDL, PL/1, PL-M/86. Előnyös tulajdonsága, hogy egyaránt futtatható nagygépen és mikroszámítógépen. Alkalmazásához legalább 768 kb-aj központi memória és merevlemez egység szükséges.

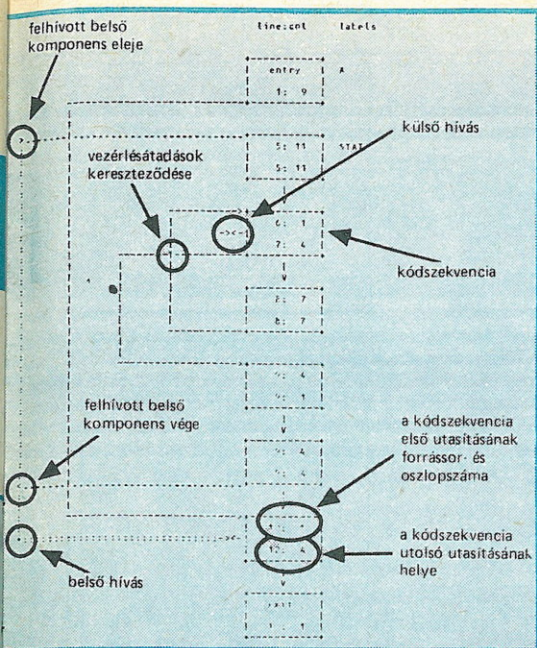
A program indításakor meg kell adni, hogy milyen forrásnyelvet akarunk vizsgálni. Egyéb opciók és paraméterek is megadhatók: például hogy teljes vagy részleges elemzést kérünk-e, vagy az intézeti szabványt. A QUALIGRAPH először minden általa ismert programozási nyelvet értelmez és egy közös nyelvvé alakítja át. Ezáltal egyes nyelven írt programokat és az egysé-



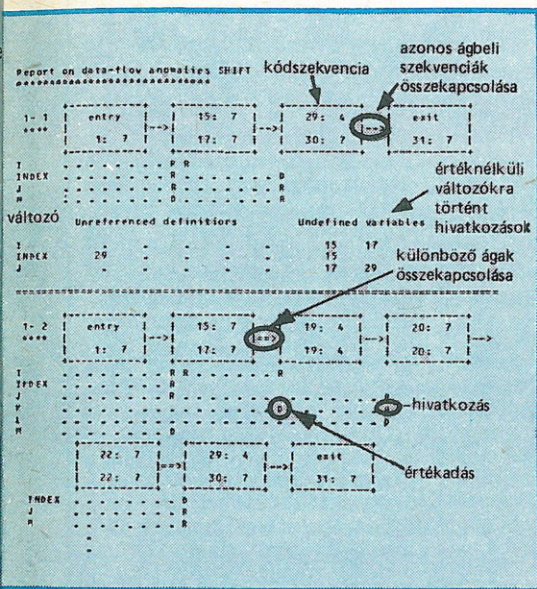
1. ábra

2. ábra





3. ábra



4. ábra

ges dokumentációt is fel tudja dolgozni, mivel az elemző, dokumentáló stb. része minden nyelvnél közös.

A forrásprogramot ún. felülről lefelé (top-down) elemzéssel vizsgálja. Ez azt jelenti, hogy először a program átfogó szerkezetéről gráf és ábra formájában, majd az egyes modulok, komponensek összetételéről és végül egészen programsormélységig kapunk információt.

A legfelső szint a hívási gráf (1. ábra), amely megmutatja, hogy a forrásprogram milyen modulokból, komponensekből áll és milyen a közöttük lévő kapcsolat. Az olvasónak valószínűleg feltűnt, hogy modul helyett általában a komponens kifejezést használjuk. Ennek az az oka, hogy a QUALIGRAPH jelenleg hét programnyelvet tud vizsgálni, és ezeknél különbözőképpen nevezik az alprogramokat: modul, szubrutin, eljárás stb. A komponens kifejezés a legáltalánosabb. Kimenő táblázataiban a QUALIGRAPH is ezt használja. A gráfban, az ábrában lévő téglalapok az adott komponens megnevezését tartalmazzák, az élek iránya és kereszteződései a hívási utakat mutatják meg.

A hívási és elérhetőségi mátrix ugyanazt az információt táblázat formájában közli (2. ábra). A mátrixban a nem közvetlen hívás azt jelöli, amikor az egyik hívott komponens további másik komponensre hív meg. Ez azért nagyon fontos információ, mert ha a fejlesztő belejavít valamelyik komponensbe, akkor tudja, hogy az milyen hatással van a többi programrészre. Ennek különösen az „idegen” program karbantartásánál van nagy jelentősége.

A komponens belsejének felépítéséről ad konkrét képet a vezérlési gráf. Ez már ún. kódszekvencia, utasításszekvencia bontásban mutatja az elágazásokat, a vezérlésátadásokat. A gráf ábrájának téglalapjai a programsorszámokat tartalmazzák, így a fejlesztők közvetlenül az utasítások hatását elemezhetik.

A példákban láthatjuk, mely területeken hasznosítható legjobban a QUALIGRAPH. Új program készítésénél a fejlesztők olyan ábrákat és táblázatokat kapnak, amelyek alapján áttekinthetik „művük” felépítését: a komponensek kapcsolatait, a változók definiálását és a programban betöltött szerepüket. A mérőszámok értékei felhívják a figyelmet bizonyos rendellenességekre, a szoftver tesztelési segédlete pedig könnyíti a programellenőrzést. Végül nem elhanyagolható szempont, hogy a fejlesztők egységes rendszer szerint szerkesztett dokumentációhoz jutnak.

A szoftverkereskedők és -felhasználók sem maradnak ki a hasznosítók sorából, hiszen a QUALIGRAPH-fal mérhetik és összehasonlíthatják a különböző programok minőségét.

Ha egyáltalán fel lehet állítani valamilyen sorrendet, akkor azt mondhatjuk, hogy a QUALIGRAPH leginkább a programok módosításánál, karbantartásánál hasznosítható, elsősorban azok számára, akik hiányosan vagy egyáltalán nem dokumentált szoftverbe akarnak „belenyúlni”, mivel az egyes változások hatását végigkövethetik a teljes programon.

A QUALIGRAPH fontosabb kimenetei

A kimeneteket négy fő csoportba sorolhatjuk: a hívási hierarchia és a rendszerkomponensek belső vezérlésének jellemzése, valamint az adatfolyam- és kommentanalízis.

A hívási hierarchia jellemzésénél említettük már a *hívási gráfot* (1. ábra), amely a rendszerben lévő komponensek közötti kapcsolatot mutatja. Szorosan illeszkedik

hozzá a *hívási utak* kimenet, amelyről a rendszerkomponensek lehetséges végrehajtási sorozatát olvashatjuk le. Szélsőséges esetben találhatunk olyan komponenset, amelynek hívására nem kerülhet sor. Ez nyilvánvaló programhiba.

A *hívási és elérhetőségi mátrixot* és táblázatot szintén említettük már. Ez a kimenet adja talán a legnagyobb segítséget a programkarbantartáshoz; a hívásoknál a hívási hierarchia nyomon követése a legnehezebb feladat, és ez okozza a legtöbb programhibát. Gyakran előfordul, hogy valamelyik komponens közvetlenül első szinten hív a program, de ugyanakkor többszörös áttételen keresztül is rákerül a vezérlés. Ezt követik a komponensek rangsorát jellemző mérőszámok, mint például a hierarchikus és strukturális komplexitás.

A kimenetek következő csoportja a rendszerkomponensek belső vezérlését jellemzi. Ebben a szakaszban már a komponensek belső működését láthatjuk. A csoport első kimenete a *vezérlési gráf*, amelynek felépítését már leírtuk. Ezt követi a *tesztfa* és a *tesztutak* kimenet. A tesztfa a komponens vezérlési szerkezetének kétdimenziós megjelenítése; ez csak segédeszköz a program ellenőrzéséhez. A tesztutak kimenet adja meg, hogyan kell a programot minimális ráfordítással lefuttatni úgy, hogy valamennyi ága legalább egyszer végrehajtsódjon. A kimenet segítségével meg tudjuk határozni a nagyon sok lehetséges végrehajtás közül a legkisebb számút. Biztosak lehetünk abban, hogy nem maradt ki egyetlen programrész sem, ami később esetleg gondot okozna. A kimenetek e csoportjában kapjuk meg a vezérlési szerkezetet jellemző mérőszámokat, mint a ciklomatikusság számát vagy a vezérlési sűrűséget.

A harmadik csoportba az adatfolyam elemzése tartozik. A változók használatáról két táblázatot kapunk: a *változók használatának összesítője* és az *adathasználati anomáliák riportja* elnevezésűt. A táblázatokból kiolvashatjuk, hogy a változó a programban hol kap értéket; van-e feleslegesen definiált változó; milyen továbbgyűrűző hatása van annak, ha egy programváltozó értékét megváltoztatjuk, azaz mely komponensek használják ugyanazt a változót. Nagyon lényeges, hogy a programban ne legyen kezdőérték nélküli változó. Ezt is ellenőrizhetjük a változók riportjából. Adathasználati anomália például, ha egy értékkel rendelkező változónak anélkül, hogy használnánk, újra értéket adunk. Nyilvánvaló, hogy az első értéktadás felesleges volt. Az adathasználati anomáliák riportját a 4. ábra szemlélteti.

Legközelebb befejezzük a kimenetek ismertetését és konkrét példán mutatjuk be a QUALIGRAPH futási „eredményeit”.

PINKE GYÖRGY

Gyalogvégjáték

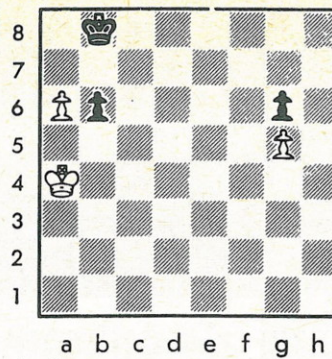
Az eddigiekben olyan gyalogvégjátékokat vizsgáltunk, amelyekben csak egy gyalog szerepelt, és a cél a gyalog bevitele volt. Ez sakk-szempontból nagyon egyszerű eset, programozástechnikailag viszont láthattuk, hogy milyen bonyolult. Ezért a mai napig sem tudunk rá teljesen általános érvényű, egyértelmű megoldást adni. Nagyon sok program a részfeladatok jól értékeli, és számos speciális esetben a lehető legjobb lépést teszi meg, biztosan vezeti figuráit a győzelemre. Most egy-két példán mutatjuk be, milyen matematikai megfontolások alapján lehet a többgyalogos végjátékok értékelését lerövidíteni.

Averbach és Maizelis átfogó könyvet írt Gyalogvégjátékok címmel. Ebben a lehetséges állásokat számos összefüggő fogalom szerint csoportosították. Például kulcsmezők szerint, korrespondáló mezők szerint, mértani és huszárpozíció szerint. Kulcsmezőknek olyan mezőket nevezünk, amelyek közül ha bármelyiket elfoglalja a támadó fél, akkor megnyeri a játszmát. Abban az esetben viszont, ha a védekező fél a kulcsmezők mindegyikét tartásan meg tudja védeni, a játszma döntetlen. Erre nagyon egyszerű példák a KKG (Király a Király plusz Gyalog ellen) végjátékok, ahol a mértani oppozíció az a taktikai eszköz, amivel az erősebbik fél a másikat a kulcsmező elhagyására kényszerítheti. A korrespondáló mezők koncepciója általánosabb, de nem olyan könnyen definiálható.

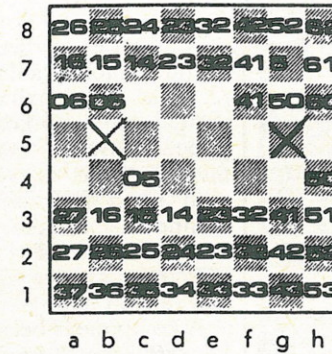
Az 1. ábrán látható példán mindkét fél királyja kedvező helyzetben van. Ha valamelyik király megtámadja az ellenfél gyalogját, úgy a védekező fél rögtön a kulcsmezőjére léphet, és ezzel nemcsak megvédi gyalogját, hanem még a támadó fél gyalogját is lenyeri, és megnyeri a játszmát. Világos kulcsmezője b5, sötété pedig a7. Ha világos lép és mindkét fél a legjobban játszik, akkor 1. Kb4 Ka8, 2. Kc4 Kb8, 3. Kb4 Ka8 után az állás döntetlen. Más változatban az egyik fél nyilvánvalóan elvesztené a játszmát. Ebben az egyszerű esetben nincs kapcsolat más mezők között, azonban egy komplexebb példát mutat Lasker híres tanulmánya (lásd a 2. ábrát).

A kulcsmező a b5 és a g5. Ugyanis ha világos e kettő közül bármelyiket is elfoglalja, behatolhat sötét térfelére és megnyeri a partit. Világos azonban nem tudja közvetlenül megtámadni e kettő közül egyik mezőt sem, mert sötét azt még idejében megvédené, és ezzel megakadályozná a behatolást. Nem vezet például célra a következő lépéssorozat: 1. Kb3 Kb7, 2. Kc4 Kb6 3. Kd3 Kc7 4. Ke3 Kd7 5. Kf3 Ke7 6. Kg3 Kf6 7. Kh4 Kg6. Mindezek ellenére a pozíciót világos megnyerheti, mert lépéskényszerű tud előidézni, és ezzel sötétet a kedvező mező elhagyására kényszeríti a következőképpen: 1. Kb2 Ka7 2. Kb3 Kb7 3. Kc3 Kc7 4. Kd3 és most sötét több folytatás közül választhat, de egyik sem menti a játszmát. Például 4. ... Kd7 5. Kc4 stb., ha 4. ... Kb6 5. Ke3 stb. Ha pedig 2. ... Ka6 lépést tenné sötét, akkor 3. Kc2 nyer.

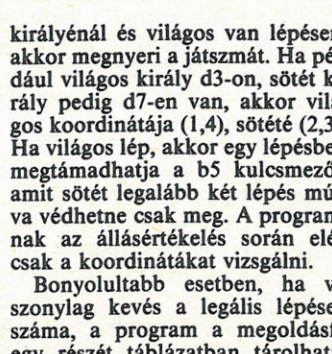
A program rögzített gyalogokkal a nyeréshez vezető utat gyorsan és pontosan kiszámíthatja. Az algoritmus megvizsgálja, hol vannak az állásban áttörési pontok (jelen esetben b5-ön és g5-ön), majd az összes olyan mezőt, amelyikre a király léphet, két koordinátával jelöli. Ezek megmutatják az adott mezőnek az egy vagy két áttörési ponttól való távolságát. Ennek alapján az előbbi hadállásra kiszámított koordinátákat a 3. ábrán láthatjuk. Az első koordináta megmutatja, hogy az adott mező milyen távolságra van a b5 ponttól, a második koordináta pedig a mező és a g5 pont távolságát jelzi. Pontosabban azt, hogy hány lépésben lehet eljutni onnan a királlyal a b5, illetve a g5 mezőig. Ha világos király olyan mezőn áll, hogy akár-melyik koordinátája kisebb a sötét



1. ábra

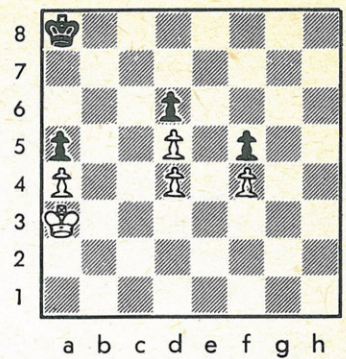


2. ábra

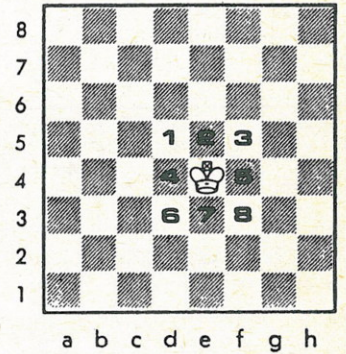


3. ábra

királyénál és világos van lépésben, akkor megnyeri a játszmát. Ha például világos király d3-on, sötét király pedig d7-en van, akkor világos koordinátája (1,4), sötété (2,3). Ha világos lép, akkor egy lépésben megtámadhatja a b5 kulcsmezőt, amit sötét legalább két lépés múlva védhetne csak meg. A programnak az állásértékelés során elég csak a koordinátákat vizsgálni. Bonyolultabb esetben, ha viszonylag kevés a legális lépések száma, a program a megoldásfa egy részét táblázatban tárolhatja (lásd az 1. táblázatot), így azt csak egyszer kell kiszámítani. Az előbbi hadállásra ezt a következőképpen használhatjuk el. Részmegoldásfának tekintjük a játékfának azt a részét, amelyben a világos király adott pozíciójából eljut a két kulcspontra valamelyikéhez. A hadállást tekintve a világos király huszonegy különböző mezőn helyezkedhet el. Tehát egy 26 x 21-es táblázatot kell készíteni. A táblázat elemeinek a száma ezért 26 x 21, azaz 546. A sorok a sötét királypozíciót, az



4. ábra



5. ábra

oszlopok pedig a világos király helyét jelzik. A táblázatban levő kétjegyű számok első számjegye megmutatja, hogy hány lépésben nyer világos, ha ő következik lépésre. A második számjegyből pedig megtudhatjuk, hány lépésben veszít sötét, ha kénytelen lépni. A zérus természetesen a döntetlent jelenti. Ha világos a program, akkor úgy kell játszania, hogy ha lehet, az első számjegy minden lépés megtételekor csökkenjen. Ezzel biztosítja a nyerést.

Olyan pozícióban, ahol világosnak csak egy nyerő lépése van, a program a 2. táblázatot is használhatja, ahol az első számjegy megadja, hogy milyen mélyen van a nyerés (ha nyerhető az állás), a második számjegy pedig megmutatja, hogy a királlyal milyen irányba kell lépni a nyeréshez. A táblázat kitöltésekor a 4. ábrán látható királylépések irányait vetjük figyelembe, a 2. ábrán bemutatott Lasker-tanulmány pozíciójában a világos királyt az a1 mezőre, a sötét királyt az a7 mezőre helyeztük át.

KOVÁCS P. ATTILA

1. táblázat

	a6	b6	f6	g6	h6	a7	b7	c7	d7	e7	f7	g7	h7	a8	b8	c8	d8	e8	f8	g8	h8
a1	60	60	14	11	11	80	0	80	0	40	14	11	11	60	0	60	0	40	14	11	11
b1	40	40	14	11	11	7	70	7	50	45	14	11	11	60	50	60	45	14	11	11	11
c1	10	30	14	11	11	10	6	60	56	45	14	11	11	10	50	40	56	45	14	11	11
d1	11	10	14	11	11	11	10	5	50	45	14	11	11	11	10	40	50	45	14	11	11
e1	11	11	14	11	11	11	11	10	0	40	14	11	11	11	11	10	0	40	14	11	11
f1	11	11	10	11	11	11	11	10	0	0	10	11	11	11	11	10	0	0	10	11	11
g1	11	11	0	10	11	11	11	10	0	0	0	10	11	11	11	10	0	0	0	10	11
h1	11	11	0	0	10	11	11	10	0	0	0	0	10	11	11	10	0	0	0	0	10
a2	60	60	11	11	11	50	0	40	0	10	11	11	11	60	0	60	0	10	11	11	11
b2	45	45	11	11	11	50	30	40	30	13	11	11	11	5	50	5	30	13	11	11	11
c2	13	33	11	11	11	10	30	20	34	13	11	11	11	10	4	40	34	13	11	11	11
d2	11	12	11	11	11	11	10	20	30	13	11	11	11	11	10	3	30	13	11	11	11
e2	11	11	11	11	11	11	10	10	0	10	11	11	11	11	10	0	10	11	11	11	11
f2	11	11	10	11	11	11	11	10	0	0	10	11	11	11	11	10	0	10	11	11	11
g2	11	11	0	10	11	11	11	10	0	0	10	11	11	11	11	10	0	0	10	11	11
h2	11	11	0	0	10	11	11	10	0	0	0	10	11	11	11	10	0	0	0	10	11
a3	60	60	11	11	11	50	0	40	0	10	11	11	11	60	0	60	0	10	11	11	11
b3	40	20	11	11	11	4	30	3	10	11	11	11	11	10	10	10	11	11	11	11	11
c3	10	20	11	11	11	10	2	20	12	11	11	11	11	10	10	10	12	11	11	11	11
d3	11	10	11	11	11	11	10	1	10	11	11	11	11	11	10	10	10	11	11	11	11
e3	11	11	11	11	11	11	10	0	10	11	11	11	11	11	10	0	10	11	11	11	11
f3	11	11	10	11	11	11	11	10	0	0	10	11	11	11	11	10	0	10	11	11	11
g3	11	11	0	10	11	11	11	11	10	0	0	10	11	11	11	11	10	10	10	11	11
h3	11	11	0	20	10	11	11	11	10	0	0	10	11	11	11	11	10	10	10	10	11
c4	10	1	11	11	11	10	10	11	11	11	11	11	11	11	11	11	11	11	11	11	11
h4	11	11	10	1	10	11	11	11	11	11	10	10	10	11	11	11	11	11	11	11	11

2. táblázat

	a6	b6	f6	g6	h6	a7	b7	c7	d7	e7	f7	g7	h7	a8	b8	c8	d8	e8	f8	g8	h8
a1	63	63	0	0	0	85	0	85	0	43	0	0	0	63	0	63	0	43	0	0	0
b1	43	43	0	0	0	75	0	75	0	53	0	0	0	62	53	62	53	0	0	0	0
c1	0	33	0	0	0	0	0	65	52	0	0	0	0	52	43	52	0	0	0	0	0
d1	0	0	0	0	0	0	0	0	51	0	0	0	0	0	0	42	51	0	0	0	0
e1	0	0	0	0	0	0	0	0	0	41	0	0	0	0	0	0	0	0	41	0	0
f1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
g1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a2	65	65	0	0	0	53	0	43	0	0	0	0	0	65	0	65	0	0	0	0	0
b2	45	45	0	0	0	52	33	42	33	0	0	0	0	55	0	33	0	0	0	0	0
c2	0	35	0	0	0	0	32	23	32	0	0	0	0	0	0	45	32	0	0	0	0
d2	0	0	0	0	0	0	0	22	31	0	0	0	0	0	0	0	31	0	0	0	0
e2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
g2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a3	68	68	0	0	0	55	0	45	0	0	0	0	0	68	0	68	0	0	0	0	0
b3	48	23	0	0	0	0	35	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c3	0	22	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0
d3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
g3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h3	0	0	0	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Szokásunkkal ellentétben ezúttal olyan hosszabb olvasói leveleket közlünk, amelyeket általában nem szoktak kitenni a kirakatba.

Az első nekem szól:

Mészáros Gyula, Budapest

Rendszeres olvasója vagyok a Magazinnak, minden számban találok két-három jó cikket. Az utóbbi időben azonban kissé megrendült a bizalmam Önben és a lapban is. Néha az az érzésem, hogy a munkatársak nem beszélnek egymással és/vagy nem olvassák a lapot. A legjellemzőbb példa erre az idei első szám 46. és 47. oldala. Érdekes egybevetni Göbölös Lászlónak írt válaszát a 46. oldalon és a 47. oldal alján található hirdetéset az Enterprise klubról. Enterprise-zal foglalkozom már egy ideje, így ennek a gépnek a „sajtóját” meg merem ítélni.

A ZX-Spectrum emulátorprogram létezik, nekem is megvan. Szívesen el is küldeném ingyen bárkinek, ha tudnám, hogy ezzel nem vétek a szerzői jog ellen. A program 16 k hosszú, és élet-hűen emulálja a Spectrumot.

A következők olvashatók Káli Lászlónak írt válasza végén (88/1. szám): „... úgy nézett rám, mintha egy Enterprise MS-DOS-t kértem volna.” Akkor lehet, hogy valaminek megelőztük az NSZK-t! Ithon ugyanis létezik az ún. IS-DOS, amely parancs és lemezformátum szinten az MS-DOS-szal, felépítését tekintve a CP/M-mel kompatibilis. Ha jól tudom, a Microteam is gyárt lemezegység-vezérlőt.

„Egy diákversenyéről, nagyon komolyan” című cikkére is szeretnék reagálni. A versenybeszámolót nem lehetett meghatottság nélkül olvasni. Élő, megoldásra váró problémát vet fel a vakok és gyengénlátók számítástechnikai képzése és munkába állása kapcsán. Teljesen egyetértek azazal, hogy az otthoni munkahely lenne az igazi.

Nem szeretném, ha a bizalma megrendült volna a lapban, hiszen azt sokan szerkesztjük, legfeljebb bennem, ha arra valóban rászolgáltam.

Az IS-DOS valóban megvásárolható a CENTRUM ÁRUHÁZAKBAN. A Spectrum emulátor gyártását a VIDEOTON Elektronikai leányvállalata megkezdte, az emulátor nemcsak szoftver, hanem hardver elemeket is tartalmaz.

Romvári Gábor, az Enterprise klub vezetője egyébként szívesen látja a szoftverlehetőségek iránt érdeklődő Enterprise-tulajdonosokat a klubban (telefonja: 810-950/473-as mellék).

Ami a vak programozók ügyét illeti, sajnos a jelzett cikkre éppen az érintettek, a vállalatok, a kutató- és fejlesztőintézetek nem reagáltak. A problémát akkor is meg kell oldani, például fel lehetne élénkíteni az ilyen típusú beruházás iránti kedvet adókedvezményekkel, de ebben az esetben az erőszakos állami beavatkozást sem tartanám elfogadhatatlannak. Úgy hallottam, hogy például a szomszédos Ausztriában is államilag előírták, egy meghatározott létszámú vállalatnak hány vak embert kell foglalkoztatnia.

A következő íráshoz csak annyit előljáróban, hogy nem egyforma az izlésünk.

Kálmán János, Budapest

Négy év óta többé-kevésbé rendszeresen olvasom folyóiratukat. Sok segítséget kaptam a Mikromagazintól, kezdve a BASIC első lépésein, egészen a Z80 assembly programozásig. Éppen ezért sajnálom, hogy a korrekt tudományos ismeretterjesztést mindinkább fenyegető, Kávéházi Konrád stílusú zsrnál-tudományoskodás újabb folyóiratukban is helyet kér, és — sajnos — kap.

A színvonalstüllyedés iskolapéldája Szulyovszky Csaba könyvismertetője a MESZ Számítástechnika Mikrokalausz c. kiadványáról. A pár flekkes írás szerzője túl szűknek érzi a recenzio valóban hálátlan műfaját: a laza stílusban előadott, virtuóz Nagy Elmélet babérjaira vágyik. Munkáját egy rendkívül jelentős, ám kevésbé plauzibilis fizikátörténeti téma felvázolásával kezdi: felvilágosít arról, hogy ha Archimédész annak idején nem fürdék meg, bizony jó 2000 esztendősz késésben lenne a fizika.

Letarolván a tudománytörténet fő kérdéseit, épp ideje, hogy a pontos, szinte időtlen zsenialitás szerzőnket a filozófia kanyargós ösvényeire vezesse. Itt is hamar rendet teremt. A megoldás — állítólag — egy ókori filozófus egy mondatában rejlene: „Minden tudás emlékezés csupán”.

Jómagam, reménytelenül és örökre laikus lévén csak kullogok a recenzens szárnyaló zsenialitása mögött, de szavai, mondatai között az összszefüggést föl nem lehetem.

Örkény írta Az élet értelme c. egypercesében: „Ha sok cseresznyepaprikát madzagra fűzünk, abból lesz a paprikakoszorú. Ha viszont nem fűzzük fel őket, nem lesz belőlük koszorú. Pedig a paprika ugyanannyi, éppoly piros, éppoly erős. De mégse koszorú. Csak a madzag tenné? Nem a madzag teszi. Hát akkor mi? Aki ezen elgondolkozik és ügyel rá, hogy gondolatai ne kalandozzanak összevissza, hanem helyes irányba haladjanak, nagy igazságoknak jöhet a nyomára.”

Szulyovszky írásában találhatunk információt az ismertetni kívánt kiadványról is, főként idézetek alakjában. Ha Örkényre hallgatva megpróbálta volna összefűzni ezeket, recenzio született volna. Így csak papír és betűk halmaza — bár utóbbi is nyomdatermék.

Szulyovszky Csaba válasza:

Tisztelt Kálmán János! Noha írásomat kritikával — még hozzá negatívval — fogadta, örömmel vettem a megkeresését. Szellemes, helyenként sziporkázó bírálata komoly kultúrhistorikus tájékozottságról tesz tanúbizonyságot. Ugyan „a színvonalstüllyedés iskolapéldájának” titulálta könyvismertetőmet, előre is tisztázni kívánom, hogy nem recenziozt írtam, sőt nem is könyvismertetőt. Szándékom — s hiszem, hogy ez sikerült is — és céloim mindössze annyi volt, hogy némiképp bepillantást nyújtsak a nagy lélegzetű, hatalmas területeket felölelő Mikrokalausz sorozataiba. A „Kávéházi Konrád stílusú zsrnál-tudományoskodás” távol áll tőlem meg a publikált cikktől is. Az ugyanis nem tudományos volt, hanem tájékoztató arról — kivételesen kerülve a megszokott szárazságot —, hogy létezik a számítástechnika világában egy kiadvány, amit csak kevesen ismernek.

Úgy tűnik soraiból, nehezényezi, hogy stílusommal szakítottam a havilap bevált „szóvirágai-

val”. Ezt bátran vállalom. Nem hiszem ugyanis, hogy egy 48 oldalas sajtótermék minden mondatának azonos stílusúnak kell lennie, s azt sem hiszem, hogy egy kissé más íz, más szín — nem állítom, hogy jobb mint a többi — értelmére van egy újságnak. A másság még önmagában nem negatívum.

Amin csodálkoztam azonban, hogy éles szemével, kutató alaposágával egy kedvcsináló írást összekever egy recenzioval. Mindenesetre élveztem stílusát, kritikus, szellemes hangvételtét, s hadd tegyek egy ajánlatot: ha szerzőink közé állna, írásival emelné a lap színvonalát. Üdvözlettel Szulyovszky Csaba, aki végül is nem Kávéházi Konrád.

Négyesi Károly és Négyesi Pál, Budapest

Kérem, közölje le ezt a levelet. Én nagyon szerettem ezt a lapot. Szerintem a legjobb magyar számítógépes szaklap. Sokoldalú és rendszeresen jelenik meg (ezért jobb a Spectrum világnál). Dehát, ahogy semmi se lehet tökéletes, úgy a Magazin sem az. Ön azt írja, hogy friss festékszallalaggyal nyomtassunk. Ez azt jelenti, hogy 20-30 programként új festékszallagot kell venni? Tudom, hogy ez a 20-30 program sok, de ha valaki rendszeresen és egy alkalommal nem egy programot küld? És ha nincs nyomtatóm, akkor hogy küldjek be programokat?

Sajnos a lapban csak szépen és világosan nyomtatott programokat tudunk közölni. Ha tehát azt akarja, hogy a programjai megjelenjenek, akkor gondoskodnia kell a megfelelő nyomtatóról! A programokat nem tudjuk hibamentesen kiszedni sem.

A következő levelet a programmásolással kapcsolatban kaptam. Egy kicsit rövidítve közreadom:

Békés Sz. András jelige

Több mint négy éve vásároltam gépet, pénzt áldoztam akkor arra is, hogy 80 k-s legyen.

De elég a nosztalgiából, maradjunk az egyszerű matematikánál. Számolok, kérem, ellenőrizzenek. 970 felhasználói- és 1480 játékprogramot archiváltam a mai napig 130 kazettán, cseréhez 40 darabot mozgatok, 20 db megsérült, más csócsálós magnója miatt. A 130 kazetta (60-as 90-es vegyesen) legalább 310 órát igényel, hogy tele legyen programmal, 30 Ft/óra országos átlag órabért, 60 Ft/kazetta egységárat és 2600 Ft számítógép- és tévéjavítási díjat figyelembe véve, négy év alatt ez — várjunk csak! Az utóbbi két év forgalma alapján kitüntetésben akar részesíteni a helyi postahivatalt! Mennyi is ez? Összesen a felsorolt 26 300 forint, havi 550 forint (a közszépzből), ez több mint 10 forint egy archivált programra.

Valóban vannak elítélendő esetek: Enterprise-program másolva 100 forint darabonként! Ezt nem meri hirdetni a Magazin! Ki mondta, hogy a C64-es gyűjtőtől venni KELL, hálapénzt, borralót KELL adni bárkinek? Tegyük őket tönkre, ne vegyünk tőlük! CSERÉLJÜNK, de ne csak EGYOLDALÚAN! Úgy érzem, elfogadható, hogy ha valakit DOLGOZTATUNK, annak MUNKÁJÁT honoráljuk.

Leveleiket — még a kritikusokat is — örömmel várja

KOVÁCS GYŐZŐ

A TII ajánlja

Zenei oktatóprogram-csomag C64 és C128 számítógépre

A programcsomag feladata, hogy első-sorban tanórán kívüli alkalmazás során tanulónak és pedagógusnak segítségére legyen.

Az öt program az alapfokú zeneoktatás legfontosabb elemeit tartalmazza úgy, hogy ezek összefüggéseikben is megmutatkoznak, de a gyakorlásnál túl zenei fantáziája, kreatív készsége is fejlődhet a tanulónak.

A programok lényege, hogy hangzó vagy vizuális feladatot adnak, a tanuló erre válaszol, s a program ennek megfelelően folytatódik tovább. Ebben az értelemben egy tanárt modellez a program, de mégsem a tanár helyettesítéséről van szó, hanem arról, hogy amit munkánk során ötször, tízszer, százszor kell elismételni, azt miért ne tegye a program. A tanárnak így is lesz bőven tennivalója, és több idő marad arra, hogy az órán az élő zenével foglalkozzon.

A programcsomag elemei az alábbiak:

"DALLAM"

Három részből áll:

- a program dallamának leéneklése
- saját dallam begépelése
- dallamírás

"HANGKÖZ"

A hangközök szolmizációját segíti ötféle nehézségi fokú gyakorlásban.

"AKKORD"

Ez a program a hangzatfelismerést segíti.

"KOMBI"

Használata fejleszti a hangsorokban való jártasságot, a szolmizációs készséget, a zenei memorizálást, a formaérzékenységet ill. ritmusfantáziát.

"MOZART"

A program a zeneoktatás egyik legproblematikusabb területével, a funkcióérzék fejlesztésével foglalkozik. Alapjául Mozart játékos dallamszerkesztési módszere szolgál.

Ára mágneslemezen: 1500 Ft + ÁFA
kazzettán: 1300 Ft + ÁFA

Szolmizációs gyakorlatok Ének-zenei oktatóprogram Commodore 64 számítógépre

A "Solmizációs gyakorlatok" programcsomag alapvető segítséget nyújt a szolmizációs készség fejlesztéséhez. Lehetőséget biztosít a kotta írás-olvasás megtanulására, eközben megismerteti és gyakoroltatja a pentatonikus, diatonikus, alapfokú kromatikus hangkészletet. Négy

különböző nehézségi fokon gyakoroltatja a kottaképről, illetve a hallás utáni szolmizálást. A feladatok végrehajtását ellenőrzi és értékeli. A szolmizációs gyakorlatok alapját négyszáz különböző dallam képezi.

A program azoknak a tanulóknak készült, akik a zenei alapfogalmakkal vagy a számítógéppel csak most ismerkednek. Megtanítja a szolmizációs hangoknak megfelelő billentyűk használatát, megismerteti a hangok előjegyzésektől függő helyét a vonalrendszeren.

A programcsomag kihasználva a Commodore 64 zenei és grafikus lehetőségeit, kiválóan használható a szolmizációs készség játékos formában történő fejlesztésére. Ezért ajánljuk minden általános és középiskolának, szakiskolának, zenei szakkörnek, továbbá minden zene iránt érdeklődő tanulónak otthoni tanulásra.

A program biztosítja a nehézségi fok (szint, tagolás, tempó), a képernyőszín és a megszólaló hangszer kiválasztását. A következő menüből lehet választani:

Programbeállítás

- 1.1 nehézségi fok megadása
- 1.2 szín beállítása
- 1.3 hangszer kiválasztása

Hangszer kiválasztása

Meghatározható, hogy a gyakorlatok milyen hangszeren szólaljanak meg:

- C64 Klarinét
 - furulya harmonika
 - csembaló trombita
 - xilofon zongora
- alapbeállítás: C64 szintetizátor

Szolmizálás hallás után

A hallásfejlesztés szempontjából ez a programcsomag legfontosabb gyakorlata. A tanuló a lejátszott dallam (dallamrészlet) alapján gyakorolhatja a hallás utáni szolmizálást, úgy, hogy a hallott dallamot újra énekelve a szolmizációs jelek begépelésével állítja elő a dal kottaképet.

- A használatához szükséges konfiguráció:
- Commodore 64 alapgép
 - TV vagy monitor
 - VC - 1541 lemezegység

Ára: 1120 Ft + ÁFA

Grafikus fejlesztő eszközök demonstrációja Graphics Development Toolkit demo-VOIDMO

Számítógép típus: IBM XT, IBM AT
Operációs rendszer: Microsoft MS-DOS
V 3.20

A VOIDMO bemutatja az IBM Graphics Development Toolkit használatát, grafi-

kus lehetőségeit. A VOIDMO-t FORTRAN nyelven írták, a forrásprogram a lemezen megtalálható. A forrás és a VOIDMO együttesen segíti a Toolkit használatának megtanulását.

A lemezhez mellékelünk magyar nyelvű (BASIC, FORTRAN és PASCAL) nyelvi füzeteket.

A grafikus megjelenítés az IBM Graphics Development Toolkit V 1.0 verziójára épül.

Ára: 10 000 Ft + ÁFA

Post Processor Program - PPP

Számítógép típus: IBM XT, IBM AT
Operációs rendszer: Microsoft MS-DOS
V 3.20

A PPP program lehetővé teszi, hogy úgynevezett EGL alfanumerikus grafikus leíró nyelven írt fájlokat képernyőre felrajzoljunk, részleteket kinagyítsunk, és a kijelölt rajrzészetet tetszőleges, az IBM Graphics Development Toolkit által támogatott grafikus nyomtatón vagy rajzológépen kirajzoltassuk.

(A PPP programmal közvetlenül kirajzolható a GSP Szubrutinkönyvtárral előállított EGL fájlok tartalma.)

A grafikus megjelenítés az IBM Graphics Development Toolkit V 1.0 verziójára épül (így a megjelenítés eszközfüggetlen).

Ára: 30 000 Ft + ÁFA

Grafikus alprogramcsomag Graphical Subroutine Package-GSP

Számítógép típus: IBM XT, IBM AT
Operációs rendszer: Microsoft MS-DOS
V 3.20

A GSP egy grafikus szubrutinkönyvtár, amely lehetővé teszi interaktív grafikus alkalmazói programok kifejlesztését, az interaktív grafika oktatását stb. A GSP egyaránt használható Microsoft C, FORTRAN és PASCAL nyelven (akár keverten írt) programokból.

Grafikus adatok tárolása
VDI Metafile-ban
EGL Alfanumerikus grafikus leíró nyelven.

A grafikus megjelenítés az IBM Graphics Development Toolkit V 1.0 verziójára épül. (Így a megjelenítés eszközfüggetlen.)

Ára: 30 000 Ft + ÁFA

Forgalmazza: Tudományszervezési és Informatikai Intézet,
Budapest, Pf.: 454. 1372

FELHÍVÁS

Intézetünk 1986-ban pályázatot hirdetett a közoktatási intézményekben használható számítógépes oktatóprogramok készítésére.

A pályázatra beadott oktatóprogramok közül azokat - bemutatást segítő, tanórán és szakkörön használható magas színvonalú programokat intézetünk forgalmazására átvette.

Ismételt felhívással fordulunk a közoktatási intézményekben használható oktatóprogramok készítőihez.

A felhívás tárgya az általános és középiskolákban elterjedt mikroszámítógép típusokra (Commodore 64, 16 és Plus/4, Videoton TVC) készített magas színvonalú, újszerű oktatási program, amely illeszkedik az oktatási intézmények tananyagához, munkájához és/vagy lehetővé teszi a mikroszámítógépek alkalmazásának bemutatását, és amelyhez hasonló a forgalomban lévő oktatási programok között nem található.

Jelentkezni lehet az alábbi típusú programokkal:

- oktatást segítő
- a számítástechnikai oktatást segítő
- demonstrációs
- szimulációs

Intézetünk csak a szerző(k) szellemi tulajdonát képező, új, forgalomba még nem hozott önálló programot fogad el.

A forgalmazásra alkalmas programokra bizományos szerződést kötünk a szerzővel. Ezen szerződésben a felek meghatározzák a program eladási árát, amelyért intézetünk a programot értékesíteni kívánja és az árbevétel megoszlását a szerző és az értékesítést végző intézetünk között.

TUDOMÁNYSZERVEZÉSI ÉS INFORMATIKAI
INTÉZET
POSTACÍM: BUDAPEST, PF.: 454. 1372

A TUDOMÁNYSZERVEZÉSI ÉS INFORMATIKAI INTÉZET

előzetes megbeszélés szerint díjmentes programbemutatót tart (vidéken is) az általa forgalmazott oktatóprogramokból.

Horváth Zsuzsa 665-011/2663 mellék
vagy 813-197

Budapest, Pf.: 454. 1372

Csillagok, csillagok...

A Tudományszervezési és Informatikai Intézet programajánlataiban tallózva három érdekes programra figyeltünk fel. Témájuk igazán kuriózum — csillagászat. A C64-es változat az oktatási programok között leginkább „csillagászati”-nak mondható árával tűnt ki (3310 forint, amelynél magasabb ár nincs is a katalógusokban), de ez nem közoktatási célokra készült. Végül inkább a két „földközeli” Plus/4-es programot néztük meg, mert kíváncsiak voltunk, hogy mit láthatnak az általános iskolások a két program birtokában.

Nos, annál biztosan többet, mint amit a földrajz tankönyvből megtudhatnak. A 8. osztályos földrajz-könyv sorra veszi az évszakokat, a zónaidőket, a Földet és bolygótársait, a Holdat fényváltozásaival, és még a Napra is bőven jut ereje. A csillagokra és a csillagrendszerekre azonban már csak néhány oldal marad. A csillagképeket mindössze az utolsó lap csillagtérképe mutatja — minden kommentár nélkül.

A kommentárról a stílusos nevű Göncöl Társaság gondoskodik, az említett két Plus/4-es programmal. A két program annyira összefügg, hogy csak együtt érdemes róluk beszélni. A programok adatait a táblázatban foglaltuk össze.

Forgalmazó:	Tudományszervezési és Informatikai Intézet
Terméknevek:	Forgó csillagok, ill. Csillagképek
Szerzők:	Polgár László, ill. Bartus Ferenc, a Göncöl Társaság tagjai
Béptípus:	Plus/4
Hordozó:	kazetta és floppy
Dokumentációk:	hiányosak, nem elég rendezettek
Árak:	440/560 forint, ill. 500/630 forint (kazetta/floppy-árak!)

A két program közül — ahogyan az árból is kiderül — a „Csillagképek” teljesítménye a nagyobb. Az elemzést az egyszerűbbel, a „Forgó csillagok”-kal kezdtük, amely a csillagos égbolt forgatását mutatja be. Rengeteg variációs lehetőségünk van. Akár percről percre végigmehetünk egy napon, de akár a teljes éven is. Ugyanakkor kiválaszthatjuk, hogy a Föld melyik koordinátákkal jellemzett helyéről szeretnénk nézni az égboltot. A koordinátát kijelölhetjük úgy is, hogy a világ nagyvárosai közül választunk (Kairó, Tokió, Lisszabon, New York, Róma, Budapest, Bécs, Párizs, Lon-

don, Varsó, Berlin, Moszkva, Stockholm, Oslo, Murmansk közül válogathatunk).

A képernyő kör alakú csillagtérképén árnyékoltan látszik az a rész, amelynek csillagai nem észlelhetők. A látható égbolt az ábrázolás tulajdonságai folytán ellipszis alakú. A csillagképek apró vonalkák, és eléggé torzítottak ahhoz képest, ahogy az égen eredetiben látjuk őket. Hasznos lehetőség, hogy a program kérésre kinagyítja a látható térséget a teljes képernyőre (zoom). Nagyon hiányzik azonban egy automatikus forgathatóság, hogy az eseményeket dinamikus is megfigyelhessük.

A programot hamar megunhatjuk, ha nem ismerjük a csillagképeket. Ezek megismerése után viszont feltétlen érdemes vissza-visszatérni hozzá, mert pontosan csak ezzel a programmal láthatjuk az évszakok hatását.

A „Csillagképek” program leírásából minden bizonnyal hiányzik valami kezelési útmutató, mert többször is elakadtunk a programmal. Érezhetően várt valamire, de nem tudtuk kitalálni mire, ezért újra kellett indítani (a floppys változatot vizsgáltuk). További érdekesség, hogy a STOP gombot nem tiltja le. Ennek hátrányáról később szólnunk.

A „Csillagképek” program rengeteg modulból áll. Ennek oka az, hogy a szerző a csillagképeket csoportokba osztotta, és az egyes csoportokat az angol ábécé betűivel kódolta. A-tól Z-ig huszonhat csoport van. A csoportokban az egy-két csillagképen kívül nevezetes csillagászati objektumok is szerepelnek. Ilyen objektumok a legfényesebb csillagok, kettőscsillagok, ködök, extragalaxisok stb. Fontos, hogy alapbeállításban a csillagképek nincsenek vonalakkal összekötve. Ez csak külön kérésre jön létre. Ugyanakkor külön kérésre jelennek meg a feliratok és a különleges objektumok jelei. A képi ábrázolás megpróbálja követni a csillagok fényességét is, de ez valószínűleg a monitorok hibái miatt nem eléggé érzékelhető. Az egyes csillagképek felismerése jól gyakorolható. A kép alján rövid magyarázat is kérhető, ami hathatós segítség, nem kell mindjárt könyvek után nézni.

Ha elég sokáig ismerkedünk a csillagképekkel, akkor igen tetemes ismeret birtokába juthatunk. Tudásunkat ellenőrizhetjük a program ASTRO TOTO-jával, amely száz kérdésből ad fel kérdéseket véletlenszerű sorrendben. A vájtűlűek ilyenkor segíthetnének magukon a STOP gombbal, és puszkázhatnának a program adataiból, ha en-

nek lenne értelme. A program ugyanis helytelen válasz esetén azonnal közli a helyeset, és ráadásul semmilyen statisztikát nem készít. Ha versenyezni akarunk, akkor bizony magunk strigulázhatjuk a helyes válaszokat a bölcs számítógép mellett. Ez apró hiba, amelyen könnyű segíteni.

A programok képábrázolása eléggé szenved a képfelbontás korlátaitól, de vannak elég jól sikerült, szép képek is. Főleg a teljes csillagtérkép ábrázolása nem éppen lenyűgöző, de ezen valamelyest segít a kinagyító funkció. A „Csillagképek” összefoglaló ábráján az évszakok is láthatók. Az évszakos jelenségeket mégis inkább a „Forgó csillagok”-kal érzékelhetjük jobban. A két programot tehát csak együtt érdemes megvenni!

Mindkét program semmibe veszi a déli félgömb csillagképeit. Kár! Pedig szívesen néznénk Buenos Aires, Melbourne stb. csillagos egét is, legalább számítógépen. Reméljük, hogy a Göncöl Társaságban erre is lesz még energia. Persze már ezért a két programért is nagy köszönet jár nekik.

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

A programpárost összességében jónak ítéljük, de kellett volna még egy kicsit csiszolni rajtuk.

A programok nemcsak általános iskolában és csupán szakköri foglalkozáson, hanem a középiskolai oktatásban és az önképzésben, valamint amatőr csillagászok képzésére is használhatók. Nem árt megemlíteni, hogy a profi csillagászok igenis sok mindenben számítanak az amatőrök munkájára, tehát tanításuk, tudásuk ápolása nagyon fontos.

ZSADÁNYI PÁL—ifj. ZSADÁNYI PÁL *Visszatérő:* A Tudományszervezési és Informatikai Intézetben tájékoztattak bennünket arról, hogy a korábban elemzett „Ábrázolás” programcsomag dokumentációját kijavították, és a programcsomag még egy további programmal is bővült. Őszintén örülünk a fejleményeknek. (A szerk.)

Kikuchi, Makoto:
Japán csoda — japán szemmel
 (Budapest, 1987. Műszaki Könyvkiadó, 160 oldal. Ára: 44,— Ft)

A könyv bemutatja a japán ipar növekedését az elektronika térhódításától kezdve napjainkig. Olvashatunk a fejlett elektronikai világ háborújáról, a társadalom egyes rétegeinek magatartásbeli változásáról. A Sony cég igazgatója hitelesen mutatja be a „japán csoda” történetét, a robbanásszerű fejlődés megdöbbentő tényeivel alátámasztva.

Aki a könyvet elolvassa, egy kicsit rá fog hangolódni a japán gondolkodásmódra. A nagy kiugrás mögötti történelmi, antropológiai, pszichológiai, szociológiai háttérnek a műszaki, gazdasági és politikai összefüggéseivel is megismerkedhet. Filozófiai mélységei mellett is kellemes és szórakoztató olvasmány műszaki szakembereknek, közgazdászoknak és a korszerű elektronika eredményei iránt érdeklődőknek egyaránt.

Lipovszki György:
PC FORTH
 (Budapest, 1987. LSI ATSZ, 133 oldal. Ára: 162,— Ft)

A könyv a Laboratory Microsystem fig-FORTH alapú, IBM PC-re készült PC/FORTH rendszerét ismerteti.

A PC/FORTH a FORTH fig-FORTH szabványán alapul. A leírás kitér a PC/FORTH és a fig-FORTH eltéréseire is.

A könyv a gép bekapcsolása utáni azonnali szükséges tudnivalók (rendszer- és programindítás, a FORTH-ból való kilépés, alapvető screenkezelés) leírása után a (fix vagy cserélhető) lemezen lévő rendszerfájlokat, a PC-DOS és a PC/FORTH kapcsolatát, az alapvető utasításszavakat, majd a szövegszerkesztési üzemmódot ismerteti.

Részletesen bemutatja a 8086-os assembler PC/FORTH-beli használatát. Külön fejezetben tárgyalja a belső felépítésű információkat, a fix paramétereket, a CPU regiszterek felhasználását, a memóriafelosztást, a fontosabb utasításszavak szótári formáját, az állapotbájt felépítését és az utasításnevek kezelését.

Végül néhány PC/FORH-ban lévő programrendszert közöl.

Peckham, Herbert:
BASIC nyelvű programozás az IBM PC-n
 (Budapest, 1987. McGraw-Hill—Novotrade, 331 oldal. Ára: 219,— Ft)

Ez a könyv H. Peckham BASIC: A Hands-on Method címmel megjelent könyvének átdolgozása. Az eredeti kiadás a különböző, párbeszédész üzemmódban működő számítógépeken alkalmazott BASIC nyelvel ismertette meg az olvasót. Ez a kiadás a korábbit dolgozza fel, speciálisan az IBM személyi számítógépen való alkalmazásokról.

A létező legtöbb BASIC tankönyvnek két, nem túl kellemes jellemzője van. Az egyik az, hogy az olvasókkal szemben már igen gyorsan magas matematikai igényeket támasztanak, és emiatt az olvasókörük beszűkül. Az olvasók többsége ugyan bizonyára ismeri a számtani alapfogalmakat, de aligha valószínű, hogy ezek BASIC-ben való programozását tanulta volna. A másik ok a szerkezetükben keresendő. Az egyes részek olyannyira összefolynak, hogy az olvasónak alig teszik lehetővé, hogy a tanulás során sok — vagy akár csak valamennyi — időt a számítógéppel tölthesse.

Az olvasók a programozás tanuláshoz is úgy kezdenek hozzá, mint minden más feladathoz, és nem veszik észre annak szükségességét, hogy kísérletezzenek a számítógéppel, és programokat futtassanak rajta. Ennek a könyvnek az a legfontosabb célja, hogy a tankönyvek szokásos szöveges részével ellentétben számítógépes nyelven, a számítógépes kísérletezést részesítse előnyben.

Moto-oka, T.—Kitsuregawa, M.:
Az ötödik generációs számítógép
A japán kihívás
 (Budapest, 1987. Műszaki Könyvkiadó, 67 oldal. Ára: 28,— Ft)

Az ötödik generációs számítógépek létrehozásának programja Japánban és külföldön rendkívül élénk érdeklődést váltott ki, és igen biztató, hogy az USA-ban és Európában még a vártnál is nagyobb visszhangot keltett.

Az ötödik generációs számítógép létrehozásának programja új távlatokat nyitott, a robottechnikaival és a mikroelektronikával együtt ez az első lépések egyike a csúcstechnológia korszakában.

Új alkalmazási területekre kíván betörni; ilyenek például a tudást feldolgozó és a mesterséges intelligenciát alkalmazó rendszerek. Ezenkívül gyökeresen meg akarja változtatni a „felhasználói interfész”, vagyis az ember és a számítógép kapcsolatát kívánja kezelhetőbbé tenni. Az a célja, hogy a különböző technológiák eredményeit — a VLSI technológiát, a számítógép-architektúrát, a programot, a mesterséges intelligenciát — egyesítse.

Számos korábbi japán fejlesztési programtól eltérően az ötödik generációs számítógép megvalósításának cselekvési sorrendjét, munkarendjét nem egy konkrét, meghatározott célra tervezték. Csak a fejlesztés irányát adták meg pontosan, és néhány hosszú távú megvalósításra vonatkozó elvet foglaltak írásba. Szándékuk az volt, hogy folyamatosan mozdítsák előre a programot sorozatos, rövid távú célok felállításával. Ebben a könyvben először írnak japán szerzők az ötödik generációs számítógép megvalósításának programjáról.

A gép is ember
Nemzetközi karikatúra-pályázat
Szerk. Halász Góza
 (Budapest, 1988. SZÁMALK, 107 oldal. Ára: 75,— Ft)

A Neumann János Számítógéptudományi Társaság a közelmúltban pályázati felhívással fordult a világ karikaturistáihoz és a rajzos kedvű amatőrökhöz, hogy küldjék el képeikben megfogalmazott humoros gondolataikat a számítástechnikáról.

„Néhány éve a számítástechnika kevesek privilégiuma volt. A „fehér köpenyesek” világa nemcsak a humán értelmiség számára, de a műszakiak nagy részének is ködben úszó elefántcsonttoronynak tűnt. Ma a számítástechnika mindenkié. Általános iskolások programozzák személyi számítógépeiket, építésszek, gépészek, vegyészek kutató- és tervezőmunkájának eszköze a számítógép.

Történészek, nyelvészek, szociológusok sem nélkülözhetik: általános eszközzé vált, bekerült életünk vékeringésébe... Nem leszünk-e a gépek rabjai? — írja és kérdezi Havass Miklós a kötet előszavában.

A karikaturisták és az amatőr rajzolóok, a számítógépes szakemberek ezekre a kérdésekre keresik rajzaikkal a választ.

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

ADOK

C Plus/4 számítógép magnóval, programokkal eladó. Tel.: 143-031, 330-345, 15-17 óra között.

C Plus/4-re, ill. kibővített C16-ra SUPERMON program! A gépi kódú programozást jelentősen megkönnyíti, az eredeti monitor parancsai mellett számos új funkciót tartalmaz. Szabad memória 1000-BFFF-ig, a magnó hallható stb. Kérjen tájékoztatót! Kopasz Krisztián, Szeged, Mátyas tér 1. 6725

C Plus/4 + magnó + cartridge turbo másoló + kézikönyvek + 500 db program eladó 20 000 Ft-ért. Telefon: 889-794 este 5-8-ig.

C64 programkazettáim eladók. Kazettán-

ADOK-VESZÉK-CSERÉLEK

ként 45-55 program. Egy kazetta 400 Ft + utánvét. Dukán Zoltán, Sopron, Laktanya u. 20. 9400

C64 (energiatakarékos, C128-hoz hasonló dobozban) eladó, magnóval, több kazetta programmal, Data-Becker könyvekkel, továbbá 42 lemez tele programokkal. Tóth Bence, Budapest, Fraknó u. 46. 1115. Tel.: 653-742, du. 5-8-ig.

C64 számítógépet, kazettás egységgel, 250 db kazettával, lemezegységemmel 160 db lemezzel eladom (együtt és külön is). Járóka László, Budapest, Szív u. 3-5. fsz. 5. 1063

Commodore 64-hez GYORS-HÁTTÉRTÁR cartridge. Kapacitása 2-31 kb-át. A GYORS-HÁTTÉRTÁR-ba maximum 7 db, célszerűen gyakran használt program vihető be. A gép bekapcsolása után menüvel jelentkez be, és gombnyomásra a kiválasztott program azonnal fut. Javasolt programcsomagok: Turbo tape, Assembler, Monitor, Supergrafik, Help plus, Turbo másoló. File másoló 1699 Ft és Turbo tape 699 Ft.

Hozott programok elhelyezése a GYORS-HÁTTÉRTÁR-ban. Trompler László, Budapest, Attila u. 22. 1201. Tel.: 287-493 este.

Commodore VC-20 felhasználói és játékprogramok eladása. Kérjen tájékoztatót! Levélcím: Juhász György, Salgótarján, Pf.: 157, 3100

Commodore, Enterprise, TVC és Atari számítógépekhez HW cikkek széles választékát kínáljuk. Joystick-ok 450 Ft-tól kaphatók. Fényceruza 2000 Ft (C16-hoz és Plus/4-hez is). Válaszboríték ellenében elküldjük katalógusunkat. COMPUTEAM GM, Kaposvár, Berzsényi u. 32. 7400

3M floppy disk, originál, kétoldalas, dupla sűrűségű eladó. Tel.: 425-242 vagy 860-609.

PHILIPS VG-8020 típusú számítógépet adok. (80 k RAM, Z80 mikroprocesszor.) Németh Csaba, Bácsalmás, Türr István u. 1. 6430

Sinclair Spectrum (128 k) eladó (nem használt, vámközelt). Ár: 24 000 Ft. Róth Gábor, Budapest, Kléh I. u. 9. 1126

Tanulj, hogy elhelyezkedhess!

Minél magasabb valakinek a szakképzettsége, annál jobb esélye van az elhelyezkedésre, mutatják az NSZK-beli tapasztalatok. Míg 1980-ban csak 260 ezer, 1985-ben már 410 ezer dolgozó fordult a munkaügyi szolgálathoz, hogy szakmailag továbbképezze magát. Az igazi áttörés 1986-ban volt: 120 ezerrel növekedett a jelentkezők száma, s elérte az 530 ezret.

A tanfolyamok témáját tekintve a résztvevők 36 százaléka új műszaki módszerekkel ismerkedik, 14 százaléka az adatfeldolgozást tanulja, 12 százaléka a vezetőképzésben vesz részt, 11-11 százaléka a kereskedelmi és az ipartechnikai tanulmányokkal foglalkozik, 5 százaléka eladóképzésben vesz részt, s csupán 4 százalékuk tanul nyelveket.

Bővítik az oktatás anyagi kereteit is. Míg 1986-ban erre a célra 5 milliárd márkát fordítottak, az elmúlt évben már tíz százalékkal többet. Mindezt az a kedvező tapasztalat alapozta meg, hogy a tanulás segít az elhelyezkedésben: a résztvevők háromnegyede a tanfolyam elvégzése után fél éven belül elhelyezkedett.

Új géppark — egységes program

Egységes számítógépparkot alakítanak ki a Vas megyei fogyasztási és takarékszövetkezetek. A korábban használt kisebb teljesítményű, egymáshoz nem kapcsolható gépeket IBM PC-vel kompatibilis rendszerekre váltják fel. A Szövetkezeti Gazdaságszervezési és Számítástechni-

kai Iroda az országban elsőként a vasiak megrendelésére alakította ki a megyei szövetkezeti számítógépprogramot. Vas megye tíz áfésze közül hat már használja a rendszert, másik kettő pedig még az idén kiépíti. Nyolc vasi takarékszövetkezet ugyancsak ebben az évben kialakítja az új gépparkot, s használatba veszi az egységes programot, amely elsősorban a könyvelést, a számvitelt és az ügyvitelt egyszerűsíti.

Sophia Antipolis

A Cannes és Monaco között húzódó keskeny partsáv már nemcsak turisztikai paradicsom. Gazdasági szerepe is egyre nagyobb. Igaz, az idegenforgalma változatlanul irigylésre méltó: az elmúlt évben például az itt pihenő 8 millió turistától mintegy másfél milliárd dollár bevétele származott. Tavaly azonban első ízben a turizmus, mint iparág már a második helyre szorult az ott megtelepedett csúcstechnológia mögött: a huszonhét-ezer embert foglalkoztató ágazat 1987. évi forgalma meghaladta a 2,1 milliárd dollárt. Ennek központjaként — a kormány és a helyi hatóságok pénzügyi támogatásával, hogy a túlszűfolt párizsi régió tehermentesüljön —, már 1974-ben elkezdték a „tudomány városa” létrehozását az ismert nyaralóhely, Antibes fölötti sziklás-erdős területen. Vonzóerejét növelte a közeli Nizza, ahol Franciaország második legnagyobb repülőtere van. A Sophia Antipolis-nak nevezett csúcstechnológiai városban ma már százhatvan francia és multinacionális vállalat működik, köztük olyan ismertek, mint az amerikai Digital Equipment és a Dow Chemical, a brit Wellcome és a francia Thomson. Itt helyezték el az Air France számítógépes helyfoglaló világközpontját s több, nagy nyugat-európai légitársaság

hasonló centrumát. Japánt többek között a Toyota egyik kutatóközpontja képviseli. Az eddigi összes beruházások mintegy 1,75 milliárd dollárra tehetőek. Ezekhez párosult Franciaország legnagyobb kongresszusi központja Nizzában, amelyet most egy irodaépületekből, szállodákból, üzletekből álló 200 millió dolláros komplexummal egészítenek ki. Bár a francia Riviera továbbra is turistaparadicsom marad, a jelek szerint igazi jövője a csúcstechnológia ottani meghonosodásával kezdődik.

Cocom

Az amerikai kereskedelmi minisztérium Washingtonban bejelentette, hogy üzembe helyeztek egy új elektronikus hálózatot az exportáló cégek által benyújtott engedélykérelmek gyorsabb feldolgozása céljából. Az új eljárás keretében a beadványokat elbíráló tisztségviselők számítógép-terminálokra dolgozzák fel a kérelmeket, határozataikról pedig a kialakított elektronikus hálózatokon keresztül értesítik az exportőröket. A kidolgozott megoldás lehetővé teszi, hogy a korábbinál sokkal rövidebb válaszidőkkel működhessen az addig sokat bírált exportengedélyezési rendszer.

A kereskedelmi minisztérium az utóbbi időben évente több, mint 100 ezer exportengedélyt dolgozott fel. Az eddigi gyakorlat alapján az exportálni szándékozó vállalatoknak írásban kellett előterjeszteniük kérelmüket, a választ pedig postai úton kézbesítették, meglehetősen hosszadalmas eljárás után. Az új rendszer segítségével az év elején adták ki az első engedélyt: a kaliforniai Hewlett—Packard cég egy 12 ezer dolláros számítógéprendszert szállított Olaszországba.

ZX-Spectrum (48 k) + dual port interfész + joystick + magnó olcsón eladó. Sürgős! Jászolcs Péter, Budapest, Simonyi Zsigmond u. 1. 1191. Érdeklődni levélben vagy személyesen 17 óra után lehet.

Közepes méretű BASIC vagy Assembler programok írására programozót keresek. Tel.: 202-155, 18 óra után.

5500 programmal rendelkezem Commodore 16, Plus/4, 20, 64, 128 és Amiga számítógépekhez. Gyermán Sándor, 23000 Zrenjanin Rade Koncara 23. Jugoszlávia.

CSERÉLEK

C16 és Plus/4 programokat cserélek kazettán. Keresem a Winter Evento, Jet Set Willy II, Super Tape nevű programokat. Kranabeth Roland, Tapolca, Alkotmány u. 12/B. 8300

C16, Plus/4 programokat cserélek. Listát kérek! Viszlai Balázs, Múcsony, Kosuth u. 81. 3744

C Plus/4 gépet (3 hónapos, garanciális) + joystick, magnó, bő szakirodalom, sok program, elcserélném hasonló felszereltségű Enterprise-ra, vagy esetleg eladnám. Eladó VIC-20 (javítható kisebb hibával) üzemképtelen, esetleg alkatrészeknek, magnóval, felhasználói könyvekkel, programokkal. Károly János, Szigetszentmiklós, Árpád u. 19. 2310

C64-es programot, dokumentációt, hardver ötletet cserélek. Minden PRG érdekel kazettán és floppy-n is. Válaszokat listával! Ács Károly, Seregélyes, Münich F. u. 70. 8111

Enterprise programokat cserélek. Czulák Tamás, Pécs, Siklósi u. 28. 7632 Spectrumhoz felhasználói és játékprogramokat cserélek. Pilláry Gábor, Pécs, Bajcsy-Zs. u. 4. 3/9. 7622

ZX-Spectrum játék- és felhasználói programokat cserélek. A válaszokat listával kérem! Máró András, Hajdúnánás, Polgári u. 101. 4080

Ez a rovatunk KODEX 2000 szövegszerkesztővel készült.

Ami késik...

Az 1988/5. számunkban már múlt időben írtunk arról, hogy nagyszámítógépek szétszedett alkatrészeit árulja amatőrök számára egy Ápisz bolt, a SZÁ-MALK. Ennek előkészítése azonban — legnagyobb sajnálatunkra — elhúzódott, és az árusítás még mindig nem kezdődött el. Sajnáljuk, és reménykedünk — olvasóinkkal együtt. (A Szerk.)

●●●● Pontvadászat ●●●●

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

A feladatok után közöljük az elérhető maximális pontszámot. A rész megoldásokat is pontozzuk.

A pontgyűjtést, vagyis a pontvadászatot az esztendő végén zárjuk. A legjobb tíz versenyző nevét magazinunkban közzé is tesszük, ők lesznek azok, akik könyvutalványt is kapnak.

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

Beküldési határidő: 1988. augusztus 17.
Címünk: Mikroszámítógép Magazin Szerkesztősége
1371 Budapest, Pf. 433.

Jó vadászatot kíván a feladatok összeállítója:
 dr. Hoffmann Tibor

1. feladat

Egy R sugarú körlepből kivágjuk annak x -ed részét, mint körkívet. Ebből tölcsért készítünk. *Mekkora x mellett lesz a tölcsérnek a legnagyobb a térfogata?* (3 pont)

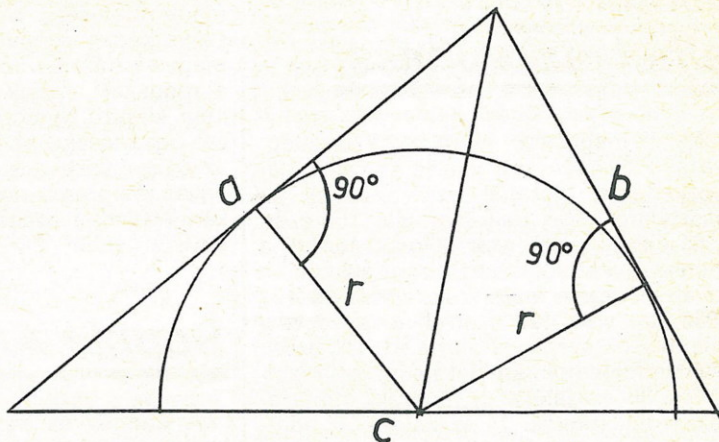
2. feladat

A gyakorlati életben hozzászoktunk a kerekítési hibáknak 10 -es számrendszerű jelöléséhez. Ha tehát például egy tizedes törtből 5 számjegyet írunk le és a hatodikat elhagyjuk, akkor a kerekítési hiba legfeljebb 5×10^{-6} . A mai számítógépek általában a 2 -es számrendszerre épülnek, melynek egy számjegye a bit. *Kérdés: hány bites szóhosszúságú számmal írhatunk le egy jól kerekített 9 jegyű tizedes törtszámot?* (3 pont)

Az 1988/5. szám feladatainak megoldása

1. feladat

Ha a háromszög tompaszögű, akkor nincs választásunk, csak a tompaszöggel szemben fekvő oldalon helyezhetjük el a kör középpontját. Más a helyzet hegyesszögű háromszög esetén.



Ha a c oldalon fekszik a középpont és r -rel jelöljük a kör sugarát, a háromszög területe az ábra szerint a következőképpen fejezhető ki:

$$t = \frac{ar}{2} + \frac{br}{2} = \frac{(a+b)r}{2}$$

vagyis

$$r = \frac{2t}{a+b}$$

Mint ahogy a terület adott, a sugár akkor lesz a legkisebb, ha $a+b$ a legnagyobb. Viszont a kerület, $a+b+c$ is adott, és így ez azt jelenti, hogy c a legkisebb. A helyes válasz tehát: akkor legkisebb a kör sugara, ha a középpontja a c oldalon van. (6 pont)

2. feladat

A vizsgálatnak az adott definíciót szigorúan betartva, feltétlenül a ciklusmag előtt kell végbemennie. Ekkor a logikai érték nem kívánatos értékénél a ciklusmagot átugorja a program. Sok esetben azonban ezt a vizsgálatot mégis a ciklusmag végén helyezik el. Ennek oka, hogy sokszor a ciklusmag végrehajtása folyamán alakítható ki a vizsgálandó logikai érték, például ha valamely numerikus változó nagyobb vagy kisebb voltának fennállása jelenti a logikai értéket. Utóbbi esetben azonban ügyelni kell arra, hogy üres ciklusnál eleve meggátoljuk azt, hogy a ciklusmag egyszer végrehajtsódjon, míg a ciklusmag előtti vizsgálatnál ez természetes. Ugyanakkor a ciklusmag utáni vizsgálatnál a ciklusból való kilépéskor nem kell a programban ugrást végrehajtani. Sok magasabb szintű programnyelven mind a két lehetőség választására mód nyílik, például ha a logikai értéket nagyobb vagy kisebb numerikus változó hozza létre. (4 pont)

A RAINBOW Számítástechnikai és Szolgáltató Kiszövetkezet új címe:



**1026 Budapest II.,
 Szilágyi Erzsébet fasor 17-21.**



Oktatrend

**Számítástechnikai
és
Elektronikai Kiszövetkezet**

IBM PC/XT-vel, AT-vel kompatibilis számítógépek, 32 bites számítógépek, rajzológépek, digitalizálótáblák, speciális hardverelemek.

Alap- és felhasználói szoftverek, kulcsrakész rendszerek fejlesztése.

Digitális és analóg áramkörök, készülékek tervezése, kifejlesztése, gyártása.

**Kedvező árak,
rövid szállítási határidő.**

1501 Budapest, Pf.: 7. Telefon: 263-910

VIDEOTON

VT-110

PROFESSZIONÁLIS SZEMÉLYI SZÁMÍTÓGÉP

A VIDEOTON új, az IBM PC/XT és AT-vel kompatibilis professzionális személyi számítógépe.

Önálló munkahelyi állomásként, lokális vagy postai hálózatba kapcsolva egyaránt széles körű alkalmazási lehetőséget kínál. VT-16 személyi számítógépre írt programok a VT-110-en futtathatók. A VT-110-es típus 640 Kb-os memóriával 10 vagy 20 Mb-os Winchester diszkkal, monochrom vagy nagy felbontású színes monitorral, 80 vagy 132 oszlopos normál vagy NLQ minőségű mátrixnyomtatóval, 1x360 Kb-os floppy diszkkal kerül forgalmazásra.

A hálózati alkalmazásokat az opcionálisan szállított hálózati csatoló és software, az adatmentést a különböző kapacitású streamerek biztosítják.

