

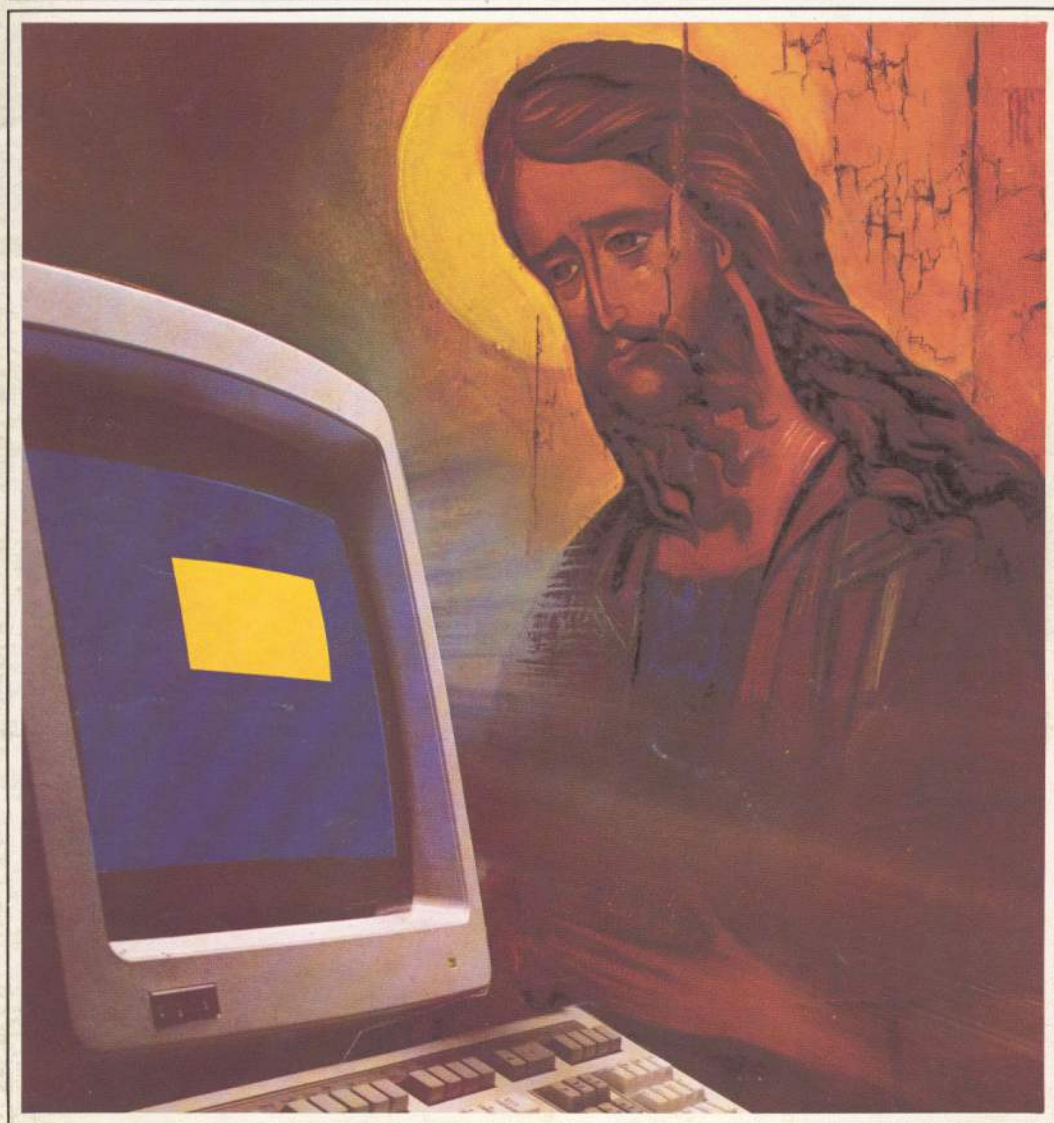


# mikro

számítógép

# magazin

Ára: 30 Ft





# MŰSZERTECHNIKA

## KISSZÖVETKEZET

Budapest, Szállás u. 21. 1107

Levél cím: Pf. 225. 1475

Bemutatóterem: Majakovszkij u. 1/D 1075

Telefon: 471-590

Telex: 22-7734

Telefon: 221-623

## Árkedvezmény

Áttérés az új adórendszerre a Műszertechnika komplex szolgáltatásával. Az áttérés meggyorsítása érdekében kiépítéstől függően 10-25% árkedvezményt adunk 1988. szeptember 15-ig az alábbi komplex rendszereinkre:

- **Munka/bér '88 komplex munkaügyi és bérszámfejtő**
- **Főkönyvi és folyószámla-könyvelési**
- **Beruházáselszámolási programcsomagok**

Fenti programok egy és több munkahelyes kiépítésű, 80 Mbájt háttérkapacitással rendelkező Novell hálózati rendszerben működnek. Komplex rendszereink telepítésére 1988 második negyedévre előjegyzést felvesszünk.

Programokkal kapcsolatban felhasználósoftware osztályunkon Zakariás Zsuzsanna osztályvezető ad felvilágosítást telefonon: 471-590 vagy személyesen 1107 Bp., Szállás u. 21.

## ETHERNET

csatlakozónkkal biztosítjuk a Novell alapú lokális hálózatokban jelenleg megvalósítható legnagyobb teljesítményű összeköttetést.

Az **ETHERNET** 10 Mbit/sec-os átviteli sebessége a legjobban elterjedt ARC-NET négyszerese.

Vállaljuk már működő Novell alapú hálózatok sebességnövelését.

Új rendszereinket kívánságra **ETHERNET** csatlakozóval szállítjuk.

**ETHERNET** alapú lokális hálózatunk megtekinthető a bemutatótermünkben.

**Felvételre keresünk** kiemelt bérezéssel, tőkés kompenzációs ügyletek szervezésében, bonyolításban jártas külkereskedelmi üzletkötőt német, angol nyelvtudással, saját gépkocsi használattal.

**Érdeklődni a 471-590/165-ös telefonon.**

**Felvételre keresünk** kiemelt bérezéssel szovjet árucereügyletek bonyolításában jártas szakembert, saját gépkocsi használattal.

**Érdeklődni a 471-590/165-ös telefonon.**





# mikro magazin

számítógép

## A NEUMANN JÁNOS SZÁMÍTÓGÉPTUDOMÁNYI TÁRSASÁG ÉS A KISZ KÖZPONTI BIZOTTSÁG LAPJA

A kiadvány  
a Tudományos-vezetési  
és Informatikai  
Intézetnél  
együttműködve készül

A szerkesztőbizottság

vezetője:  
Kovács Győző  
A szerkesztőség  
munkatársai:  
Babos János  
(tervezőszerkesztő)  
Bakos Tamás  
(programozástechnika)

Broczkó Péter  
(hírek)  
Énekes Ferenc  
(KISZ)

Jakab Ágnes  
(olvasószerkesztő)

Kovács Győző  
(levelezés)

Krasznai Éva  
(Diákszerkesztőség)

Lindner László  
(sakk)

Petróczy Judit  
(könyvek)

Pinke György  
(NJSZT, alkalmazások)

Simonyi Endre  
Szébenzski Sándor

Szulovszky Csaba  
Tamásné Lakó Erika

Terebessy Ákosné  
Varga András

(TII, iskola-számítógép)  
Vizessy Mária

Felelős szerkesztő:  
Könyves Tóth Pál

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

Levél cím:  
1371 Budapest  
Pf. 433.

Kiadja az Ifjúsági Lap-  
és Könyvkiadó Vállalat

Felelős kiadó:  
dr. Király G. István  
igazgató

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

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

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

Szikra Lapnyomda  
Budapest (86-0212)  
Felelős vezető:  
Csöndes Zoltán vezérigazgató

INDEX: 25 629  
ISSN 0236-6088

### Tartalom

Oktatástechnológia, nyílt tanulási rendszerek	2
Személyi számítógépek ikonutasításos rendszere	14
Egér-menü-ablak	18
Merre tart a világ?	19
Z80-as utasítástáblázat	23
Egy sarokkal olcsóbb!	28
Közprogramok	29
Rendszerfejlesztési eszközök	30
Mikrokalauz	32
Adok-veszek-cserélek	34
Olvastunk...	38
Új föld	40
μINFORM	42
Programtermék	45

### ISKOLA-SZÁMÍTÓGÉP

TechnoMIR	3
-----------	---

### DIÁK

Képernyőkezelési ötletek	4
Pardon!	5
TOP-lista	5
Rajzolóprogram Plus/4-re	6
Egy kis hardver	7

### PROGRAMOZÁSTECHNIKA

BASIC és gépi kód	8
Bináris fák	12

### μKLUB

Integrált szoftver	33
Adom a magyarázatot!	34
Építsünk számítógépet?...	35
Hogyan kerül kutya a számítógépbe?	36

### SAKK

Király plusz gyalog a király ellen I.	43
---------------------------------------	----

### AZ OLVASÓ ÍRJA

	44
--	----

### KÖNYVEK

	46
--	----

### HÍREK, ÉRDEKESSÉGEK

	47
--	----

### PONTVADÁSZAT

	48
--	----

Címképünk:  
Ramocsai Imri munkája

μ mikro  
magazin





... a' ki sokat tud, sokat akar, hogy tudjanak a' tanítványi és sokszor megesik, hogy a' kapacitásukon túl ragadtatik, és ők innen maradnak."

(Herpei Ádám nagyenyedi professzornak Boilyi Farkashoz írt leveléből)

# Oktatástechnológia, nyílt tanulási rendszerek

Az 1987/7. számban emlékeztem meg a 28 éve indult teremtamatikus-képzésről, az egyik első olyan egyetemi szakról, amelyben a megtanulandó diszciplínák többsége már a számítástechnikával volt kapcsolatban. Akkor — 1960-ban — az volt az újdonság, hogy számítástechnikai tantárgyakat tanítottunk, ma pedig az, hogy az oktatásban különféle számítógépes technológiákat alkalmazunk.

A technológizált oktatásnak még ma is többféle elnevezése van, én is váltakozva használtam az oktatás és tanulás kifejezéseket egy folyamat két részének meghatározására. A klasszikus oktatási rendszerekben a tanár a tanteremben oktat, a tanuló pedig otthon tanul. A tanárok a tantermi oktatásban különféle eszközöket (könyveket, táblát, krétát, tablót, filmet, videót és még sok minden mást is) használnak azért, hogy a diákok az anyagot már az iskolában megértsék, ti. a tanár is tudja, hogy magyarázat és szemléltetés nélkül a legtöbb diák az anyagot otthon képtelen volna egyedül megtanulni. Ezért amíg a tanteremben a tanárnak egy teljes oktatástechnológiai arzenál áll rendelkezésére, addig otthon a diáknak a tankönyvön, munkafüzetén és saját jegyzetein kívül szinte semmi. Ha tehát egy diák az iskolában figyelmen kívül hagyja a tanár jó tanácsait, azaz nem alakította ki saját tanulási technológiáját, akkor rosszul tanul, lemarad a többiekől, súlyosabb esetben kibukik az iskolából.

A klasszikus tantermi oktatásban a hangsúly tehát a tanári munkán van és csak másodsorban a tanuláson; vagyis a tantermi oktatás tanárorientált. Csak megjegyzem, hogy valószínűleg ezért kötelező a klasszikus oktatási rendszerben az iskolába járás és nemcsak az elemi és középiskolákban, de még az egyetemeken és a főiskolákon is, ti. a diákok többsége, miután nincs saját és az önálló tanulást támogató technológiája, képtelen a tananyaggal egyedül megbirkózni. Ennek a merev rendszernek persze azok látják kárát, akik szinte semmi segítséget nem igényelnek, még akár könyvből is elböldögölnek. Ráadásul van olyan diák, aki magától is gyorsan halad, míg van, aki folyamatosan, de lassan és megfontolva. Az első a klasszikus rendszerben vissza szokták tartani, a lassút pedig a tehetségét meghaladó „rohanásra” ösztökélik.

A klasszikus oktatásnak mindehhez természetesen megvannak a maga eszközei, például a napi feleltetés vagy az egyeteme-

ken a zárthelyi dolgozatok rendszere. Pedig végsősoron csak a teljes anyag tudásának ellenőrzésére lenne szükség; egy nagyon részletes vizsgával az anyagon belüli, illetve a más diszciplínákkal kapcsolatos összefüggések ismeretét is ellenőrizni lehet. Nyilvánvaló, hogy a klasszikus oktatási rendszer azért nem él ezzel a lehetőséggel, mert nem rendelkezik azzal a technológiával, amellyel az ilyen bonyolult ellenőrzést jó hatásokkal elvégezhetné.

A számítógép vagy a számítógéphez kapcsolott terminál az a mai eszköz, amely nemcsak a tanítást, hanem a tanulást is segíti, és amellyel ráadásul a folyamatos önellenőrzés, de még az alkalmankénti vizsgáztatás is sikerrel megoldható. Ezért látszik valószínűnek, hogy a számítógép és a tanulási technológiák megjelenése az iskolákban nemcsak az oktatási-tanulási folyamatot, de az iskolai rendszert magát is meg fogja változtatni.

A változások elsősorban a tanulást segítő programokra van szükség, amelyekből már nálunk is sokat használnak az iskolákban. Ezeket az oktatóprogramokat egyelőre magas szintű nyelven programozni tudó tanárok és diákok készítik, a tananyagot szinte kivétel nélkül BASIC-ben programozzák. Nem tehetnek mást: az iskolákban is és otthon is a rendelkezésre álló gépeknek (Commodore család, Sinclair, HT stb.) nincs és nem is lehet megfelelő szoftverkörnyezetük ilyen grafikalvas vegyes program fejlesztésére. Sajnos tanulóprogramfejlesztésben is lekörözött bennünket a világ, ma már az iparilag fejlett országokban, de a fejlődő országok jó részében is ún. szerzői rendszerek (authoring systems) vezérlése alatt írják a tanulóprogramokat. Ezek a rendszerek tulajdonképpen igen magas szintű nyelvek, amelyek segítségével a képernyőn a „mondanivaló” szöveges, rajzos, esetleg fényképes, sőt mozgóképes, de animált formában is gyorsan megkomponálható. Ennek a magas szintű nyelvnek a megtanulásához általában nem kell programozói ismeret, ugyanis a nyelv utasításait nem nehéz megjegyezni és alkalmazni. Így érthető, hogy ahol bevezették az oktatási programok írására a technológiát, ott már nem a programozók a vezéralakjai a tananyagkészítési munkának, hanem az adott tantárgyat jól ismerő például zenészek, irodalmárok, kémikusok, egyszóval a szakemberek.

Ezeket a programozási eszközöket nem a felsorolt „játék”-, hanem PC kompatibilis gépekre tervezték, de hasonló szoftver hátteret biztosít az 1988 végén nálunk is induló postai videotex-szolgáltatás is. A terminálként alkalmazott MUPID ugyanis (lásd a jM 1987/10. számától kezdődő cikksorozatát) az AUTOOL nevű szerzői rendszerrel programozható.

Ezért az a véleményem, hogy sürgősen felül kellene vizsgálni az iskola-számítógépprogramot, meg kellene határozni, hogy a tanulási technológia szabta követelményekhez milyen számítógép-beszerzési politikát volna célszerű követni. Azt hiszem, hogy inkább kevesebb, de PC kompatibilis gépet és hálózatba kapcsolt vtx-terminálot kellene vásárolni. Meg kell említenem, hogy a magániporra ez az értékesítésre vonatkozó jelenlegi szabályok (1988. február) a fejlődésüket illetően hátrányosan érintik az iskolák PC kompatibilis gépvásárlási törekvéseit, miután a rendelettel megszűnt az a forrás, ahonnan könnyen és gyorsan tudtak számítógépeket beszerezni.

Lehet, hogy nem volna szabad jósolni, de azt hiszem, hogy a „jövő” technológizált iskolája teljesen más lesz, mint a mai; nemcsak a környezete változik meg, de a tanár és a diák viszonya is. Feltehetően nem lesz ebben a jövőbeni iskolában a mai értelemben vett tanterem, a pedagógus nem tanít, hanem tananyagokat készít valamilyen hardverre, legfeljebb néha tart egy-egy diáknak konzultációt. Ezekből az anyagokból a diák a tanár személyes közreműködése nélkül is képes lesz a szükséges ismereteket megszerezni. A diákok különféle tesztekkel ellenőrizhetik a tudásukat, és ha megfelelő módon felkészültek, akkor jelentkezhetnek az ugyancsak technológizált gépi vizsgára. Miután mindenki a tehetsége diktálta ütemben tanul, ezek a vizsgák nem lesznek időhöz kötve, mindenki akkor vizsgázik, amikor az anyagból jól felkészült. Ha valaki a vizsgán nem felelt meg, akkor a procedúrát annyiszor ismételteti, ahányszor csak akarja. Így azután a mércét igen magasra lehet állítani, nem kell félni a sok bukástól, mert valójában az nem is lesz.

A jövő iskolája már ma sem utópia, a megvalósításnak — azt hiszem, ebben egyetértünk — nem technikai, sokkal inkább emberi akadályai vannak. Talán ezért lehet csak lassan megvalósítani.

KOVÁCS GYÖZÖ



# TechnoMIR

## Egy érdekes bemenet

A megszakító bemenet (interrupt input vagy röviden ITIN) modul lényegét és fő felhasználását tekintve egy digitális bemenet. A különbség ehhez képest annyi, hogy a beérkező jel hatására a modul megszakítást kezdeményez. Ezzel kiküszöböli a DIN modul azon hátrányát, hogy elveszti az információt, ha időben nem kérdezzük le. Meg lehet csinálni az ITIN-nel azt, amit a DIN-nel nem: valamilyen más feladat végrehajtása közben is végezhető számítógépes adatkezelés. Mivel élvezet azonban az egyszerű programozhatóság, ezt a modult csak annak ajánljuk, aki igazán ismeri a számi-

tógépét, és assemblerben is tudja programozni.

Az itt következők a HT-1080Z gépeken dolgozóknak segítenek, de mások is meríthetnek ötleteket belőle. A programot a modul fétalalói készítették, tőlük vettük át.

### Assembler program ITIN modul használatához

1. Csatlakoztassuk az ITIN modult a HT-1080Z számítógéphez.

2. Kapcsoljuk be a számítógépet. A NEW LINE gomb lenyomása után a képernyőn > READY látható.

3. Gépeljük be: SYSTEM és nyomjuk le a NEW LINE (a továbbiakban: NL) gombot.

4. A képernyőn megjelenő kérdőjel után gépeljük be: /12710 NL.

5. Monitor üzemben vagyunk, a képernyőn megjelennek a processzor regiszterei. Gépeljük be a listát.

Látszólag semmi nem történik. A processzor 7104(h) címen HLT utasításon áll. Ha rövidre zárjuk a modul valamelyik bemenetét, a képernyőn megjelenik a beolvasott kódnak megfelelő ábra. A program a RESET gombbal állítható meg.

Bővíthető a program például úgy, hogy HLT helyett az érkező kódnak megfelelő elágazásokat szervezzünk.



### A külső tápegység

A TechnoMIR rendszer eddig ismertett digitális moduljai nem nagy áramfelvételűek, de a számítógépek szokásos terhelhetősége (kb. 0,15 A) nagyon könnyen túlléphet. A DINT és DOUT együttes alkalmazása, ha azok teljes terhelhetőségét kihasználjuk, legalább 0,2 A áramigényt jelent. Ez a számítógép buszrendszerén át nem elégíthető ki, külön táplálás kell.

A modul típusa	+5 V	+12 V	-12 V
HCLK	0,09 A	—	—
A/D	0,075 A	0,025 A	0,05 A
D/A	0,09 A	0,1 A	0,04 A
ITIN	0,14 A	—	—
AMUX	0,15 A	0,03 A	—
DINT	0,13 A	—	—
DOUT	0,15 A	—	—
JOY	0,085 A	—	—
PB	0,085 A+X	X	X
D100	0,025 A	—	—
C64	0,085 A	—	—
OPTO	0,12 A	—	—
UNI	0,11 A+X	X	X

Az analóg modulok működéséhez +12 V és -12 V szükséges; ezt a számítógépek nem, vagy nem elég pontosan adják. A különböző modulok áram- és feszültségigényét a táblázat mutatja. Látható, hogy általában két digitális modul használhatunk együtt, ennél többet csak a POW moduldal együtt csatlakoztathatunk a számítógéphez.

A POW modul három változata közül a legegyszerűbb a fényképen látható. Ez az első is együtt. A modulhoz csatlakoztatott +8 V—+10 V egyenfeszültségből egy 7805 típusú stabilizátor IC állítja elő a +5 V-ot. Az analóg modulok +12 V, -12 V táplálásához 18 V-os váltakozó feszültséget kell a POW modulhoz csatlakoztatni.

Hamar kiderült azonban, hogy ez nem túl előnyös konstrukció, az iskolák zömében nem egyszerű feladat a szükséges feszültségeket „elővarázsolni”. További gond, hogy nem elég „diákálló” és „tanárálló”, nincs túláram és fordított polaritás elleni védelem. Ez a típus az OPI által kipróbálásra kiosztott készletekkel került forgalomba.

A hátrányos tulajdonságokat kiküszöbölendő fejlesztették ki a két újabb típust. Egy nem igazán esztétikus, szürke doboz tetején helyezték el az eredeti modult. Erre az alapdobozra vezették ki a gyártók a tápegység által előállított feszültségszintek csatlakozóit, a főkapcsolót és egy túláramjelző LED-et.

Az előállított feszültségek és terhelhetőségek:

+ 5 V referencia	100 mA
+ 5 V	1 A
+12 V	100 mA
-12 V	100 mA

A felhasználó szempontjából még lényegesebb változtatás a táplálást érinti. Az M6.1 POW A típusú modulhoz 220 V váltakozó feszültség szükséges. Az érintésvédelmi előírásoknak megfelelően nem csatlakoztatható közvetlenül a hálózathoz, leválasztó transzformátort kell alkalmazni. Az M6.2 POW B típus 24 V-tal kell táplálni, így az érintésvédelmi problémák is megoldódnak.

A POW modul alkalmazásával a TechnoMIR rendszer lehetőségei ugrásszerűen kiszélesednek. Gyakorlatilag bármekkora torony építhető a modulokból, azaz igen összetett érzékelési-beavatkozási műveletor hajtható végre.

KIRÁLY LÁSZLÓ—ALBU LÁSZLÓ

Utasítás	Értelmezése
M4012	Cím beállítás, a képernyőn megjelenik a cím és tartalma.
C3	
00	JP 7000(h) a program kezdő címe
70	
X	Újra megjelennek a regiszterek címét
M7000	
F3	DI IT le-tiltás
00	IN A,(n) Input ITIN
19	nn=19(n)=20(d)
32	LD (nn),A
0F	nn=30E(h) az ITIN tartalma a képernyőre
3C	
FE	CP n
FF	nn=FF(h)=200(d)
20	JR Z,(VEDE)
03	relatív ugrás a VEDE címre
C3	JP nn
01	nn=7001(h)
70	
VEDE F3	EI IT an=bedeltetés
C9	
X	RET Újra megjelennek a regiszterek
M7100	
F3	DI IT mód beállítás
ED	IM1
56	
F3	EI
Itt folytatódóan a felhasználói	
Program. Pl.:	
76	MLT A PRO-cesszor vár
C3	
01	JP nn
71	nn=7101(h)
X	
G7100 NL	Vezérlés 7100(h) címre



# Képernyő-kezelési ötletek

C128

A C128 képernyőkezelésre vonatkozó néhány tapasztalat, segédprogram, rutin remélhetőleg segítséget nyújt azoknak, akik a gép ilyen lehetőségeit is ki akarják használni.

## Grafikus képernyő

Az első két ötlet a grafikus képernyő lehetőségeit bővíti.

A PRINT SHOP nevű programmal grafikát és nagyméretű betűkből álló szöveget készíthetünk, a KOALAPainter segítségével többszín-üzemmódban színes rajzokat állíthatunk elő. E két, C64-re készült népszerű program ábrát a C128-on is betölthetjük, de az utóbinnál egy kis átalakításra van szükség, mert míg a PRINT SHOP csak a bitérképet tárolja, addig a KOALAPainter ezt egyben veszi fel a két szín RAM-mal (együtt 40 blokk), amely a C128-on máshol helyezkedik el.

A PRINT SHOP-nál érdemes kiszínezni a képet. Ezt egy rövid gépi kódú rutinnal megtehetjük, a háttér és előtér legkedvezőbb összeállítását választhatjuk ki.

Indítás után a program (1. lista) először beállítja a gépi kódú rutint (200–250-es sor), majd megkérdezi a tárolt kép nevét.

```
10-REM ***** PRINT SHOP-ATALAKITO *****
20 GRAPHIC 0,1:GOSUB 200
30 INPUT "NEV";N$
40 GRAPHIC 1,1:BLOAD (N$),B0,P0192
50 E=0:H=0
60 GETKEY AS
70 IF AS="E" THEN E=E+1
80 IF AS="H" THEN H=H+1
90 IF AS=CHR$(13) THEN GOTO 140
100 IF E=16 THEN E=0
110 IF H=16 THEN H=0
120 POKE 2027,H+E*16:SYS 2016,COLOR4,H+1
130 GOTO 60
140 BSAVE (N$),B0,P7168 TO P16192
150 GRAPHIC 0,1
160 END
200 FOR I=0 TO 24
210 READ A:POKE 2016+I,A
220 NEXT I:RETURN
230 DATA 169,28,133,251,169,0,133,250
240 DATA 162,4,169,1,160,0,145,250,200
250 DATA 208,251,230,251,202,200,244,96
```

### 1. lista

Betöltés után az E és H betűkkel növelhetjük az előtér, illetve a háttér színkódját. A megtaláltuk a megfelelő színkombinációt, cseréljünk lemezt, és nyomjuk meg a RETURN-t. Ekkor a gép felveszi a képet eredeti nevét, így csupán 4 blokkal bővült a fájl. Ezután a kép a következő utasítássorozattal bármikor betölthető:

GRAPHIC 1,1:BLOAD "név"  
A gépi kódú listát a monitorba lépve a D B00 B18 parancssal nézhetjük meg.

Fent említették miatt a KOALAPainter-nél a dolog nem ilyen egyszerű. A C128 standard üzemmódi szín RAM-ja 7168–8167-ig, bitérképe 8192–16191-ig, a többszín-üzemmódban a további szín RAM (több szín 2) 55296–56295-ig terjed, így ezeket a részeket külön kell kimenteni.

Amikor a KOALAPainter-rel (2. lista) felvesszünk egy képet, nem szükséges nevet adni neki; használhatjuk csak a betűjét

```
10 REM ***** KOALAPainter-ATALAKITO *****
20 GRAPHIC 1:GRAPHIC 0,1:PRINT
30 INPUT "A KÉP BETŰJE";B$
40 INPUT "A KÉP NEVE";N$
50 N$=CHR$(129)+"PIC "+B$+" "+N$+" "
60 PRINT "TEDD BE A LEMEZT!";GETKEY AS
70 BLOAD (N$),B0,P4064
80 PRINT "TEDD BE A C128-AS LEMEZT!";
90 GETKEY AS
100 F$=B$+" "+N$+" "
110 BSAVE (F$+"1"),B0,P4064 TO P12064
120 BSAVE (F$+"2"),B0,P12064 TO P13064
130 BSAVE (F$+"3"),B0,P13064 TO P14064
140 PRINT "KESZ.A TOLTOVEL BEOLVASHATO."
150 END
200 REM ***** KOALAPainter-BETOLTO *****
210 GRAPHIC 0,1:PRINT
220 INPUT "A KÉP BETŰJE";B$
230 INPUT "A KÉP NEVE";N$
240 INPUT "HATÁRSZIN";H
250 F$=B$+" "+N$+" "
260 GRAPHIC 3,1:POKE 0,44
270 COLOR 0,H+1,COLOR 4,H+1
280 BLOAD (F$+"1"),B0,P0192
290 BLOAD (F$+"2"),B0,P12064
300 BLOAD (F$+"3"),B0,P15296
310 SOUND 1,5000,0,1
320 GETKEY AS:GRAPHIC 0,1
```

### 2. lista

Is azonosítottunk a lemezen (SAVE vagy SAVE&NAME). Az átalakító rész a 10–150 sorokig terjed. Először meg kell adnunk a betűt, és ha van, a nevet (ha nincs, akkor a RETURN a válasz). Majd behelyezzük a lemezt, amin a kép van, s a program betölti. Ezután azt a lemezt tegyük be, amire fel akarjuk venni az átalakított képet, és nyomjuk meg egy gombot. Ha készen vagyunk, és ellenőrizzük akarjuk a felvételt, GOTO 200-azt tölthetjük be a képet. Itt a háttérszint is meg kell adni. Az olvasás végét a gép sipolással jelzi.

Ha más programban is használni akarjuk a többszínű képet, csak a 250–300-as sorokat kell meghagyni. A 260-as sor bekapcsolja a grafikus képernyőt és a be-kim-netet.

## Karakteres képernyő (40 × 25)

Bizonyos programoknál szükség lehet a képernyő különböző irányú eltolásaira.

### 3. lista

```
10 REM ***** SCROLL BALRA, JOBBRA *****
20 FOR I=0 TO 165:READ A:POKE 2016+I,A
30 S=S+A:NEXT I
40 IF S=25547 THEN GOTO 60
50 PRINT"HIRA VAN AZ ADATOKBAN.":END
60 KEY1,"SYS 2016"+CHR$(13)
70 KEYS,"SYS 2096"+CHR$(13)
80 SCNCLR:PRINT"71-BALRA, 72-JOBBRA"
90 DATA 169,28,133,251,169,0,133,250,162
100 DATA 26,169,32,145,250,169,48,24,181
110 DATA 250,133,250,144,2,230,251,202
120 DATA 208,230,132,252,200,132,250,162
130 DATA 4,160,0,169,4,133,251,133,253
140 DATA 177,250,145,252,169,216,133,251
150 DATA 133,253,177,250,145,252,200,200
160 DATA 233,230,30,1,230,48,11,202,200
170 DATA 224,169,4,141,38,11,169,216,141
180 DATA 48,11,96
190 DATA 169,3,133,251,169,255,133,250
200 DATA 160,0,162,26,169,32,145,250,169
210 DATA 48,24,181,250,133,250,144,2,230
220 DATA 251,202,200,230,160,231,132,252
230 DATA 136,132,250,162,5,160,0,167
240 DATA 133,251,133,251,177,250,145,252
250 DATA 169,219,133,251,133,253,177,250
260 DATA 145,252,136,192,255,200,231,206
270 DATA 122,11,206,132,11,202,200,222
280 DATA 169,7,141,122,11,169,219,141
290 DATA 132,11,96
```

A le- és felfelé scrollt nem kell megírunk, mert a rendszer rutinjai között megtalálható, így akár gépi kódú programból is meghívhatók.

scroll fel: SYS 51900  
scroll le: SYS 51914  
A balra, illetve jobbra történő görgetést már nekünk kell elkészíteni.

A BASIC betöltőprogram (3. lista) jelzi az esetleges adatbepótlási hibákat. Az F1 gombbal balra, az F3 gombbal jobbra tolhatjuk a képernyő tartalmát. A szín RAM-ot is kezeli a rutin, tehát a karakterek színükkel együtt mozognak. A két gépi kódú lista monitor üzemmódban a D B00 B4F, illetve a D B50 BA5 parancsokkal írható ki. Felvehető a BSAVE "név", B0,P2816 TO P2982 utasítással.

A következő rutinnal a további képernyőket használhatjuk, melyek között az F1, F3, F5, F7 gombokkal kapcsolhatunk át. Ez hasznos lehet hosszú programok hibakeresésénél, belövésénél, mert fontosabb részeit külön oldalak listázhatjuk. Ha az eredeti képet szeretnénk visszakapni, üssük le a HELP billentyűt. Ez egyben visszaállítja a megszámláltsékvectort, és az eredeti állapotba jutunk. Egy-egy átkapcsolás után nyomjuk meg a CRSR→HOME gombot, csak ekkor jelenik meg a villogó kurzor.

```
10 REM ***** 4 KÉPERNYŐ *****
20 FOR I=4096 TO 4185:POKE I,0:NEXT I
30 FOR I=0 TO 115:READ A:POKE 2016+I,A
40 S=S+A:NEXT I
50 IF S=11090 THEN PRINT"RENDBEN.":END
60 PRINT"HIRA VAN AZ ADATOK KOZOTT.":
70 DATA 120,169,13,141,28,3,169,11,141
80 DATA 21,3,08,96,72,138,72,166,213
90 DATA 224,4,208,13,169,132,141,44,18
100 DATA 169,32,141,59,10,76,110,11,224
110 DATA 5,208,13,169,148,141,44,10,169
120 DATA 36,141,59,10,76,110,11,224,6
130 DATA 208,13,169,164,141,44,10,169,40
140 DATA 141,59,10,76,110,11,224,3,208
150 DATA 13,169,180,141,44,10,169,44,141
160 DATA 59,10,76,110,11,224,64,208,20
170 DATA 169,28,141,44,10,169,4,141,59
180 DATA 10,169,101,141,28,3,169,250,141
190 DATA 21,3,104,170,104,76,101,250
```

### 4. lista

A betöltő begépelése előtt írjuk be: POKE 45,1:POKE 46,49:POKE 12544,0:NEW

Ez feljebb teszi a BASIC-kezdőt. A grafikus képernyőt így nem használhatjuk: itt vannak a képernyők és a program.

A program (4. lista) törli a funkciósbillentyűket és a HELP gombot (20-as sor). Jelzi az adathibákat. A RENDBEN felírt megjelenése után SYS 2816-tal indíthatjuk a rutint. A gépi kódú listát megtekinthetjük a monitorba lépve a D B00 B71 parancssal. Felvehető a BSAVE "név", B0, P2816 TO P2932 utasítással.

Van néhány tárcim — a gép rendszerváltózi —, amelyek átírásával érdekes dolgokat érthetünk el.

A betűismétlés várakozási idejét a 2595-ös címen írhatjuk át. A 2598-as tárcim tartalmaz 64-re változtatva kikapcsolható a kurzor villogása. Ha nullát írunk a 2599-es regiszterbe, akkor a kurzor a program futása közben is láthatóvá válik.

NÉMETH ISTVÁN



# Pardon!

Sajnálatos módon hiba került az 1988/2. számunk 8. oldalán megjelent, Egy kis zene a Plus/4-en című cikkbe. A jelzett lapoldal felső és alsó programja egyenértékű; az utóbbi egy elegánsabb megoldás akart lenni. Eleganciáját rontják a benne található rossz ugrási címek, valamint a gépi kódú lista hiánya.

A hibás sorok ugráscímei helyesen:

a 270-es sorban 250

a 320-as sorban 300

A dallamot a SYS 16718-as utasítással kapcsolhatjuk ki. Végezetül a programokhoz tartozó gépi kódú listát is közöljük. A „zenebonáért” olvasóink szíves elnézését kérjük. A szerkesztőség

## A gépi kódú lista

```
>4020 20 91 94 20 E1 90 A5 15 : ██████████
>4028 85 09 A5 14 85 08 20 91 : ██████████
>4030 94 20 E1 90 A5 15 85 08 : ██████████
>4038 A5 14 85 0A 78 A9 95 80 : ██████████
>4040 14 03 A9 40 80 15 03 58 : ██████████
>4048 AD 12 FF 29 FC 8D 12 FF : ██████████
>4050 AD 10 FF 29 FC 8D 10 FF : ██████████
>4058 A9 01 8D 00 40 8D 01 40 : ██████████
>4060 8D 02 40 8D 03 40 A0 00 : ██████████
>4068 B1 08 0D 12 FF 8D 12 FF : ██████████
>4070 A5 08 18 63 01 85 08 A9 : ██████████
>4078 00 65 09 85 09 A0 00 B1 : ██████████
>4080 DA 0D 10 FF 8D 10 FF A5 : ██████████
>4088 DA 18 63 01 85 DA A9 00 : ██████████
>4090 65 08 85 08 60 48 98 48 : ██████████
>4098 8A 48 CE 00 40 10 42 A9 : ██████████
>40A0 EF 2D 11 FF 8D 11 FF AD : ██████████
>40A8 02 40 F0 35 CE 02 40 F0 : ██████████
>40B0 03 A9 00 8D 00 40 4C E1 : ██████████
>40B8 40 A0 00 B1 08 8D 0E FF : ██████████
>40C0 C8 B1 08 8D 00 40 C8 B1 : ██████████
```

```
>40C8 D8 8D 02 40 AD 11 FF 09 : ██████████
>40D0 1F 8D 11 FF A9 03 18 65 : ██████████
>40D8 08 85 D8 A9 00 65 09 85 : ██████████
>40E0 09 CE 01 40 10 42 A9 DF : ██████████
>40E8 2D 11 FF 8D 11 FF AD 03 : ██████████
>40F0 40 F0 35 CE 03 40 F0 08 : ██████████
>40F8 A9 00 8D 01 40 4C 28 41 : ██████████
>4100 A0 00 B1 DA 8D 0F FF C8 : ██████████
>4108 B1 DA 8D 01 40 C8 B1 DA : ██████████
>4110 8D 03 40 AD 11 FF 09 2F : ██████████
>4118 8D 11 FF A9 03 18 65 DA : ██████████
>4120 85 DA A9 00 65 08 85 08 : ██████████
>4128 AD 02 40 D0 19 AD 03 40 : ██████████
>4130 D0 14 78 A9 0E 8D 14 03 : ██████████
>4138 A9 CE 8D 15 03 58 AD 11 : ██████████
>4140 FF 29 CF 8D 11 FF 68 AA : ██████████
>4148 58 A8 68 4C 0E CE 78 A9 : ██████████
>4150 00 8D 11 FF A9 0E 8D 14 : ██████████
>4158 03 A9 CE 8D 15 03 58 60 : ██████████
>4160 0E CE 00 FF 00 FF 00 FF : ██████████
>4168 00 FF 00 FF 00 FF 00 FF : ██████████
```

## BASIC Merge Primón

A Merge olyan összefűzési lehetőség, amivel a már tárban lévő programhoz magánról tölthetünk más programrészeket. A művelet igen hasznos lehet, amikor például valaki összerak magának egy „szerszám-

mosládát”. Ez ugyanis mindazt a program-emet tartalmazza, amely valamennyi programban egyforma, így elég egyszer megírni. Ezután ezt egy félkész programhoz töltve, nem kell listabeírással vagy újrással foglalkozni, mert igény szerint egyszerűen csak elővesszük a „szervényeket”, és azokat használjuk. Az alkalmazás úgy istenigazából jó szórakozás.

De menjünk tovább egy egyszerű megoldásra! Íme a Merge. A BASIC program kezdőcímet a memóriában a 16548-as bajton kezdődő kétbájtos mutató adja meg, ezt kiíratva jegyezzük fel: PRINT PEEK (16548)+256\*PEEK (56549), ami általában 17386. Ugyanígy írassuk ki a 16633 ci-

men kezdődő kétbájtos mutatót is. Ez két bájtalt az utolsó programsor mögé mutat. Ezért a BASIC kezdete mutatóba két bájtalt kevesebbet kell írunk: POKE 16548,x-256\*int(x/256),x/256, ahol x az érték.

A gépből lévő program nem listázható és nem is indítható. Tehát beírandó a LOAD, majd betöltés után vissza kell írni a 16548-as címre az értékeket. Ügyeljünk arra, hogy a második program legkisebb sora is nagyobb legyen, mint az előző programé, mert különben csúnya keveredések lehetnek az azonosságából.

LILING ZOLTÁN

## TOP-lista

A külföldi szaklapok példáiából okulva, mi is indítani kívánunk egy TOP-listát, két kategóriában. Az elsőben a felhasználói, a másodikban a játékprogramok szerepelnek. A legjobbakat, a legérdekesebbeket rangsorolnánk a beküldött javaslatok alapján. Ehhez kérjük az olvasók közreműködését. C64-re, ZX-Spectrumra, Enterprisera, TVC-re, ATARI-ra és IBM-re készült programrangsorokat várunk havonta.

Címünk: 1371 Bp. Pf.: 433. Mikroszámítógép Magazin Szerkesztősége.

Diákszerkesztőség

### Felhasználói programok

- |                    |               |
|--------------------|---------------|
| 1. GEOS 1.3        | C64           |
| 2. ART STUDIO 1.2B | C64, Spectrum |
| 3. CRACKER COPY II | C64           |
| 4. PRINT FOX       | C64           |
| 5. MUSIC SHOP      | C64           |
| 6. LIGHT SHOW      | Spectrum      |
| 7. ARTIST SHOW     | Spectrum      |
| 8. LASER-GENIUS    | Spectrum      |
| 9. GAME MAKER      | C64           |
| 10. SPEED LOAD     | C64           |

### Játékprogramok

- |                          |          |
|--------------------------|----------|
| 1. DEFENDER of the CROWN | C64      |
| 2. WIZBALL               | C64      |
| 3. THE LAST NINJA        | C64      |
| 4. TETRIS                | IBM      |
| 5. ARKANOID              | C64      |
| 6. BETTY                 | Spectrum |
| 7. BARBARIAN             | Spectrum |
| 8. EXOLON                | Spectrum |
| 9. SPACE GAMES           | C64      |
| 10. WORLD GAMES          | C64      |







# Egy kis hardver

Az alábbi kiegészítést elsősorban azoknak ajánljuk, akik nem teljesen járatosak a hardver területén. Ugyanakkor kellő odafigyeléssel természetesen bárki elvégezheti a javasolt módosítást Commodore gépében. Segítségével nyomom követhető a párhuzamos adatforgalom a user porton, illetve a floppy működése a megszakításokon keresztül. Lehetőségünk nyílik továbbá arra, hogy a gép működését bármely állapotban felfüggesztjük. (A programok nagy része addig marad „lefagyasztott” állapotban, amíg a kapcsolót bekapcsolva tartjuk.)

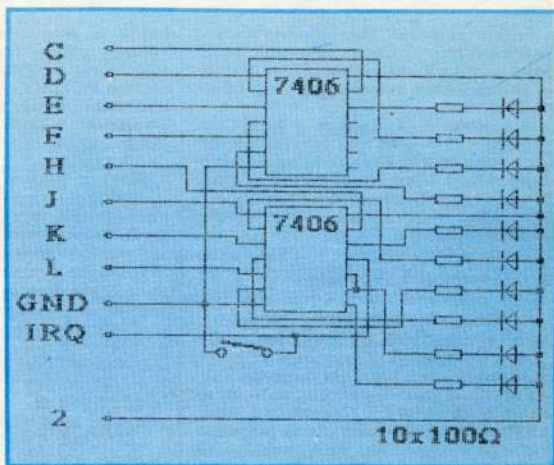
A lista olyan programot tartalmaz, amely egy futófényt vezérel. A sebességet a 49175 tárcím módosításával változtathatjuk. Természetesen nemcsak 44 darab lépés programozása lehetséges. Ha megtervezünk egy villogási sorozatot, akkor a lépéseket sorban az 50000 címűtől kezdődően rögzítjük, a darabszámot pedig a 49192 címre írjuk. (Például amikor azt szeretnénk, hogy az első és az utolsó LED villogjon felváltva, akkor a következőket kell beírni: POKE 50000,1 : POKE 50001,128:POKE 49192,2.) Mivel a program belső változói futás közben módosulnak, célszerű beírni az adatokat a SYS 49152 meghívása előtt. A program egyébként megszakításvezérelt, így újraindítását is az előbb említett utasítással érhetjük el.

Sajnos, a LED-eket csak a régi típusú számítógép belsejébe lehet beépíteni, akiknek viszont újabb típusú C64-ük van, azok a fentiek egy részét a user portra csatlakoztatott futófényt NYÁK-kal valósíthatják meg. AZ IRQ kapcsolót a bővítés helyére lehet csatlakoztatni. Mivel ezek egymástól szétválasztható bővítések, külön-külön is elkészíthetők.

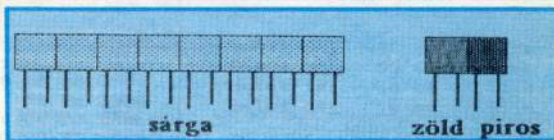
A megvalósítás figyelmet igényel. A konstrukciót azonban mindenki maga választhatja ki! Aki nem tud nyomtatott áramkört maratni, az egyszerűen forrasszon egy IC-foglatat lábaira vezetékeket. (Én is ezt a módszert választottam.)

```

10 REM *** BB SOFTWARE & ATOMHARD ***
11 FOR I=0 TO 68
12 READ A : POKE 49152+I, A : S=S+A
13 NEXT
14 IF S<>8806 THEN PRINT"HIRA !":STOP
15 FOR I=0 TO 43
16 READ A : POKE 50000+I, A
17 NEXT
18 SYS 49152
19 REM -GEPI KODU PROGRAM-
20 DATA 120, 169, 255, 141, 3, 221, 169, 21
21 DATA 141, 20, 3, 169, 192, 141, 21, 3, 88
22 DATA 96, 234, 234, 234, 234, 162, 15, 224
23 DATA 20, 240, 6, 238, 25, 192, 76, 49, 234
24 DATA 169, 1, 141, 25, 192, 162, 44, 224, 44
25 DATA 240, 15, 174, 42, 192, 189, 79, 195
26 DATA 141, 1, 221, 238, 42, 192, 76, 49, 234
27 DATA 169, 0, 141, 42, 192, 76, 48, 192, 234
28 REM -FUTOFENY ADATOK-
29 DATA 0, 128, 64, 32, 16, 8, 4, 2, 1, 129, 65
30 DATA 33, 17, 9, 5, 3, 131, 67, 35, 19, 11, 7
31 DATA 135, 71, 39, 23, 15, 143, 79, 47, 31
32 DATA 159, 95, 63, 191, 127, 255, 254, 252
33 DATA 248, 240, 224, 192, 128
    
```



1. ábra



2. ábra

Elsőként a gép fedőlapjáról távolítjuk el a klaviatúrát, melyet csupán néhány csavar rögzít. Majd szikével vágunk ki a szellőzőnyílásszerű középső mélyedéséből egyszer 1, egyszer pedig 4,1 centimétert. A vágásokat felülről csináljuk. Először hosszirányban, utána keresztirányban. A keletkezett részekbe kerülnek be a világító diódák, melyeket előzőleg pillanatragasztóval ragasszunk össze a 2. ábrának megfelelően. Az oldalait azután dörzspapírral csiszoljuk simára és méretre úgy, hogy pontosan illeszkedjenek a kialakított résbe. A LED-ek rövidebb lábait vágjuk még kisebbre, s forrasszuk fel az ellenállásokat, a hosszabbik lábukat forrasszuk össze. Ha nem készítettünk nyomtatott áramkört, akkor az 1. ábrán látható kapcsolási rajznak megfelelően használjuk be az alkatrészeket, majd ragasszuk fel őket. (A kapcsolási rajzon szereplő legelső dióda piros, a felette lévő zöld, az összes többi sárga színű, de mindegyik szögletes.) A diódák beragasztásánál ügyeljünk arra, hogy tetejük egy síkba essen a gép dobozának felületével. A kapcsolók hátoldalhoz való rögzítése ezek típusától függően változhat. Az áramkörünk felől érkező vezetékeket ne forrasszuk rá közvetlenül az alaplapra, hanem a szétszedhetőség miatt, lehetőség szerint, iktassunk be egy csatlakozópárt is. (Vigyázat, a C-től L-ig terjedő user port kimenetek az alsó oldalon találhatók!)

A fent javasolt módosítás a gép alapvető működésére nincs hatással, tehát ezzel üzemképtelenné vagy bizonytalan működésűvé tenni nem lehet. Viszont akinek beválik a változtatás, annak ajánljuk, hogy lépjen tovább a kazettavonalak kivezetésével, netán az audiovonal hangszóróra vezetésével, aminek persze csak erősítés mellett van értelme.



# BASIC és gépi kód

Legutóbb az USR függvény lehetőségeit kiterjesztő rutinok C64-es változatát láttuk. Most — a múltkori assembly lista alapján — részletesebben is megvizsgáljuk a működésüket.

A mostani 1—4. listákban a BASIC interpreter ROM-jának egy-egy részlete látható: azok, amelyekhez az új rutinok kapcsolódnak. A leírásban többnyire az előző részben közölt listák címére hivatkozom, a kivételeket a címtartomány eltérése miatt könnyű felismerni.

## A vektorokat átíró rutin

Működése egyszerű: az A és Y regiszterbe a saját eljárások kezdőcímeinek alsó és felső bájtyát töltjük, és onnan a megfelelő vektorokba írjuk. Illő lenne a vektorok előző tartalmát valahova eltenni, mi most ezt nem tesszük meg. Ezt a — sajnos gyakran alkalmazott — módszert „önző programozás”-nak neveztem el. Hátrányairól, kiküszöböléséről majd később írok.

A \$0308...\$0309 címeken lévő utasítás-értelmezés vektor eredeti tartalma \$A7E4 volt, ez az, amit felülírunk a DEF USR uta-

```
a7e1 6c 08 03 jmp ($0308)
a7e4 20 73 00 jsr $0073
a7e7 20 ed a7 jsr $a7ed
a7ea 4c ae a7 jmp $a7ae
```

### 1. lista

sítást feldolgozó rutin kezdőcímevel. A \$030A...\$030B címeken található kifejezésértékelés vektorba a \$AE86 helyére az USRn függvényeket kezelő eljárás címe kerül.

## A DEF USR feldolgozó rutinja

Az 1. listán látható ROM-részletbe szúrjuk be. Az első utasítás — a CHRGET hívása — mindkét helyen megegyezik.

\$0351...\$035B: a CHRGET-tel beolvasuk az utasítás első szóköztől különböző

bájtyát, és megvizsgáljuk, hogy az a DEF kulcsszó tokenje-e. Ha igen, \$035E-re, a feldolgozás folytatására ugrunk, ellenkező esetben a CHRGET-tal visszaállítjuk a feltételbitekét, és a ROM \$A7E7 címére adjuk a vezérlést. A beszurás helyét követő utasítást kihagyjuk, hiszen a párját az imént hajtottuk végre.

\$0361...\$0365: CHRGET-tel a következő bájtot olvassuk, és megnézzük, hogy nem az USR tokenje-e. Ha nem, akkor a DEF FN utasítást feldolgozó interpreter rutinnal folytatjuk a feldolgozást.

\$0368: az A regiszter tartalmát egy — a nulla lapon levő — szabad tárhelyre tesz-

```
b7f7 a5 66 lda #66
b7f9 30 9d bmi $b798
b7fb a5 61 lda #61
b7fd c9 91 cmp #91
b7ff b0 97 bcs $b798
b801 20 9b bc jsr $bc9b
b804 a5 64 lda #64
b806 a4 65 ldy #65
b808 84 14 sty $14
b80a 85 15 sta $15
b80c 60 rts
```

### 2. lista

```
ae83 6c 0a 03 jmp ($030a)
ae86 a9 00 lda #00
ae88 85 0d sta $0d
ae8a 20 73 00 jsr $0073
ae8d b0 03 bcs $ae92
```

### 3. lista

szük. A tárolt értéknek nincs jelentősége, csak annak, hogy a 7. bit 1-re van állítva, vagyis előjeles számként kezelve negatívnak tekintendő.

\$036A...\$036F: a BASIC program következő bájtyát olvassuk, és a feltételbitek állásától függően háromféle folytatás közül választhatunk. Ezek:

Ha a Z bit 1-re van állítva, akkor utasítás végéhez értünk, ami szintaktikai hibát jelent, ezért a hibarutinra ugrunk. Ha a C bit

értéke 1, ez azt jelenti, hogy az A regiszterbe töltött karakter nem ASCII számjegy. Ekkor az eredeti USR függvény címének definiálását feltételezve az egyenlőséggel meglétező vizsgálatára ugrunk. C=0 esetén viszont a soron következő utasítással folytatódik a program.

\$0371...\$0373: az ASCII számjegyet binárisra alakítjuk. Ez azért megy ilyen egyszerűen, mert a számjegyek ASCII KÓDJA \$30-cal különbözik a bináris értéküktől. A felső félbájty nullázásával éppen a bináris értéket kapjuk meg. Ezt az ASL utasítás segítségével megszorozzuk kettővel. Azért van erre szükség, mert egy olyan táblázathoz fogjuk indexként használni, melynek címei kétféle alakúak.

\$0374...\$0376: a bináris számot a már korábban említett 8 lapos címen tároljuk, egy pozitív értékkel felülírva az eddigi, negatív tartalmat. Ezután a CHRGET-tel a következő karaktert olvassuk az A regiszterbe.

\$0379...\$037D: megvizsgáljuk, hogy az A-ban az egyenlőséggel tokenje van-e. Ha nem, a hibarutinra ugrunk, egyébként a CHRGET-tel újabb karaktert olvassunk be.

```
afd1 20 f1 ae jsr $ae f1
afd4 68 pla
afd5 a8 e6 tay
afd6 b9 ea 9f lda $9fea,y
afd9 85 55 sta $55
afdb b9 eb 9f lda $9feb,y
afde 85 56 sta $56
afe0 20 54 00 jsr $0054
afe3 4c 8d ad jmp $ad8d
```

### 4. lista

\$0380...\$0383: két — korábban már megismert — interpreter rutint hívunk. Az FRMNUM az egyenlőséggel követő aritmetikai kifejezést kiértékeli, és az eredményt a lebegőpontos akkumulátorba (FAC-ba) viszi. A GETADR a FAC tartalmát 16 bites pozitív egész számmá alakítja, és a \$14...\$15 címekre teszi le. A GETADR a 2. listán látható, és megfigyelhetjük, hogy a számot az A és Y regiszterbe viszi át. Mi a következő lépésekben kihasználjuk, hogy a regiszterek tartalma nem változik, tehát közvetlenül a regiszterekből tesszük a címet a táblázatba.

### 5. lista

```
100 rem usrdef C64
110 k=828 : print chr$(147)
120 def fnd(x)=x-48+7*(x>64)
130 for l=1 to 15 : print chr$(19);k;
140 s=0 : read a$,b
150 for i=1 to len(a$) step 2
160 a=16*fnd(asc(mid$(a$,i,1)))+fnd(asc(mid$(a$,i+1,1)))
170 poke k,a : s=s+a : k=k+1 : next
180 if b<>s then print "hiba!" : stop
190 next
200 sys 828 : end
828 data a951a0038d08038c0903a99e,1044
840 data a0038d0a038c0b0360207300,714
```

```
852 data c996f0062079004ce7a72073,1371
864 data 00c9b7f0034cb3b385022073,1343
876 data 00f068b008290f0a85022073,876
888 data 00c9b2d05a207300208a2d0,1199
900 data f7b7a602300a9ddb03989dda,1562
912 data 034ceaa78d1203988d11034c,1031
924 data eaa7a900850d207300c9b7d0,1455
936 data 0f207300f029900ea57ad002,1098
948 data c67bc67a2079004c8da290f,1241
960 data 0a482073002071ae68a8b9da,1351
972 data 038555b9db0385564ce0af4c,1398
984 data 08af48b248b248b248b2,1433
996 data 48b248b248b248b248b2,1250
```



```

100 rem usrdef vc20
110 k=828 : print chr$(147)
120 def fnd(x)=x-48+7*(x/64)
130 for l=1 to 15 : print chr$(19);k;
140 s=0 : read a$,b
150 for i=1 to len(a$) step 2
160 a=16*fnd(asc(mid$(a$,i,1)))+fnd(asc(
mid$(a$,i+1,1)))
170 poke k,a : s=s+a : k=k+1 : next
180 if b<>s then print "hibal" : stop
190 next
200 sys 828 : end
828 data a951a0038d08038c0903a99c,1042
840 data a0038d0a038c0b0360207300,714
852 data c996f0062079004ce7c72073,1403
864 data 00c9b7f0034cb3d385fe2073,1627
876 data 00f066b008290f0a85fe2073,1126
888 data 00c9b2d058207300208acd20,1229
900 data f7d7a6fe300a9dd093989dd8,1842
912 data 034ceac785029885014ceac7,1442
924 data a900850d207300c9b7d00f20,1101
936 data c300f029900ea57ad002c67b,1372
948 data 767a2079004c8dce290f0a48,1034
960 data 20730020f1ce68a8b9d80385,1435
972 data 55b9d90385564ce0cf4c08cf,1507
984 data 48d248d248d248d248d2,1692
996 data 48d248d248d248d2,1128

```

6. lista

```

100 rem usrdef c16
110 k=828 : print chr$(147)
120 for l=1 to 15 : print chr$(19);k;
130 s=0 : read a$,b$ : b=dec(b$)
140 for i=1 to len(a$) step 2
150 a=dec(mid$(a$,i,2))
160 poke k,a : s=s+a : k=k+1
170 next
180 if b<>s then print "hiba" : stop
190 next
200 sys 828 : end
828 data a951a0038d08038c0903a99b,411
836 data a0038d0a038c0b0360207304,2ce
844 data c996f0062079004cd98b2073,535
852 data 04c9b7f0034c9d9a85e82073,5fa
860 data 04f065b008290f0a85e82073,453
868 data 04c9b2d05720730420e19da6,581
876 data e8300a9dd803989dd7034cdc,5d1
884 data 8b8d0205988d01054cdc8ba9,4a6
892 data 00850d207304c9b7d00f2073,41b
900 data 04f029900ea53bd002c63cc6,535
908 data 3b2079044c1e94290f0a4820,280
916 data 730420859468a8b9d7038555,52d
924 data b9d80385564ce954ca1941c,5d9
932 data 991c991c991c991c991c,43e
940 data 991c991c991c99,2b8

```

7. lista

§0386...§0388: a korábban eltt értékét az X regiszterbe töltjük. Ha pozitív, akkor a táblázatban indexelünk vele, ha negatív, akkor az eredeti USR függvény feldolgozásánál folytatjuk.

§038A...§0391: X-szel indexelve az A és Y regiszterekből a definiált címet a cím-táblázatba írjuk, végül visszalépünk az interpreter ciklusba.

§0394...§039B: annyiban különbözik az előző bekezdésben leírtól, hogy a definiált címet — indexelés nélkül — a „hivatalos” USR vektorba írja.

## A USRn függvényeket kiértékelő rutin

§039E...§03A2: ez a három utasítás megegyezik azzal, ami a beszúrás helyén a ROM-ban van; a típusjelzőbe, ami a §0D címen van, θ-t, a numerikus típus kódját töltjük, és a CHRGET rutinnal beolvasuk az A regiszterbe a következő nem szóköz bájtot.

§03A5: megvizsgáljuk, hogy a regiszterben az USR kulcsszó tokenje van-e, ha nem, §03B8-ra ugrunk.

§03A9...§03AE: a CHRGET-tel újabb bájtot olvasunk be, és a feltételbitek állásától függően — ugyanúgy, mint a függvény-definiálást feldolgozó rutinban —, háromféle folytatás közül választhatunk. Ezek:

A Z bit I értéke itt is hibát jelent, a futás a hibarutinon folytatódik. A C bit törölt állapota USRn típusú függvényre utal — a feldolgozó rutin folytatására ugrunk. Ha a C bit be van állítva, akkor viszont az eredeti USR függvényről van szó. Szintaktikai ellenőrzését és feldolgozását az interpreter

bizuk. Ezt készíti elő a §03B0...§03B6 címeiken található utasításcsoport: a BASIC szövegmutató (TXTPTR) értékét egygel csökkenteni.

§03B8...§03BB: a CHRGET rutinnal az aktuális bájttartalmat újra betöltjük az A regiszterbe, és a ROM §AE8D címen folytatódik a feldolgozás. A 3. listáról leolvashatjuk, hogy a §AE86...§AE8A címeiken levő utasításokat átugrottuk, mert feladatukat a mi rutinunk elvégezte. Ne felejtse el, hogy az eredeti USR esetén A-ban az USR tokenje van, mert a §03A9-en levő CHRGET-hívást a TXTPTR visszaállításával hatástalanítottuk.

§03BE...§03C2: a korábban látott módon az ASCII számjegyet binárisra alakítjuk, megszorozzuk kettővel, majd a verembe tesszük megőrzésre. A CHRGET hívásával beolvasuk a következő karaktert, melynek a szintaktikai szabályok szerint nyitó zárójelnek kell lennie.

A következő hét utasítás — nem véletlenül — kísértetiesen hasonlít a 4. listán látható ROM-lista utasításaira, csupán az indexelt LDA utasítások operandusaiban térnek el egymástól.

§03C5: egy olyan ROM-rutint hívunk meg, mely egy zárójelcs kifejezést olvas be a BASIC programszövegéből, közben ellenőrzi a zárójelcs meglétét is.

§03C8...§03D4: a veremből kivesszük a megőrzött sorszámot és az Y regiszterbe tesszük. Az USR táblázatból Y-nal indexelve kiemeljük a hívott függvény címét és a §55...§56 címeiken tároljuk. Befezésül az interpreternek arra az utasítására ugrunk, amely a függvény kiértékelése után folytatja az azt tartalmazó kifejezés kiértékelését.

§03D7: közös hibarutin, közvetlen ugrás az interpreter SYNTAX ERROR-t kiíró utasítására.

## A betöltőprogramok

Az 5—7. listákon a BASIC nyelvű betöltőprogram három géptípushoz tartozó változata van. Újszerű az adatok formája. Ennek feldolgozása a korábbinál lassúbb, de a begépelésnél fáradságos takarítunk meg. A programok működéséről a következő alkalommal írok részletesebben, most csak néhány megjegyzést teszek.

A betöltési idő kicsit több 6 másodpercnél, a C16-os változaté kb. a fele. A merev képernyő előtt őrve ez nagyon hosszúnak tűnik, ezért a képernyő sarkában mindig kiíródik a feldolgozás alatti DATA utasítás sorszáma. Soronként halad az ellenőrzés, hiba esetén a hibás adatot tartalmazó sor száma után megjelenik a „hiba!” felirat, és a program megáll. Javítás után újrainditható.

Futtatás előtt célszerű a programot tárolni; különösen fontos ez, ha mágnesszalagos egységgel dolgozunk. A kazettapufferban való elhelyezés miatt a gépi kódú rutinok aktivizálása után a magnó használatra a rendszer megbénulásához vezethet. Ennek magyarázatát és elkerülésének módját a következő részben adom meg.

C16-on a DEF USR utasítás parancs módban csak akkor működik, ha elé vagy egy másik utasítást, vagy legalább egy ketőspontot írunk. Ennek okáról ugyancsak a következő alkalommal írok.

BARNA LÁSZLÓ



**A kategóriájú kereskedelmi vállalat szervezési osztálya felvesz közép- vagy felsőfokú végzettségű**

- **OPERÁTOROKAT**
- **PROGRAMOZÓKAT**
- **RENDSZERSZERVEZŐKET**

**K+F témában megvalósuló operatív (online) áruforgalmi rendszer megvalósításához, új számítóközpontba, TPA 11/440 számítógépes környezetbe**

**Hosszú távú koncepció!**

**Továbbképzési lehetőségek!**

**Rugalmas munkaidő!**

**Jelentkezés a 121-608-as telefonon, vagy részletes szakmai önéletrajzzal személyesen, Kereszty Péter osztályvezetőnél.**

**Cím:**

**TEXÉRT Kereskedelmi Vállalat**

**Budapest V., Alpári Gy. u. 17.**



ENERGIAGAZDÁLKODÁSI

**CASSYS**  
INTÉZET

## ELVIHETI, KIPRÓBÁLHATJA

és ha megfelel Önnek,  
már 60 ezer forintért  
(+ ÁFA)  
megvásárolhatja

a

**BÉR-ADÓ című,**

**IBM PC-re készített  
bérszámfejtő és személyi-  
jövedelemadó-elszámoló  
rendszerünket.**

**Energiagazdálkodási Intézet**

Budapest, Bem rkp. 33—34. 1027  
Telefon: 354-165, 153-633

**Számítástechnikai  
Informatikai  
Fejlesztő Vállalat**



**Kereskedelmi Irodájának  
vezetésére**

- **felsőfokú iskolai végzettséggel,**
- **számítástechnikai ismerettel és kereskedelmi gyakorlattal,**
- **legalább öt év vezetői gyakorlattal,**
- **nyelvismerettel**

**rendelkező munkatársat keres.**

**Jelentkezés:**

Marjai Sándor gazdaságigazgató-helyettesnél, a 353-580 vagy a 350-140/103-as telefonszámon.



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

VEVŐSZOLGÁLAT

VÉTEL-ELADÁS

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

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

**JAVÍTÁSA, KARBANTARTÁSA  
VIDEOTON TVC-számítógép márkaszerviz**

**ÁTALÁNYDÍJAS JAVÍTÁSI  
ÉS KARBANTARTÁSI SZERZŐDÉS**



**ÖRÖK  
GARANCIA**

**SZERVIZEINK:**

1053 Budapest V., Magyar u. 12—14. Telefon: 173-551  
Telex: 22-7621

1083 Budapest VIII., Szigony u. 11. Telefon: 343-153

7623 Pécs, Kolozsvár u. 20. Telefon: (72) 11-812

9700 Szombathely, Szalonok u. 31. Telefon: (94) 14-519

6726 Szeged, Székelysor 13. Telefon: (62) 13-377

5600 Békéscsaba, Bartók Béla u. 37. Telefon: (66) 27-195

3525 Miskolc, Fazekas u. 1—3. Telefon: (46) 17-011

3100 Salgótarján, Arany János út 3. Telefon: (32) 14-007

4034 Debrecen, Holló László u. 14.

7400 Kaposvár, Füredi u. 24. Telefon: (82) 16-307

FOTOELEKTRONIK

— NOVOTRADE

— ALFA GT



# Bináris fák III.

## Kiegyensúlyozott bináris fát kezelő algoritmus



### Az algoritmust megvalósító program változó

Tekintsük az I. részbeli 3. ábrát (lásd 1988/2. számunkban — A szerk.). A mellékelt lista szerinti program a fa csúcspontját ilyen elvek szerint tárolja. Ehhez be kell vezetni a megfelelő változókat és tömböket:

- $ms$  — a tömbök felső határa
- $a\alpha(ms)$  — az azonosítók tömbje
- $b\%(ms)$  — a bal mutatók tömbje
- $j\%(ms)$  — a jobb mutatók tömbje
- $x\alpha$  — a keresett azonosító
- $j\%(0)$  — az első üres tömbelem indexe, beépítés után a beépített elem indexe:  $j\%(0) - 1$
- $b\%(0)$  — a gyöker indexe

Az egyensúlyozás kezeléséhez még szükségünk van a csúcsokhoz tartozó egyensúlyi állapotok (= a jobb részfa magassága — a bal részfa magassága)

- ha  $h(T1) < h(T2)$ , akkor  $e\%(i) = 1$
  - ha  $h(T1) = h(T2)$ , akkor  $e\%(i) = 0$
  - ha  $h(T1) > h(T2)$ , akkor  $e\%(i) = -1$
  - ahol  $T1$  a bal,  $T2$  a jobb részfája  $t$ -nek, és  $h(fa)$  a fa magassága
  - $v(yM)$  — egy  $ym$  elemű verem, a keresőút előző szintjeihez tartozó csúcsok indexeinek tárolására ( $ym$ : a verem mérete)
  - $vp$  — veremmutató
- Munkaváltozók:
- $bj$  — beépítésjelző (ha  $x\alpha$  nincs a fán, akkor  $bj = 1$  esetén le van tiltva a beépítés)
  - $i$  — keresés, beszúrás, törlés, bejárás futóindexe
  - $ix$  — keresésből való visszatéréskor a megtalált vagy a beépített elem indexe a tömbben
  - $j$  — a beépített, ill. törölt elem indexe
  - $h$  —  $1$ , ha beszúrásnál a részfa magassága növekedett vagy törlésnél a részfa magassága csökkent

- $i1, i2, e1$  — az átindexelések segédváltozói
- $e2$  — egyensúlyozásnál
- $vq$  — ha törölnél a törlendő csúcson két részfája van, akkor a keresőutat folytatni kell a bal részfa jobb szélső eleméig. Ilyenkor  $vq$  marad a törlendő csúcson megfelelő veremelemnél,  $vp$  mutatja a verem tetejét.

### A program részei

A program áttekintéséhez az alábbi részletezést készítettük (lásd a táblázatot).

### A program alkalmazása

A program nem használ nyomtatót, a bináris fához nem kapcsolódik lemezfájl. Ezt a kapcsolódást, továbbá képernyőmaszkhoz való kapcsolódást, amennyiben érdeklődésre tart számot, külön leírásban részletezem. A 3000—4300-ig tartó binárisfa-kezelő rutinok bármely gép BASIC-jéhez könnyen igazíthatók, de a fájl és a képernyőkapcsolat már erősen gépfüggő. Az eredeti programban if-then-else szerkezetű a 3190, 3200, 3370, 3380, 3510, 3560, 3570, 3660, 3710, 3720, 3840 és a 4320-as sorszámu utasítás.

A működés lépései az alábbiak:

- tömbméret kérése
- menü
- keresés: keresendő azonosító kérése ha megtalálta: szerepel, index =  $xx$  ha nem találja: nem szerepel visszatér a menü
- beszúrás: beszúrandó azonosító kérése ha megtalálta: szerepel, index =  $xx$  ha nem találja: beépítve, index =  $xx$  visszatér a menü

Tevékenység	Programsorok		
1. Demonstrációs környezet	3000 alatt	3.4. keresés a kisebb ágon (veremelés rekurzív hívás egyensúlyvizsgálat)	3920-3960
2. Keresés, beszúrás	3000-3430	3.5. keresés a nagyobb ágon - " -	3980-4030
2.1 $x\alpha$ nincs a fán, $bj=0$ esetén beépítés	3010-3060	3.6. kiiktatás, ha mindkét részfa üres	4045
2.2 $x\alpha$ kisebb, mint a vizsgált csúcs azonosítója	3080-3230	3.7. kiiktatás, ha a jobb részfa üres	4040-4060
2.2.1 verembe, rekurzív hívás, veremből	3080-3100	3.8. kiiktatás, ha a bal részfa üres	4070-4080
2.2.2 bal oldali ág magassága nőtt	3120-3230	3.9. kiiktatás, ha egyik részfa sem üres	4090-4140
2.2.2.1 nem kell egyensúlyozni	3120-3130	3.10. törlés szubrutinjai	
2.2.2.2 BB-forgatás	3140-3160	3.10.1. bal ág magassága csökkent	3450-3580
2.2.2.3 BJ-forgatás	3180-3210	3.10.1.1. nem borult fel az egyensúly	3460-3470
2.3 $x\alpha$ nagyobb, mint a vizsgált csúcs azonosítója	3250-3410	3.10.1.2. JJ-forgatás	3500-3520
2.3.1 verembe, rekurzív hívás, veremből	3250-3270	3.10.1.3. JB-forgatás	3540-3580
2.3.2 jobb oldali ág magassága nőtt	3290-3410	3.10.2. jobb ág magassága csökkent	3600-3740
2.3.2.1 nem kell egyensúlyozni	3290-3300	3.10.2.1. nem borult fel az egyensúly	3610-3620
2.3.2.2 JJ-forgatás	3310-3330	3.10.2.2. BB-forgatás	3650-3670
2.3.2.3 JB-forgatás	3350-3390	3.10.2.3. BJ-forgatás	3690-3730
2.4 $x\alpha$ egyenlő a vizsgált csúcs azonosítójával	3420-3430	3.10.3. törlés, ha a törlendő csúcs egyik részfája sem üres	3750-3880
3. Törlés	3440-4190	3.10.3.1. ha jobb szélső elem, ugrás az áthelyezésre	3760
3.1. törlés meghívás	3891	3.10.3.2. bal részfa jobb szélső elem keresés	3750-3790
3.2. üres elem képzése, felfűzés a láncra	3893	3.10.3.3. szükség esetén egyensúlyozás	3800
3.3. $x\alpha$ nincs a fán	3900	3.10.3.4. a jobb szélső elem kiiktatása, törölt elem helyébe való beiktatás	3820-3880
		3.10.4. többször ismétlődő tevékenység	4150-4190
		4. Inorder bejárás	4310-4360
		5. Demonstrációs környezet	4195-4300
			4400-től



```

16 ren keferendo valtozok
20 c1#="#####" c2#="#####"
30 poke 53200,0 : poke 53201,0 : print chr
  c(5)
40 print c2# : input " tomb meret =
  n :
50 dim a$(ns),b$(ns),j$(ns),e$(ns)
60 j$(0)=1
70 ren menu
80 print c1# : "m e n u "
90 print "1 1. kereses"
100 print "2 2. beszur-as"
110 print "3 3. torles"
120 print "4 4. listaz-as"
125 print "5 5. bejar-as"
130 print "6. vese"
140 get kds : if kds="" then 140
150 kd=val(kds) : if kd<1 or kd>6 then 14
  0
160 on kd gosub 190,230,270,400,415
165 if kd=4 or kd=5 then gosub 4500
170 if kd=6 then end
180 goto 80
190 print c2# : input "Keresendo azonosito
  to : " :
200 b=a$ : gosub 3000 : b#=""
210 if i=b# then hs=1 : gosub 5000 : retur
  n
220 hs=2 : gosub 5000 : return
230 print c2# : input "Beszur-ando azonosito
  to : " :
240 if j$(0)hs then hs=5 : gosub 5000 : re
  turn
240 gosub 3000
250 if i=b# then hs=2 : gosub 5000 : retur
  n
260 hs=3 : gosub 5000 : 0# : return
270 print c2# : input "Torlendo azonosito
  to : " :
280 gosub 3090
290 if i=b# then hs=1 : gosub 5000 : return
300 hs=4 : gosub 5000 : j# : return
400 gosub 4400
410 sk#=""
420 for sl=0 to ns
430 if sk(2k) then 460
440 gosub 4500
450 sk=1 : gosub 4400
460 print sl:tab(6);sk(sl):tab(17);b$(sl)
  :tab(25);j$(sl):tab(32);e$(sl)
470 sk=sk+1
480 next sl
490 return
2990
3000 ren k e r e s e s , b e s z u r a s
3005 i=b$(0) : i#=""
3010 if i<0 goto 3070
3015 if b#i then i#="" return
3020 i=j$(0) : j#=""
3025 if i<0 goto 3040
3030 j$(0)=j$(0)+1
3040 i#(i)=i#(i)+1 : j$(i)=0 : e$(i)=0
3050 h#=""
3060 goto 4150
3070 if i#=#(i) goto 3240
3080 v#=#(i) : v$(v#)=1 : i=b$(i)
3090 gosub 3010
3100 i#(v#)=v#(v#)+1
3110 if h<1 or b#i then return
3120 if e$(i)=1 then e$(i)=0 : h#="" retur
  n
3130 if e$(i)=0 then e$(i)=1 : return
3140 i=i+b$(i)
3150 if e$(i)>1 goto 3180
3160 b$(i)=j$(i) : j$(i)=1 : e$(i)=0 : i =
  i+1
3170 goto 3220
3180 i2=j$(i1) : j$(i1)=b$(i2) : b$(i2)=i1
  : b$(i1)=j$(i2) : j$(i2)=i
3190 e$(i1)=0 : if e$(i2)=1 then e$(i1)=1
3200 e$(i1)=0 : if e$(i2)=1 then e$(i1)=1
3210 i=i+2
3220 e$(i)=0 : h#=""
3230 goto 4150
3240 if i#=#(i) goto 3420
3250 v#=#(i) : v$(v#)=1 : i=j$(i)
3260 gosub 3010
3270 i#(v#)=v#(v#)+1
3280 if h<1 or b#i then return
3290 if e$(i)=1 then e$(i)=0 : h#="" retu
  n
3300 if e$(i)=0 then e$(i)=1 : return
3310 i=i+b$(i)
3320 if e$(i)>1 goto 3350
3330 j$(i)=b$(i1) : b$(i1)=i : e$(i)=0 : i =
  i+1
3340 goto 3400
3350 i2=b$(i1) : j$(i2)=i1 : j$(i1)=b$(i
  2) : b$(i2)=i
3360 e$(i1)=0 : if e$(i2)=1 then e$(i1)=1
3370 e$(i1)=0 : if e$(i2)=1 then e$(i1)=1
3380 i=i+2
3390 e$(i)=0 : h#=""
3400 e$(i)=0 : h#=""
3410 goto 4150
3420 h#="" : i#=""
3430 return
3435
3440 ren t o r l e s , m u t i o s
3450 ren e 2 : v e s e 1 : bal as csokkent
3460 if e$(i)=1 then e$(i)=0 : return
3470 if e$(i)=0 then e$(i)=1 : h#="" retur
  n
3480 i=j$(i) : e1=e$(i1)
3490 if e1<0 goto 3540
3500 j$(i)=b$(i1) : b$(i1)=i
3510 e$(i1)=0 : if e$(i1)=0 : if e1=0 then e$(i)
  =1 : e$(i1)=1 : h#=""
3520 i=i+1
3530 goto 4150
3540 i2=e$(i1) : e2=e$(i2)
3550 b$(i1)=j$(i2) : j$(i2)=i1 : j$(i)=b$(
  i2) : b$(i2)=i
3560 e$(i1)=0 : if e2=1 then e$(i1)=1
3570 e$(i1)=0 : if e2=1 then e$(i1)=1
3580 i=i+2 : e$(i2)=1
3590 goto 4150
3600 ren e 9 : v e n 2 : jobb as csokkent
3610 if e$(i)=1 then e$(i)=0 : return
3620 if e$(i)=0 then e$(i)=1 : h#="" retu
  n
3630 i=b$(i) : e1=e$(i1)
3640 if e1<0 goto 3690
3650 b$(i)=j$(i1) : j$(i1)=i
3660 e$(i1)=0 : e$(i1)=0 : if e1=0 then e$(i)
  =1 : e$(i1)=1 : h#=""
3670 i=i+1
3680 goto 4150
3690 i2=j$(i1) : e2=e$(i2)
3700 j$(i1)=b$(i2) : b$(i2)=i1 : b$(i)=j$(
  i2) : b$(i2)=i
3710 e$(i1)=0 : if e2=1 then e$(i1)=1
3720 e$(i1)=0 : if e2=1 then e$(i1)=1
3730 i=i+2 : e$(i2)=1
3740 goto 4150
3750 ren ket reszfa van
3760 if j$(i)=0 goto 3830
3770 v#=#(i) : v$(v#)=1 : i=j$(i)
3780 gosub 3750
3790 i#(v#)=v#(v#)+1
3800 if h#="" then gosub 3600
3810 return
3820 if v#>0 then b$(0)=i : goto 3950
3830 if x#=#(v$(v#)) then b$(v$(v#))=i : go
  to 3850
3941 j$(v$(v#))=i
3950 if v#>v#-1 then j$(v$(v#))=b$(i) : b$(
  i)=b$(j)
3960 j$(i)=j$(i) : e$(i)=e$(j) : v$(v#)=i+1
3970 h#=""
3980 return
3985
3990 ren t o r l e s , b e l e p e s
3991 gosub 3095
3992 if j#="" then return
3993 i#(i)=i#(i) : b$(i)=0 : j$(i)=j$(0) : j$(0)
  =i : e$(i)=0
3994 return
3995 i=b$(0)
3996 if i=0 then j#="" : h#="" return
3997 if x#=#(i) goto 3970
3998 v#=#(i) : v$(v#)=1 : i=b$(i)

```

- törles: törlendő azonosító kérés ha megtalálta: törölve, index==xx ha nem találja: nem szerepel visszatér a menü
- listázás: a felvitel sorrendjében felíródik a képernyőre a csúszhoz tartozó sorszám (tömbindex), azonosító, bal mutató, jobb mutató, az I. részbeli 3. ábrához hasonlóan
- bejárás: rendezetten íródik fel a képernyőre a fa azonosító
- A listázásnál és a bejárásnál minden 20 elem után, illetve a végén egy billentyű megnyomását követően folytatódik a lista, illetve tér vissza a menü.

**Megjegyzések:** A fa egy csúcának törlése

után az üresen maradt helyeket az algoritmus láncba szervezi, majd újra bevonja a feltöltésbe. J%(0) az utóljára törölt elem indexét tartalmazza, annak jobb mutatója az utolsó előtti törlés indexét mutatja stb. A lánc végén lévő elem jobb mutatója az első olyan tömbelemre mutat, amely még nem volt feltöltve. Ily módon a köztes üres elemek újrafeltöltésénél nem nyúl a program új elemhez. Ebből következően is igaz, hogy maximum annyi helyre van szükség, ahány különböző elem szöve jöhet a fán, akárhány műveletet is végzünk. Ily módon valósul meg a dinamikus helyfoglalás.

A program jó néhány rekurzív hívást tartalmaz. Itt a visszatérési címeket a BASIC

által kezelt verem tartalmazza. Az algoritmus 10 szintű fánál 9 visszatérési cím tárolását igényli, az egymásba ágyazott GO-SUB, illetve FOR utasítások száma legfeljebb 3. Ez még a C64 gép esetében is (vérem hossza: 256) lehetővé teszi a szokásos maximum 4-5 egymásba ágyazású programhoz való kapcsolást. A bejárásoknál a csúcsok megőrzésére a program készít vermet, mivel a dimenzionális automatikus, esetünkben legfeljebb 10 szintű fa hozható létre, ez maximum 1023 elem. A gép 2.0 BASIC tömbkezelési sebességet ismerve maximum 1000 elem kezelése javasolt C64 gépen.

FLENDER ANDRÁS



# Személyi

A Macintosh óta sok-sok személyi és házi számítógép mondhatná ezt magáról, de azon sem csodálkoznánk, ha piktogram-vezérlésűnek minősítve magukat, dicsekednének.

Szakmai körökben igen divatos szó.



„En nem írok, nem olvasok.  
én egy ikonutasításos  
számítógép vagyok!”

azért, mert a gépre írt egyik ikonvezérlő szoftver már az 1.1-es változatában olyan hibátlannak ígérkezett, hogy azt EPROM-os alakban is forgalmazták; főképp mégis azért, mert megjelenésével rögtön nagyot emelt a gép emberközelségén, és ezzel sokat javított akkori rossz piaci helyzetén.

Akik ismerik a QL finoman kidolgozott, néha eseten, gyakran túl cizellált utasítási szintaxisait — legyünk őszinték: a Tony Tebby által megírt Toolkit II ezeken kicsit egyszerűsített —, azoknak üdvözlőniük kell minden könnyítést. Nem véletlen, hogy kedvezően fogadták az olyan törekvést, ami végeredményként a bonyolult utasítási rendszert megkerülve, lehetővé teszi a műveletek kényesebb elvégzését is, egy vagy két billentyű lenyomásával (kurzor esetén nyomva tartásával).

QL-re több mint fél tucat okosabb és kevésbé okos ilyen rendszer kapható már. Ezek képességeikben sokszor átfedik, más-

kor jól kiegészítik egymást, és így a felhasználónak kedve lenne egyiket-másikat össze- gyúrni, ami persze már nem is olyan egyszerű dolog.

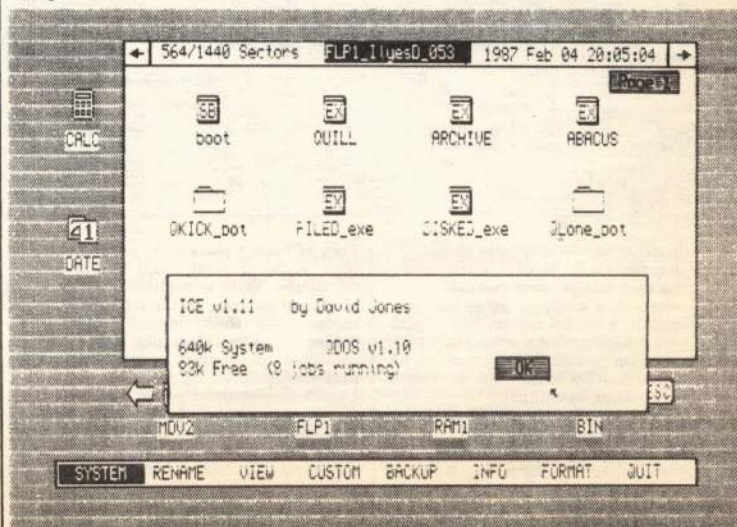
## Evolúció

A QL-re megjelent első ilyen rendszer, amit később részletesen is tárgyalunk, az *Eidersoft ICE* (ejtsd: ájsz). Szerzője, David Jones, méltán büszke lehet művére. Az ICE betűszó, és az *Icon Controlled Environment* szavak kezdőbetűiből lett. A név kb. annyit tesz, hogy ikonnal vezérelt környezet. Miután ez kevesebb mint 16 kb-át memóriaterületet igényel, ezért nemcsak szoftverben, hanem a QL-hez hátulról becsatlakoztatható ROM-kártyán is kapható. A QL memóriaterképén a 0C000 és 0FFFF közötti területet foglalja el. Ehhez a szoftverhez és hardverhez természetesen botkormányt, illetve egeret is kínálunk a kereskedők.

## Új viszonyulás

Egy házi számítógép ma már nem is számít valamirevalónak, ha rendszerét csak pontosan a gép helyesírásának (szintaxisának) megfelelően begépelte utasításokkal lehet vezérelni. Mi több, nem is lehet személyi számítógépnek igazi piaci sikere anélkül, hogy ne lenne úgynevezett *front-ends*, amilyen például a Microsoftnak a GEM, az Atarinak a TOS, az Apple-nek a MAC-(DOS). A legfejlettebb Commodore 128-asnak már a ROM-jába van beépítve a GEOS program, ami több is, mint egyszerű ikonvezérlő. A drágább gépektől pedig egyenesen mindenki elvárja, hogy rendelkezzenek ikonutasításos rendszerrel, mert ez leegyszerűsíti az utasítások bevitelét, és gyakran olyasmikre is lehetőséget nyújt, amik egyébként BASIC-beli utasításokkal nem fogalmazhatók meg. Az egyik legjobban átgondolt példa erre a Macintoshra írt GEM, amely szinte szabvánnyá vált a képernyőkezelő ikonutasításos rendszerek között, és teljes egészében ikonok és mozgatható (de főképp lehúzható) ablakok, menükombinációjából áll. Az ikonutasításos rendszer, ha nem is nevezhető generációváltásnak, de mindenképpen szemléletváltás: új felületet adott a felhasználóknak az ember-gép kapcsolat értelmezéséhez és ezáltal az áttértekléséhez is.

1. kép



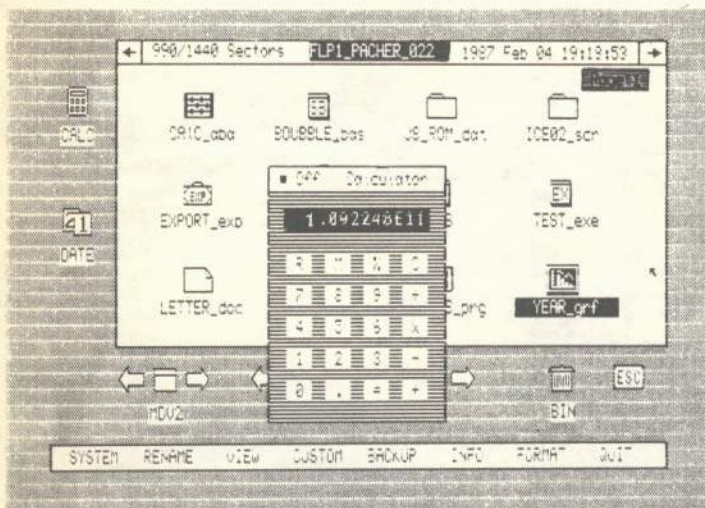
## Példaválasztás

Hogy egy ilyen ikonutasításos rendszert közelebbről szemügyre vehessünk, konkrét géphez és szoftverhez kell fordulnunk. A választás a közel sem legelterjedtebb gépre, a Sinclair QL-jére esett — talán



# számítógépek ikonutasításos rendszere

a WIMPS (Windows, Icons, Mice, PointerS)  
Jelentése: ablakok, ikonok, egerek, mutatók.



2. kép

Ezután jelent meg a nyugatnémet ABC Elektronik gondozásában, de elsőként Angliában piacra dobott *Giga-Desk* és *E.A.S.E.* (Easily Applicable System Environment), ami nemcsak teljes operációs környezetet, de egy két nyomógombos egeret és egy *Giga-Basic* szoftvert is kínál.

Később jött az *Ultrasoft QKICK*-je, ami nem kimondottan front-end, mivel a leg-egyszerűbbet — egy szoftver behívását és aktivizálását — nem tudja. Ugyanakkor olyan finomságokra is képes, mint az automatikus képernyő ki-be kapcsolás (auto-screen on-off), természetesen állítható türelmi idővel, ami által a monitor kímélődik. Ezenkívül található benne határidőnapló és jegyzetfüzet mini editorral, hogy akik az IBM vagy az Apple kínálta Sidekick iránt éreznek nosztalgiát, azok ne pusztán a Qkick nevében találjanak rá utalást.

Nagy port vert fel az ugyancsak az Ultrasoft gondozásában kiadott *JAM*, amiről

először mindenki azt hitte, hogy a nevével a GEM-re szeretne hasonlítani, de mint kiderült, ez a Job Application Manager betűszava. Saját belső dinamikus RAM-lemeze különösen kényelmessé teszi a használatát.

Nagyon érdekesnek ígérkezett a Cope Software ház *QATS*-e, ami csak a felületes fantáziát emlékeztetheti a „macskák”-ra; jelentése *QL Applications Traffic Supervisor*. Érdekessé sokoldalú printervezérlő képessége és megbízható *wild card* (Joker) felismerő tulajdonsága tette. Rövidsége, tömörsége miatt ugyancsak kapható ROM-modul formában is.

Meg kell még említenem a QL-es berkekben jól ismert Talent szoftverház termékét, a *QIMP*-et (*QL Icon Management Program*).

És végül itt a *QRAM*, a legújabb front-end, amit Tony Tebby neve fémjelmez. A tervek szerint a Sandy által fejlesztett SuperQ-board új bővítőjébe akarják beépíteni, és így ez — azok után, hogy 512 kb-ajos me-

móriabővítőmenyen kívül lemez meghajtó-vezérlőt, párhuzamos printerillesztőt, RAM-lemezt és még egy olyan Toolkit II-t is tartalmaz, amire még Clive Sinclair is büszke lenne — valóban méltó lesz a szuper jelzőre.

Úgy tűnik, mind közül az utolsó a legjobb, de még külföldön sem a legelterjedtebb. Az eladott példányszámot tekintve első — a legtöbb QL-es által használt front-end — az *ICE*.

## Mire jó az ICE?

Tulajdonképpen grafikus burokban nyújtja át a felhasználónak a bonyolult szintaxisokat, hogy ezzel megkímélje azok megtanulásától, illetve időrabló bebillentyűzésétől. A gép bekapcsolása után (feltételezve, hogy az ICE ROM-kiegészítőként van a QL erre kialakított hátsó csatlakozójába bedugva), F1 vagy F2 lenyomására a BASIC ablakok helyett azonnal a munkaterület jelenik meg a kis alakzatokkal (ikonokkal), amelyek a QL-hez csatlakoztatható eszközöket, a microdrive-okat és lemez meghajtókat, illetve a bármikor „elővehető” naptárt, számológépet és az ugyancsak „kénél levő” szemétkoszarat jelképezik. Ha azt szeretnénk elérni, hogy az ICE helyett rögtön a háttértárolón levő *BOOT* nevű program induljon el autostarttal, akkor a QL reset utáni feléledése alatt az *ALT* gombot tartuk nyomva. Ha ezt elmulasztjuk, akkor sem kell ismét resetelnünk, hiszen az alsó munkalécen azért van a *QUIT*, hogy azzal visszamehessünk a régi jó *BASIC*-ba.

## Tudnivalók a kezelésről

Az *ICE* használatához a kezelőnek semmi mást nem kell tennie, mint a négy kurzorbillentyű (jobb esetben a botkormány vagy az eger) segítségével a képernyőn található nyilat a kívánt ikonra kormányozni. A kurzorbillentyűk, illetve a botkormány nem pótolja teljes mértékben az egeret, mert bár a nyíl sebessége a későbbiekben leírt módon beállítható, a legnagyobb sebesség mellett sem helyettesítik az eger azon képességét, hogy sebessége menet közben változtatható, és így az ikon durva megközelítésekkel sokkal gyorsabban lehet a nyilat mozgatni, mint az ikonhoz érve.

Ha a kis nyíl valamelyik ikonon vagy annak közvetlen környezetében van, egyszerűen a *SPACE* billentyűt kell megnyomni egyszer vagy gyors egymásutánban kétszer aszerint, hogy mit szeretnénk elérni. Ha például valamelyik háttértár (microdrive, lemez meghajtó, RAM-lemez) ikonján egy kettős leütést végzünk a *SPACE* billentyűvel vagy botkormány esetén a tűzgombbal, akkor ezzel a *dir.eszköz*+*ENTER* begépe-



lést karantinjuk meg. Könnyen betárolhatjuk, hogy ez például egy dir mdvl + [ENTER] esetén is 10 billentyű helyett csupán a SPACE kétszeri leütése.

## Ikonkészlet

Mindegyik háttértár ikonjának jobb és bal oldalán nyíl van, amelyre SPACE-t ütve 1-től 8-ig növelhetjük bármelyik háttértár sorszámát, mivel a QL egyidejűleg maximum nyolcat tud bármelyik háttértárból kezelni. A CUSTOM (lásd lejjebb) segítségével az is megoldható — ha nem használunk lemezmeghajtót vagy RAM-lemezt —, hogy ezek helyét kölcsönvegyük a micro-drive számára: átkereszteljük az FLP-t vagy a RAM-ot MDV-re, és a számát SPACE leütéssel 2-re változtatjuk. Csak micro-drive-ot használva így nem kell lépten-nyomon átszámolni az ikont, elég, ha az ikonok között válogatunk.

A dir-t behívó kettős SPACE leütése után a munkaterületen, annak egyetlen lapján egyszerre 12 fájl jelenik meg más-más szimbólumú ikonokkal, aszerint, hogy az Super Basic fájl, Exec fájl vagy a négy felhasználói Psion program fájlja. Ezeket nemcsak az ikon alatti fájlnev kiegészítője különbözteti meg:

QUILL esetén \_doc  
 ABACUS esetén \_prg  
 ARCHIVE esetén \_pro, \_dbf  
 EASEL esetén \_grf

hanem maga az ikon is. Külön ikonja van még a Psionok közötti exportálható-importhálható \_exp kiegészítésű fájloknak, az ICE által fel nem ismert típusú fájlknak és végül a csak ICE alatt futtatható CHOICE-ban definiált \_tsk kiegészítésű programcsomagoknak (lásd lejjebb).

Vigyázat! Az ikon alatti fájlnev csakán jelenik meg, ha annak utolsó kiegészítő négy karakterét nem számolva a fájl neve hosszabb, mint 8 karakter. A csonkított fájlnev az INFO segítségével teljes hosszában előhívható. A munkaterület jobb és bal felső sarkába víve a kurzort, egy SPACE leütéssel lehet egyet előre vagy hátra lapozni, de ha sokszor 12 fájlt tartalmaz az adathordozó, és a kezelő tudja, hogy a kívánt lap messze van, akkor a SPACE kettős leütésével négy lapot is lapozhat egyszerre.

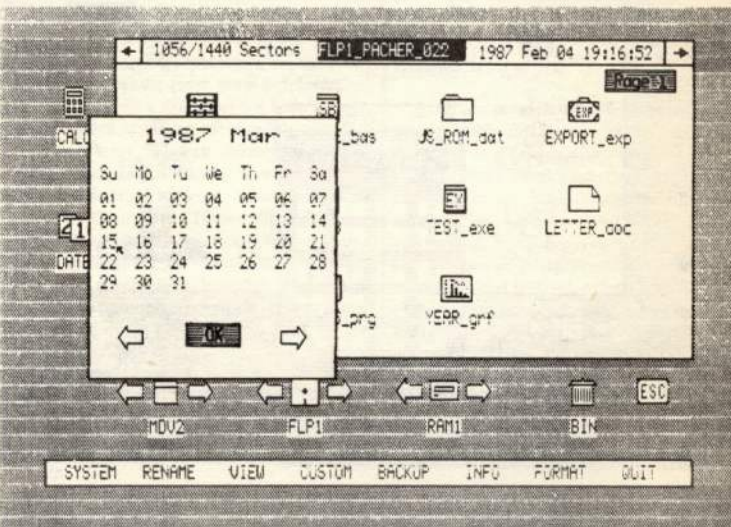
## A munkalécek funkciói

A munkaterület felső munkalécén még olyan fontos információk olvashatók, mint az adathordozó fajtája, neve, az összes és a még szabad szektorok (512 bájt) száma, valamint a gép tudomása szerinti dátum és pontos idő; ez utóbbi külön jobkint fut.

A SPACE billentyű kettős leütésével a program — függetlenül attól, hogy BASIC

program vagy EXEC-ként indítható job — arról az eszköztől hívódik és indul el automatikusan, amelyik eszköz a tartalomjegyzéket a képernyő munkaterületére adta. Sajnos az ICE egyik nagy hibája, hogy ennél kevesebb nem létezik. Az ICE kettős szöközleütése helyettesíti ugyan az egész lrun mdvl\_név\_bas vagy exec\_w flpl\_név\_exe szintaxist, de nem tudja megoldani az edításhoz elengedhetetlenül szükséges egyszerű load mdvl\_név parancs végrehajtását. (Igaz, hogy a gyakorlott programozón ez nem fog ki, mert ha gyakran szeretné szerkeszteni a programját, de az ICE-szal dolgozik, akkor beír egy l PAUSE utasítást, és ha edítálni akar, akkor a beolvasás után BREAK-ét nyom, ha pedig nem, akkor SPACE-t; ezzel a problémát feloldotta.)

tum — SET TIME/DATE — beállítás, aminek hibája, hogy nem hagyható el, csak akkor, ha végigjártaszott az egész beállítást; a géphez csatlakoztatott eszközök meghívási neveinek — CHANGE DEVICE — megváltoztatása (például flpl helyett fdkl), a printer csatlakoztatási módjának — DEFINE PRINTER — (par, ser), valamint az adatátvitel sebességének (Baud rate) beállítása, a SET CONTROL segítségével a kurzornyal sebességének és a billentyűismétlődés késedelmi idejének a beállítása. Mindezen beállítások (kivéve a kurzor mozgatási sebességét) ugyancsak az ablakban kínálózó — SAVE CHOICES — segítségével menthetők a háttértár adathordozójára, hogy a kikapcsolási utáni bármelyik újbóli üzembe helyezéskor a CUS-



3. kép

Az alsó munkalécen látható a SYSTEM-re egy SPACE-t ütve a munkaterület közepén megjelenő ablakban az ICE közli saját verziószámát, a QDOS verziószámát, a gép összes és még szabadon felhasználható RAM területét kbájtban, valamint azt, hogy a gépben jelenleg hány job fut egyszerre. Ezt az ablakot az összes információjával az ablakban található O.K. feliraton való egy SPACE leütéssel lehet a munkaterületről eltüntetni (lásd az 1. képet).

Ugyancsak az alsó munkalécen található a CUSTOM, ami egy SPACE leütéssel szintén ablakot hív a munkaterületre, hogy újabb, a kurzorbillentyű fel-le gombjaival kiválasztható lehetőségeket kínáljon a felhasználónak. Ilyen lehetőség az idő és dá-

TOM alatt található — LOAD CHOICES — úján visszahívhatók legyenek a gépbe, sőt az ICE első aktivizálódásakor automatikusan meg is hívja a kiemelt háttértárról a del\_dfb néven tárolt CHOICE-ot, így azt első alkalommal nem szükséges kézzel aktivizálnunk. A CUSTOM ablakból a kurzor CANCEL-ra mozgatásával és egy SPACE leütéssel szállhatunk ki.

Körültekintően dolgozták ki a munkalécen szereplő FORMAT lehetőséget is. Egy külön ablak által nemcsak arra szólítanak fel, hogy jelöljük ki az eszközt, amelyen formattálni szeretnénk, de még az adathordozó keresztelje előtt külön figyelmeztet a rendszer, hogyan kell kiszállnunk az egészről, „amíg nem késő”.



A BACKUP megadja, hogy az adathordozón, amiről a tartalomjegyzéket kértük, összesen hány fájl van, és azt hajlandó is válogatás nélkül akármelyik általunk kijelölt eszközben levő adathordozóra átmásolni. Ha netán ugyanarra az eszközre szeretnénk a másolatot készíteni, mint amelyiken a forrás van, akkor ezt is meg tudja oldani, de időnként kéri a forrásszerepű és a másik adathordozó cseréjét a közös eszközben. Ez nyilván attól függ, hogy puffertérületnek mekkora memóriaterületet tarthat fenn a gép a másolásakor. Hiányossága, hogy írásvédett cartridge-ot nem ismer fel, és bár nem ír rá, de nem is közli, hogy a másolás ilyen akadályokba ütközik.

A kijelölt fájlok mindvégig jól elkülönülnek, mivel az ICE munkaterületén e fájlok ikonjai inverzben látszanak mindaddig, amíg az ESC-re lépve egy SPACE-t ütve a kijelölést nem töröljük vagy a kijelölés célját (másolás, info stb.) el nem végezzük.

A munkalécen található még a RENAME, VIEW és az INFO. Ezek mindegyike ugyanúgy, mint a fájlonkénti másolás, előzőleg elvárja, hogy jelöljük ki azt a fájlt, amelyikre vonatkoztatjuk a műveletet. A RENAME a kijelölt fájlt

Régi név: név  
Új név: név  
párosításban kiteszi egy ablakba, és még egy kurzort is ad hozzá, hogy azonnal el tudjuk végezni a fájl kivánt átkeresztelését.

A VIEW is igényli az előre kijelölt fájlt, majd választási lehetőséget ad rá, hogy azt

hogy ezt egy megjelenő ablakban az O.K.-ra SPACE-szel választolva még egyszer jóváhagyjuk.

## Üveggyöngyök

Az ICE szolgáltatásai közé tartozik a képernyő bal felső sarkából bármikor közelebb hívható zsebszámológép (lásd a 2. képen), de ez olyan kezdetleges, hogy egyetlen kis változtatás is elég lett volna a mérírássá ahhoz, hogy használója ne mérgeződjön: nem kellett volna megírni. Még csak túlzás, hogy egy számológéppel versenyre kelhet egy fejben számoló csak azért, mert a számjegy beírását a szoftver szerzője nem a számjegy bebillentyűzésével, hanem „kurzortölténgésben szenvedve”, a kis nyílkap a számjegyre mozgathatva és egy SPACE leütéssel oldotta meg.

Az ICE másik nem túl erős, de nem is ennyire hibás szolgáltatása a naptár (lásd a 3. képen), amelyik azonban csak havonta lapozható, és semmilyen határidőnapló funkciója nincs, mint például az IBM PC-re írt Sidekickben.

## Igéreték

Milyen egyéb programok bővítik az ICE képességeit? Időközben született néhány, az ICE-ra — és természetesen a sikerére is — támaszkodó, azt jól kiegészítő program. Ezek ismertetése legyen egy később megjelenő, újabb cikk feladata. Most csak néhány szóban róluk. Itt van például az ICICLE, amely lehetővé teszi nemcsak a négy Psion felhasználói programnak, de Superbasic programoknak is a felhasználó által definiált ikonokkal való vezérlését.

Az ARTICE kurzorgombokkal is, de főleg egérrel vezérelhető grafikai program.

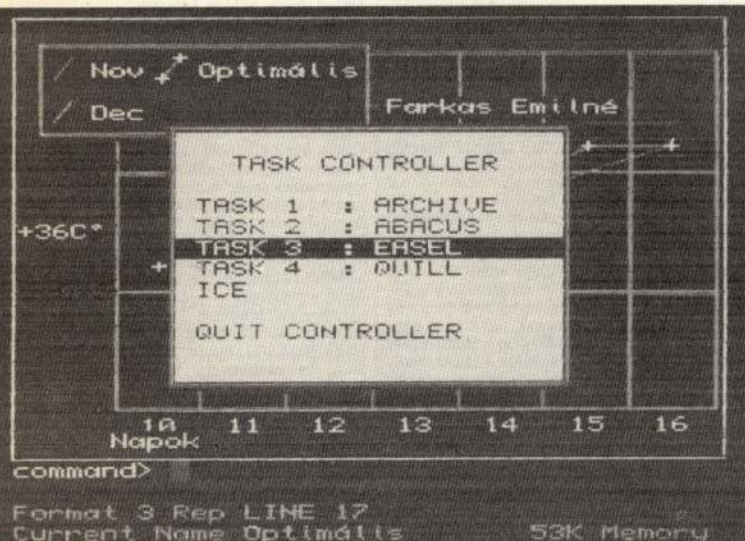
A ChoICE a felhasználó által kiválasztott és programcsomagba fűzött programok (max. 4 + ICE) kvázi multitaskban való futtatását teszi lehetővé (lásd a 4. képen).

Az ICE Toolkit (pop up, windows, menülc, alert, boxes) olyan felhasználói segédletek gyűjteménye, amelyek nemcsak a Tony Tebby-féle Toolkitból, de az ICE-ből is kimaradtak, és jól összeférnek az ICE-szal.

Mindezt valószínűleg egyetlen szoftverbe is bele lehetett volna süriteni, de sajnos a számítástechnikai szoftverek piacán gyakran a legerősebb hatású az az elv, hogy „amit ma megírhat, ne halaszd holnapra, de csak holnapután add el!” Így a felhasználó kétszer (n-szer) fizet.

(Az ICE iránt érdeklődők kielégíthetik kíváncsiságukat bármelyik szerzőn. 5 és 9 óra között a Csokonai Művelődési Ház QLub-jában (Bp. XV., Eötvös u. 64–66).

PACHER GÁBOR



4. kép

Egyes általunk kiválasztott fájl vagy fájl másolása ennél sokkal egyszerűbben is megy. Miután egy SPACE leütéssel kiválasztottuk a másolandó fájlokat, csak rá kell vezetnünk a kis nyílat a háttértár ikonjára, és ott egy újabb SPACE leütés után az ICE közli, hogy hány fájlt jelöltünk ki, aminek másolását egy O.K.-ra helyeslő SPACE leütés után hajlandó elkezdeni, mégpedig a kiválasztás sorrendjében! Mindkét másolási módszerrel menet közben jelzést kapunk, ha a kérdéses fájlal azonos nevű fájl már szerepel a cél háttértárolón. Ilyenkor — RE TRY-ra „bólintva” egy SPACE leütéssel — módunk van felülírást elrendelni vagy CANCEL-re azt kikerülni (lásd később).

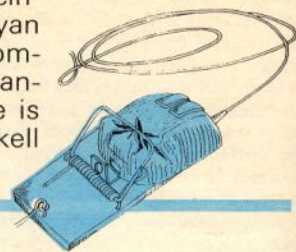
a képernyőn TO SCREEN vagy a soronyomatón TO PRINTER tekintsük meg.

Az INFO az előre kijelölt fájlról (több fájl esetén az utoljára kijelölről) elmondja a fájl nevét, típusát, hosszát és azt, hogy a QL órája szerint másodperc pontosan mikor került az adathordozóra.

Majd mindegyik ablakban van egy CANCEL opciója arra az esetre, ha menet közben meggondolnánk magunkat és mégsem akarnánk elvégezteni a géppel az előre kijelölt műveletet.

A BIN elnevezésű (Dustbin = szemétküldő) ikonra lépve és ott egy SPACE-t leütve tartalomjegyzéket adó adathordozóról — véletlen törlés lehetőségét kizárva — kitörölhetjük az összes előre kijelölt fájlt úgy,





Néhány évvel ezelőtt először akadt a kezem ügyébe Apple Macintosh számítógép. Az egérrel, menüvel, ablakkal való munka olyan szenzációs élmény volt nekem, öreg programozónak, hogy nyomban cikket írtam róla a Mikromagazinnak. Meglepetésemre ugyanabban a számban még két másik szerzőnek a Macintosh-cikke is megjelent. Nekik is olyan élmény volt, hogy úgy érezték: meg kell osztaniuk a közönséggel.

A visszhang kétféle volt. Akik csak a mi cikkeinkből értesültek az újdonságról, általában azt tartották, hogy túllelkendeztük. Új gépek, végül is, állandóan jelennek meg: miért éppen erről kellett ennyit írni? Ellenben akik maguk is kipróbálták már a Macintosht, azok azt kifogásolták, hogy túl hűvösen írunk róla: nem beszélünk „új korszakról”, nem emlegetjük a „programozás forradalmát”.

Azóta világosabb lett a helyzet, és most már tárgyilagosan tudjuk értékelni a jelenséget. Az elmúlt évben az USA-ban a Macintosh volt a legkelendőbb személyi számítógép (az eladott gépeknek kb. 10 százaléka). Az IBM PC, XT és AT együttesen 9 százalékot tett ki, a kompatibilis utánpótlással együtt azonban kb. 70 százalékot. Az Apple cég néhány évig veszteséges volt, és már azt latolgatta, hogy megállítja a Macintosh gyártását. Ehelyett annyira megnövekedett a kereslet, hogy Apple-ék új gyárat nyitottak az Írországhoz tartozó Cork szigetén. Ennek feladata az európai piac ellátása. Nehezebb dolguk lesz, mint a Cupertino-i részlegnek Kaliforniában vagy a szingapúrinak Délkelet-Ázsiában. Az Apple cég ugyanis büszke arra, hogy a Macintoshban egyetlen angol szó van csak beépítve: a gép neve (egyébektől ez sem angol, hanem skót). A corki üzemenk gondoskodnia kell arról, hogy minden európai ország a maga nyelvén értő gépet kapjon. Ezt megkönnyíti, hogy a szabályosan írt Macintosh programokban az emberi nyelvű szavak nincsenek szanaszét szórva, hanem egyetlen táblába gyűjtve. Így csak ezt kell egy darabban kicserélni. (Az Országos Terhivatalban ezt megcsinálják, gépekkel magyarul lehet szót érteni.)

Az IBM személyi számítógépeit többnyire MS—DOS operációs rendszerrel működtetik. („MS” a programtervező cégnek, a Microsoftnak a monogramja, „DOS” pedig lemezes operációs rendszert jelöl.)

Az MS—DOS skoban hasonló a CP/M operációs rendszerhez, csak hogy a CP/M nyolcbites processzorhoz, rendszerint Z80-hoz készült, utódja ellenben 16 bites processzort, többnyire 8086-ot vagy 8088-at használ.

Mind a CP/M-ben, mind az MS—DOS-ban meg kell tanulni egy csomó értelmen belülmagzalt. Angol szavak rövidítései ezek, de sok angol kollégával beszélgettem, aki tudta ugyan, hogy mit csinál a gépe, ha ezt írja be: IPL—II, de arról fogalma sem

volt, hogy ez miért jelenti azt, hogy „file transfer package” (állománykezelő programrendszer).

Az utasítás rövidítése után rendszerint egy vagy több állomány neve vagy egyéb paraméter következik. Ezeket magunk találjuk ki, csak hogy mégis van a dolgoknak bökkenője: sokszor fogjuk a neveket beírni. Ha a név tömör rövidítés, hamar elfelejtjük, hogy mit jelent. Ha viszont hosszú, akkor unalmas sokszor beíllentyűzni, és könnyen el is hibázhatjuk.

A CP/M és az MS—DOS még abból az időből származnak, amikor a számítógép terminálja papírra író szerkenyű volt, tulajdonképpen olyan írógép, amellyel a gép és a használója felváltva írt ugyanarra a papírra. A mai terminál többnyire képernyő, amelyen a kurzor nemcsak soronként egymás után, hanem tetszőlegesen lehet mozgani. A mozgatóra sokféle eszköz van, botkormány, fényceruza stb., de az utóbbi években legjobban az egér terjedt el. Ez egy kis doboz, amelyet az asztalon úgy guríthatunk ide-oda, mint a játékegert. Hosszú farka csatlakozik a számítógéphez, és ahogy az egér mozog az asztalon, úgy mozog a kurzor a képernyőn.

A képernyőn látjuk az utasításokat. Nem torz rövidítéseket, hanem teljes hosszúságú neveket. (Mint mondtam, nem nehéz őket egyszer s mindenkorra magyar nevekre kicserélni.) A kurzor ráállítjuk a kívántra, és megnyomjuk az egér hátán levő billentyűt. Egy billentyűnyomás elég akkor is, ha a szó 20-30 betűből áll. Csak akkor kell a nevet teljes hosszában beíllentyűzni, ha a gép még nem ismeri, vagyis éppen most találjuk ki. A továbbiakban már mindig elég rámutatni és egyet billentyűzni.

Az egészben az a legszebb, hogy minden program egyformán kezelhető. Az egérjátékot akármilyen felhasználó fél óra alatt, és aztán már minden programhoz használhatja. Szintaktikus hibát nem ejt: a képernyőn csak az jelenik meg, ami szintaktikusan helyes, mást nem választhat.

A Microsoft cég érdeme, hogy az első között ismerte fel az új programozási mód előnyeit, és számos kitűnő programot készített gépei számára. Bizonyos, hogy ezzel sokat tett azért, hogy az Apple cég mutatója a veszteség oldalról a nyereség oldalra lendüljön át.

Hát a nagy konkurens, az IBM mit csi-

nált? Egeret gyártott és megbízta a Microsoftot, hogy neki is készítsen olyan programokat, amelyek az Apple-ével vetélkednek. Az új IBM gépek 80286, 80386 és 80C86 processzorokkal dolgoznak. Ezek felülről kompatibilisek a korábbi 8086-tal és 8088-cal. A Microsoft ezekre a gépekre új operációs rendszert dolgozott ki, amelyet MS—OS2-nek hívnak. Ez már mindazt tudja, amit a Macintosh: egér, ablak, menü stb. A régi, MS—DOS-ra írt felhasználói programok azonban az új operációs rendszerrel nem futtathatók. Az átmenet megkönnyítésére tehát írtak Microsofték egy Windows (=ablakok) nevű programot. A Windows az MS—DOS alatt fut, és lehetővé teszi, hogy MS—DOS-ra írt felhasználói programokat egeres-ablakos-menus módszerrel lehessen kezelni. Ez azt a nehézséget is áthidalja, hogy még sok 8086 vagy 8088 processzoros gép van forgalomban, és ezeken az MS—OS2 operációs rendszer nem futtatható.

Megjegyzendő még, hogy az egéren kívül szükség van a Hercules vagy az EGA grafikus kártyára is a Windows vagy az MS—OS2 futtatásához. A Macintoshban az ennek megfelelő áramkörök be vannak építve, nem kell külön megvenni.

Persze a Macintosh sem pihent meg háberáin. Legújabb változataiban a főtár 4 M-ig (!) kiterjeszhető. Processzora a 68000 után a 68020, majd a 68030 lett. Ezeknek az utasításkészlete felülről kompatibilis a korábbiéval, tehát annak programjai az új gépen is futnak. Ennek ütemideje 20 MHz. Számos trükk gyorsítja meg tovább a működését. A 68020 256 bajtot tárol a processzor gyors regisztereiben, ha tehát ezekhez kell hozzáférni, akkor nem szükséges a tárból olvasni az utasítást. A 68030 még tovább megy: a 256 utasításbajton kívül 256 adatbajtot is tárol, tehát még ritkábban kell a tárhoz fordulni.

Említést érdemel még, hogy a két nagy versenytáron, az IBM-en és az Apple-en kívül az Atari és a Commodore Amiga is egerezik és ablakozik. Mivel ez a programozási módszer sokkal kényelmesebb és annyiál könnyebben megtanulható a korábbiaknál, hogy nem is sportszerű az összehasonlításuk, a következő években általános elterjedésére számíthatunk.



Most induló sorozatunkban azokat az új hardver- és szoftvertermékeket ismergetjük, amelyek várhatóan általánosan elterjednek és meghatározó szerepük lesz a fejlődés irányának kialakításában.

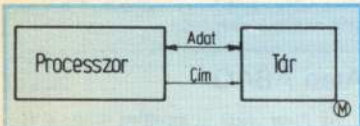
## Merre tart a világ?

# Új típusú processzorok

### Párhuzamos feldolgozás

Az egyik korai számítógép, az ENIAC elkészülte óta a gépek hasonlítanak abban, hogy különálló központi egységük és táruk van. Egy utasítás végrehajtásához a megfelelő adatot ki kell keresni a tárból, be kell olvasni a központi egységbe, ott el kell végezni a szükséges műveletet, majd az eredményt vissza kell juttatni a tárhoz (1. ábra).

Ezt az elrendezést sorosnak hívják, mert a feldolgozás egyes lépései egymást követik. A soros kialakításnak műszaki okai vannak. A félvezető táruk előtti korszakban a központi egységek és a táruk más-más anyagból készültek. A tár gyártási költsége



1. ábra

volt az alacsonyabb, így a központi egység minél jobb kihasználására törekedtek. Ezt a soros elrendezést valósította meg. Jelenleg azonban már mindkét egység szilíciumlapkán található. A felhasznált felület döntő többségét a tár foglalja el. A processzor szinte „folyton” dolgozik, míg a tár jó része szinte sohasem. A számítógép árát elsősorban a tár ára szabja meg, így az optimális megoldás jelentős árcsökkenést tételezne.

A megoldás egyik módja sok kisméretű, egyidejűleg dolgozó processzor működtetése, amelyek mindegyike saját kis tárral rendelkezik (2. ábra). (Az ábrán V=vezérlő, P=processzor, T=helyi tár, K=kommunikációs egység.)

Miért most vált ez fontossá? A probléma természetesen nem ma jelentkezett először, de a mesterséges intelligencia (artificial intelligence, AI) rendszerekben a feldolgozandó fogalmak számánál sokkal gyorsabban nő a köztük levő — és megvizsgálandó — kapcsolatok száma. (Emlékeztetjük az olvasót tavalyi sorozatunkra: Mesterséges értelem címmel öt részben közzöltük. — A szerk.) A hagyományos soros számítógépeket egymás után dolgozta fel, ezért már aránylag kis elemszám esetén is túl lassúnak bizonyul. A párhuzamos feldolgozás (parallel processing) és az ezt megvalósító integrált áramkörök a párhuzamos processzorok (parallel processor) a „nagy ugrás” egyik eszközeit jelentik.

### Csökkentett utasításkészlet

A másik, talán hasonló fontosságú változás a csökkentett utasításkészletű számítógép (reduced instruction set computer, RISC) megjelenése. A hagyományos számítógépek ún. komplex utasításkészletű számítógépek (complex instruction set computer, CISC). Ezeknél a tervezők arra törekedtek, hogy az utasítások számának, teljesítményének növelésével fokozzák az általában egyprocesszorú számítógépek teljesítményét. (Az utasítások száma többszáz.) A RISC-gépeknél az irányzat teljesen más: gyorsan végrehajtható, kisszámú utasításból álló készlettel felépített processzor, amiből esetleg több is van egy számítógépben. Az utasítások száma itt csak több tíz.

A CISC-gépekben egy utasítás ún. mikroutasítások sorozata, amelyeket a processzor egymás után, tehát sorosan dolgoz fel.

portja egy munkája során gyorsabb processzort igényelt, és a gyorsítás céljából elhagyta a ritkán előforduló utasításokat. Így az utasítások értelmezése gyorsabban ment, hiszen kevesebb lett a lehetséges esetek száma, és ezzel gyorsabb lett a processzor is. Az igazi RISC-gépek azonban csak a 80-as évtized közepén jelentek meg.

Melyek a jellemzői egy RISC-gépnek?

1. Minden utasítás végrehajtható egy óraciklus alatt. Így az utasítások nagyon egyszerűek, és/vagy a processzornak van kiegészítő hardverje, amely segít az összetettebb utasítások végrehajtásában. Hiányoznak tehát az esetenként akár tízciklusnyi végrehajtási idejű utasítások.

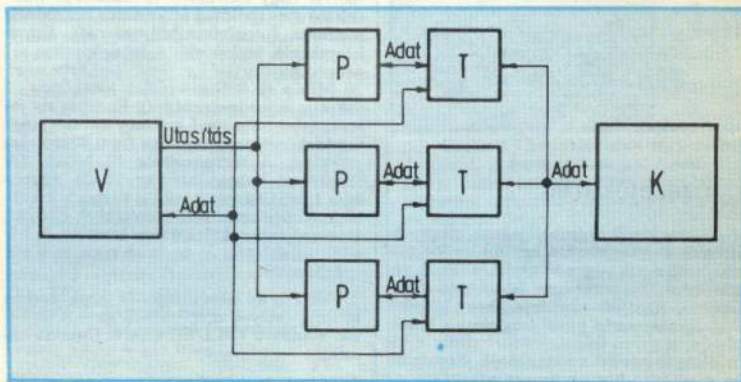
2. Töltésd/átíró el forma. Csak a betöltés és eltávolítás utasítás esetén dolgozik a processzor külső adatokkal. Hiányoznak tehát a töltésd és csinálj valamit — például adj hozzá, hasonlítsd össze — típusú utasítások.

3. Bedrótolt vezérlés. Nincs különbség az utasítások végrehajtási idejében, ezért nincsenek bonyolult, összetett utasítások. Az összetett utasítások időben egymás után végrehajtása helyett egyes utasításokat hardverben megvalósított és párhuzamosan végrehajtott utasításként dolgoz fel.

4. Viszonylag kevés utasítás és címzési mód. Emiatt a tervezés egyszerű.

5. Rögzített utasításforma.

6. Hosszú programkészítési idő. A csekély számú és egyszerű utasítás miatt a magas szintű nyelvek fordítóprogramja sok —



2. ábra

A feldolgozáshoz először is veszi a következő utasítást, tárolja, felbontja mikroutasításokra, azokat is tárolja, majd egymás után végrehajtja őket, tárolva közben a használt adatokat, eredményeket. Az adatokat esetleg a processzoron kívül levő tárból veszi, és az eredményeket oda teszi. Így a processzor viszonylag bonyolult felépítésű, bonyolult kapcsolatrendszerű egység, amelyen belül egy utasítás feldolgozása több óraciklust igényel.

A RISC rendszerű processzorok alapfogalomszerűsége a mikrogépek megjelenésével. 1975-ben az IBM egyik kutatócso-

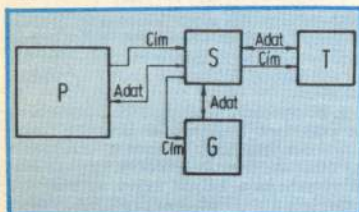
a CISC-rendszereknél jóval több — utasításból állnak, és megírásuk is nehezebb, mint a CISC-gépeknél.

A RISC-processzorokat különböző kiegészítővel látják el. Így például nagyszámú — gyakran több mint 100 — regiszter, lefolytatók, lebegőpontos hardver, szorzók, folyamatos közötti kommunikációs hardver tartozik a processzorhoz. A RISC-gépek programozása azért sokkal munkaigényesebb és bonyolultabb, mint a CISC-gépeké, mert az előbbieket utasításainak egész sorozata felel meg az utóbbiak egy-egy utasításának.

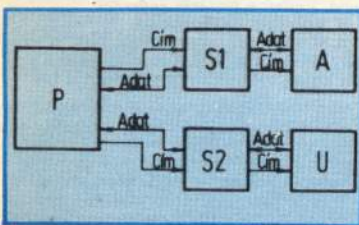


## A Neumann-féle szűk keresztmetszet feloldása

A mikroprocesszor adatfeldolgozási sebessége nagyobb, mint az adatátvitel sebessége a processzor és a tár között. Ezt nevezik Neumann-féle szűk keresztmetszetnek. A probléma hagyományos megoldásmódja: gyorstár hozzáépítése a processzorhoz (3. ábra). Az új jelölés: S=társzervező, G=gyorstár). Az újabb változat az ún. Harvard-architektúra, melynél külön-külön gyors utasítás- és adattár van (4. ábra. U=utasítástár, A=adattár). Ez gyakorlatilag kétszeres buszszélességet jelent. Így a processzornak nem kell annyit várnia az időigényes adatátvitelre, mert egyszerre hívható be az utasítás és az általa használt adatok a tárból. A programozás ugyanezért itt is nehezebb, hiszen a programozónak erre figyelni kell.



3. ábra



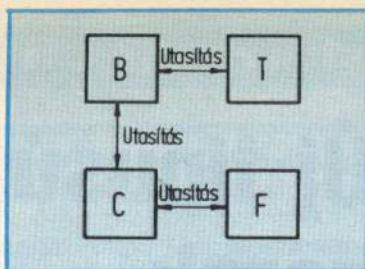
4. ábra

## Csővezetékvezés

Az utasítások végrehajtásának meggyorsítására az utasításokat egy ún. csőbe töltik: a cső egyik végén töltik be, a másik végén veszi elő a processzor feldolgozó része. Ez a módszer lehetővé teszi, hogy lassabb feldolgozás esetén a cső megteléséig feltöltődjen, a gyors feldolgozásnál pedig a kiürülésig fogyjanak az utasítások. Ezenfelül, ha a cső elég hosszú, az ugróutasítások címeit is itt találhatja meg a processzor. Ilyen ugrás végrehajtási ideje igen rövid. Egy ilyen egység látható az 5. ábrán (ahol B=betöltő, C=csővezeték, F=feldolgozó egység).

## Kombinációk

Az eddigiekben ismertetett megoldások az új processzorokban először külön-külön jelentek meg, de a legújabbakban már a kombinációkat használják. Ezzel igyekeznek az előnyöket optimalizálni és a hátrányokat csökkenteni. Most áttekintjük a már iparilag megvalósított processzorokat,



5. ábra

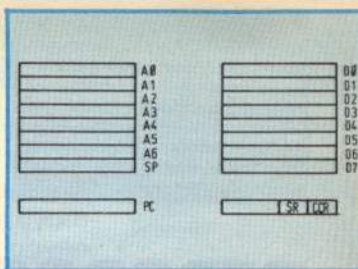
vagyis amelyeknek a sorozatgyártását elkezdték. Ezek mindegyike több-kevesebb elem kombinációja.

## SAIC Delta Floating Point Processor™

A Science Applications International Corporation (SAIC) évi 600 millió dollár forgalmú, a korszerű technológiák kutatásával, bevezetésével foglalkozó cég. Szerteágazó tevékenységére jellemző, hogy a számítástechnikai kutatásoktól a vitorlás hajó tervezésig terjed a kutatási skálája. (Ez utóbbi eredménye volt a hosszú távú vitorlázás világbajnokságának számító Amerika Kupán győztes Stars and Stripes hajó tervezése.)

A Delta a cég Artificial Neural Systems (ANS) technológiája szerint készült. Ez az emberi idegrendszer és az agytevékenység bizonyos részeinek tanulási és információfeldolgozó működését utánozza. A processzornal nagy hálózatok szimulációja valósítható meg (például folyamatos beszédnél, autonóm forgalomszabályozásnál). Ezt a következők segítik elő. A lebegőpontos segédprocesszor (lásd a fényképet) sebessége 20 Mflop (a Mflop=millió lebegőpontos művelet másodpercenként). Ez tehát azt jelenti, hogy sebessége mintegy két nagyságrenddel nagyobb, mint egy Intel 80386 alapú gépé. A tárcapacitása 12 Mbájt. (A fényképen látható kártyán 12 db, egyenként 1 Mbájt kapacitású kis kártya a RAM-tár.) A processzor csővezetékekkel ellátott, Harvard-architektúrájú (egy utasítás- és két adatgyorsító tárral) és 1024 regiszterű (!). Jellemzői:

- 32 és 64 bites integer és lebegőpontos adatok, melyek megfelelnek az ANSI/IEEE Standard 754 DEC F és G formátumának;



6. ábra

- több mint 1800 utasítás (beleértve 96 feltételes, formátumátalakító, mátrix- és SQRT utasítás);

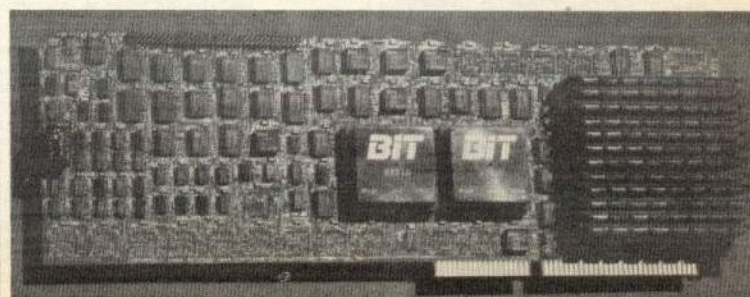
- az utasítások ciklusideje 100 ns (a legtöbb utasítás végrehajtódik egy cikluson belül, beleértve a lebegőpontos szorzást is);
- 32 bit széles a perifériabusz: 32 külső készülékkel tarthat kapcsolatot 40 Mbájt/s sebességgel, és rendelkezik PC/AT busszal is. Mindezek a kártyán levő csatlakozókon át, egyetlen PC/AT kártyán (a megfelelő hűtés érdekében az egyik szomszédos csatlakozót azonban nem használhatjuk);
- gazdag a szoftver-ellátottsága is.

A COMDEX/FALL 87-en, ahol a terméket bevezették, George Works, a SAIC AND Division igazgatója nyilvánosságra hozta, hogy kidolgozta egy speciális kiterjesztő egységet, ami 5 Delta-kártyát használhat, és amelynek a *számítási teljesítménye megegyezik a Cray-1 számítógépével.* (Ez a világ egyik leggyorsabb, tudományos célú nagyszámítógépe.)

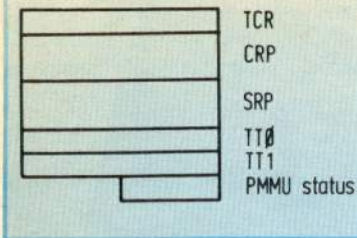
## Atari ABAQ

Az Atari céget — egyetlen itthon is forgalmazott gépe alapján — az olvasók ismerik. Valószínűleg kevésbé ismert azonban, hogy a cég újabban „felfelé lépked”. Korszerű termékei 16 és 32 bites mikroprocesszorú gépek, amelyekről e sorozat későbbi részében lesz szó. Ezekhez készült az ABAQ (az abacus, az ősi számolóeszköz nevének szógyöke).

A cég elnöke, Sam Tramiel (a cégtulajdonos és korábban a Commodore-elnök, Tramiel fia) a COMDEX-en fénykép minőségű grafikát produkáló, különlegesen nagy teljesítményű számfeldolgozóként jellemezte ezt az eszközt, amely egy szuperszámítógép teljesítményét adja egy mikroprocesszor. Úgy vélte, hogy reális áron 100 MIPS (millió utasítás másodpercenként) is elérhető lesz ezzel az eszközzel.

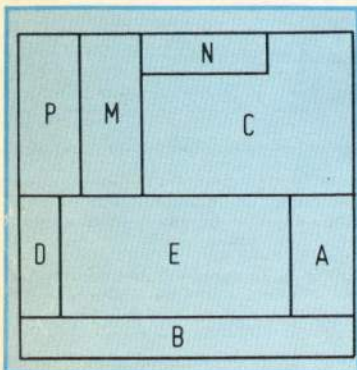






7. ábra

A processor RISC rendszerű, párhuzamos processzorokból épül fel. Beépített nagy sebességű soros csatornája lehetővé



8. ábra

teszi, hogy sok processzoros tömbként (multi-processor array) vagy helyi hálózatként (local area network, LAN) használják. A rendszer az Inmos T-800 típusú, 32 bites, 10-12 MIPS sebességű mikroprocesszorra épül.

- Jellemzői:
- a számítások sebessége 1,5 Mflop;
  - 4 bájt RAM-ot tartalmaz és további 1 Mbajt RAM a kijelzőhöz kártyánként;
  - a rendszer felfűzhető 3 kártyáig (kártyánként 4 processzorral);
  - a képernyő 1280 × 960 pontos, 16 színű vagy kétszínű képponttal vagy 1024 × 768 pontos, 256 színű képponttal vagy 640 × 480 pontos, 256 színű kétképernyős képponttal vagy 512 × 480 pontos, 16 millió színű (!) képponttal rendelkezik.
- Az eszközökhöz Unix-szerű operációs rendszert készítettek.

## A Lowell egyetem Parallell Data Flow Coprocessor terméke

A Massachusetts kormányzója által alapított egyetem Produktívítás-kiterjesztő Központja egyrészt a különböző tesztszkeke együttesének kutató/tanító munkáját hangolja össze, másrészt együttműködik a különböző ipari intézményekkel. A központ a Commodore cég Amiga gépeihez fejlesztette ki a címben szereplő párhuzamos adatáramlású koprocesszort. Képi folyamatok

előállítására, folyamatok beszédszintézisére, grafika generálására/módosítására, digitális folyamatokban, képkompresszióval, elektronikus nyomtatás teljesítménynövelésénél, numerikus folyamatok teljesítménynövelésénél, tömbfolyamatoknál, mátrix/vektorműveleteknél, adatitkosításnál és elektronikus áramkörök szimulálásánál kívánják használni.

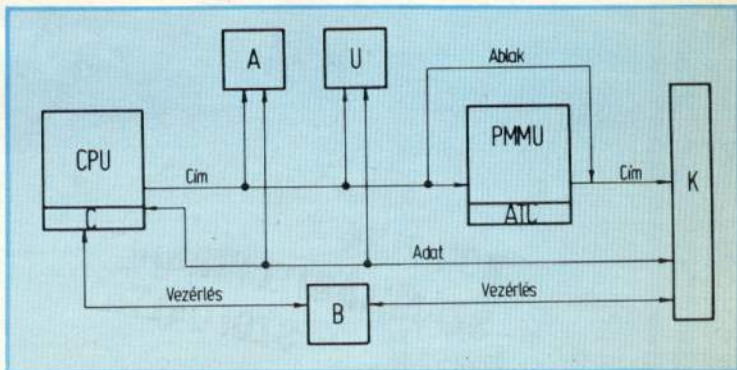
A COMDEX-en a fentiekben kívül csak annyit közöltek, hogy a NEC által készített, hét integrált áramkörből álló eszköz 30 MIPS sebességet is felülmúl.

## Motorola MC68030

A COMDEX-en jelentette be a Motorola cég is a régóta ígért és specifikációjában ismert 32 bites processzora árusítását. Mivel erről a processzorról már szó volt a CW-

kailag kettőszótható egy szoftverfordító és egy végrehajtó részre. Ez a 8. ábrán látható, ahol az alsó fél egy RISC-gépnek megfelelő felépítésű. (Az ábrán N=nano ROM, M=mikro ROM, P=PMMU és ATC=address transfer control, címátadás-vezérlés — C=control és PLA, D=DTAGS, E=EU, A=adatgyorsító, B=bus control.)

Az egyenként 64 duplaszavas gyorsítók közül az adat irható is, az utasítás csak olvasható. Az adatgyorsító regiszterként is használható. A két gyorsító önálló működése és a mindkettőhöz egyidejű hozzáférés lehetősége egy módosított Harvard-architektúrát eredményezett (9. ábra, melyen C=utasítás csövezeték, K=külső busz). Itt három belső busz van, amelyek mindegyike a többitől függetlenül működtethető. A csövezeték áramláspontú, és kapcsolható a külső busszal háromféle ciklustípussal va-



9. ábra

Számítástechnika 1987. augusztus 26-i számának 20. oldalán, csak az ott nem tárgyalatról, valamint a jelen cikkkel kapcsolatos dolgokról írok.

A 6. ábrán a felhasználói programozói modell látható. Van tehát 7 címregiszter, egy-egy veremmutató (stack pointer, SP), programszámláló (program counter, PC), 8 adatregiszter (ezek duplaszavasak=32 bit), egy egybájtos állapotregiszter (status register, SR), állapot-kódregiszter (control code register, CCR). A 7. ábra a felügyelői modell regisztereit mutatja be. Ezek az átviteli irányító regiszter (translation control register, TCL), a négyeszeres szavas CPU gyökérmutató (CPU root pointer, CRP) és a felügyelői gyökérmutató (supervisor root pointer, SRP), a két transzparens átviteli regiszter (transparent translation, TT0 és TT1), amelyek ismét duplaszavasak, valamint az egyszerűs lapptárgyelő állapot (paging memory management unit status PMMU.)

Bár az utasítások két óraciklus idejűek, a belső és külső busz aktivitásának és a belső processzor-részegységek működésének időbeni átfedése miatt az átlagban egy utasításra fordított idő körülbelül egy ciklus. Az utasítás tényleges végrehajtása úgy történik, hogy a végrehajtó egységben (execution unit, EU) az első ciklusban egy hívás generálódik a mikrookodolás első szintjére. Ez egy ún. nanokódot vált ki, és az hatodik végre a második szinten. A mikrookodok az ellenőrző hardver módosítása nélkül változtathatók, mivel az integrált áramkör fizi-

lósítja meg. Az egyik az összes korábbi Motorola processzornál is használt aszinkron buszciklus, a második egy két óraciklusnyi szinkron buszciklus, ami jól használható az újfajta dinamikus RAM-okhoz: az ún. statikus lap és félbájt és oszlop üzemmódúakhoz. Ezeknél 5 óraciklus alatt átvihető négy duplaszó. A harmadik az ún. csomó típus, ami a változó igényű periféria kezeléshez ad szabadon választható hosszúságú ciklusmegadási lehetőséget.

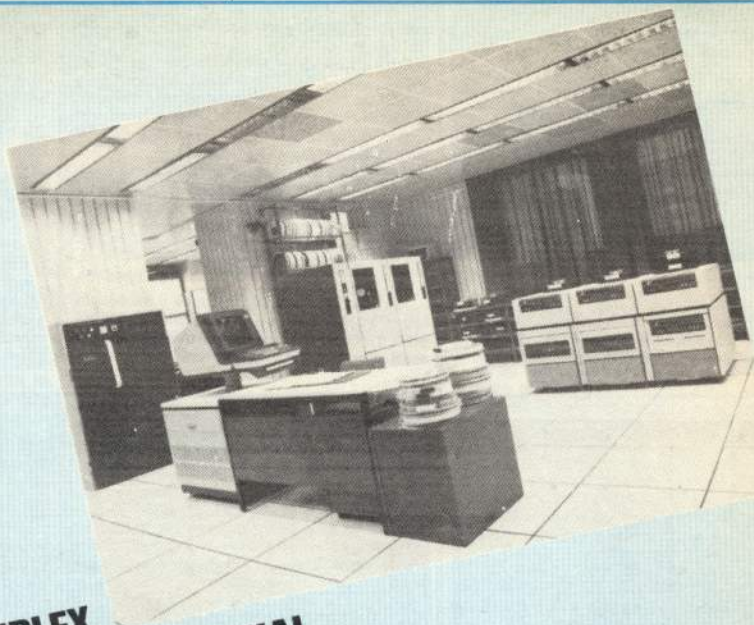
A RISC-gép koncepció megvalósításához a processzorban van léptető (dupla szavak léptetésére, forgatására egyetlen óraciklus alatt), lapptárgyelő (ami a készenléti címáthelyezésre szolgál a processzor egységei között, elvégezve ezzel az adatforgalom irányítását). A RISC-gép tulajdonságok miatt az órafrekvencia elérheti a 25 MHz értéket 5 MIPS teljesítmény mellett. Dr. S. E.

A HIRADÁSTECHNIKAI TUDOMÁNYOS EGYESÜLET  
Mikroszámítógépes programnyelvek  
és Operációsrendszerek Szakosztálya  
gyakorlati bemutatóval egybekötött  
előadást szervez az alábbi témákból:

- I. IBM PC és ZX-Spectrum illesztése az RS 232 soros vonalon keresztül
- II. ZX-Spectrum és VC 1541 Commodore meghajtó (printerek) stb. illesztése

Előadó: Moharos Ince  
Helye: MTE SZ Székház, Budapest V. ker.  
Kossuth tér 6-8. 111/333.  
Ideje: 1988. Április 5. 14 óra





## KOMPLEX SZÁMÍTÁSTECHNIKAI SZOLGÁLTATÁS!



### TEK V. számítástechnikai főosztály

Budapest VII., Vörösmarty u. 16.  
Tel.: 427-338, 202-415

A Ferroglobus számítástechnikai főosztálya nagy adatfeldolgozási hagyományokkal, jól képzett, nagy tapasztalatú munkatársakkal rendelkező számítóközpontja széles skálájú, komplex számítástechnikai szolgáltatásait ajánlja:

- kereskedelmi szakmai, készletgazdálkodási, pénzügyi, számviteli, bér- és munkaügyi stb. rendszerek szervezése, programozása és fejlesztése nagy kapacitású, modern ICL típusú számítógépen
- mágneses adathordozóra történő adat-előkészítés — rögzítés
- számítógépidő bérbeadása
- tanácsadás

**Ferroglobus**



# Z80-as utasítástáblázat

Lapunk 1985/4-es számának mellékleteként közöltük a Z80-as mikroprocesszor utasításainak kódtáblázatát. Ezt követően olvashatták 7 részes sorozatunkat „Z80 programozási gyakorlatok” címen. Ezt a témakört most a Z80-as utasítástáblázat közreadásával tesszük teljessé.

TAKÁCS FERENC

Utasítás- alak	Funkció	Idő- igény	Felt.reg. SZHPNC	Hexa.kód 1. 2. 3. 4.
<u>8 bites adatmozgatás</u>				
LD r,s	r=s	4	.....	<u>40</u>
LD rx,xs	rx=xs	8	.....	<u>DD 40</u>
LD xr,sx	xr=sx	8	.....	<u>DD 40</u>
LD r,(HL)	r=(HL)	7	.....	<u>46</u>
LD (HL),s	(HL)=s	7	.....	<u>70</u>
LD r,(x+d)	r=(x+d)	19	.....	<u>DD 46</u> d
LD (x+d),s	(x+d)=s	19	.....	<u>DD 70</u> d
LD r,n	r=n	7	.....	<u>06</u> n
LD xr,n	xr=n	11	.....	<u>DD 06</u> n
LD (HL),n	(HL)=n	10	.....	36 n
LD (x+d),n	(x+d)=n	19	.....	<u>DD 36</u> d n
LD A,(BC)	A=(BC)	7	.....	<u>0A</u>
LD A,(DE)	A=(DE)	7	.....	1A
LD (BC),A	(BC)=A	7	.....	<u>02</u>
LD (DE),A	(DE)=A	7	.....	12
LD A,(hl)	A=(hl)	13	.....	3A 1 h
LD (hl),A	(hl)=A	13	.....	32 1 h
LD A,I	A=I	9	++0+0.	ED 57
LD A,R	A=R	9	++0+0.	ED 5F
LD I,A	I=A	9	.....	ED 47
LD R,A	R=A	9	.....	ED 4F
<u>16 bites adatmozgatás</u>				
LD dd,hl	dd=hl	10	.....	<u>01</u> 1 h
LD x,hl	x=hl	14	.....	<u>DD 21</u> 1 h
LD HL,(hl)	H=(hl+1),L=(hl)	16	.....	2A 1 h
LD ss,(hl)	ssH=(hl+1),ssL=(hl)	20	.....	ED <u>4B</u> 1 h
LD x,(hl)	xH=(hl+1),xL=(hl)	20	.....	<u>DD 2A</u> 1 h
LD (hl),HL	(hl+1)=H,(hl)=L	16	.....	22 1 h
LD (hl),ss	(hl+1)=ssH,(hl)=ssL	20	.....	ED <u>43</u> 1 h
LD (hl),x	(hl+1)=xH,(hl)=xL	20	.....	<u>DD 22</u> 1 h
LD SP,HL	SP=HL	6	.....	F9
LD SP,x	SP=x	10	.....	<u>DD</u> F9
PUSH qq	(SP-1)=qqH,(SP-2)=qqL,SP=SP-2	11	.....	<u>C5</u>
PUSH x	(SP-1)=xH,(SP-2)=xL,SP=SP-2	15	.....	<u>DD</u> E5
POP qq	SP=SP-2,qqL=(SP-2),qqH=(SP-1)	10	.....	<u>C1</u>
POP x	SP=SP-2,xL=(SP-2),xH=(SP-1)	14	.....	<u>DD</u> E1
<u>Csere, keresés és tömbátvitel</u>				
EX DE,HL	DE <-> HL	4	.....	EB
EX AF,AF'	AF <-> AF'	4	.....	0B
EXX	BC <-> BC', DE <-> DE', HL <-> HL'	4	.....	D9
EX (SP),HL	H <-> (SP+1), L <-> (SP)	19	.....	E3
EX (SP),x	xH <-> (SP+1), xL <-> (SP)	23	.....	<u>DD</u> E3
LDI	(DE)=(HL),DE=DE+1,HL=HL+1,BC=BC-1	16	.0+0.	ED A0
LDIR	(DE)=(HL),DE=DE+1,HL=HL+1,BC=BC-1 Ha BC<>0, akkor PC=PC-2	21<>.000.	ED	B0 16=
LDD	(DE)=(HL),DE=DE-1,HL=HL-1,BC=BC-1	16	.0+0.	ED AB
LDDR	(DE)=(HL),DE=DE-1,HL=HL-1,BC=BC-1 Ha BC<>0, akkor PC=PC-2	21<>.000.	ED	B8 16=



Utasítás- alak	Funkció	Idő- igény	Felt.reg. SZHPNC	Hexa.kód 1. 2. 3. 4.
-------------------	---------	---------------	---------------------	-------------------------

CPI	Ha A=(HL), akkor #Z=1,HL=HL+1,BC=BC-1 Ha BC=0, akkor #P=0	16	+++1.	ED A1
CPIR	Ha A=(HL), akkor #Z=1,HL=HL+1,BC=BC-1 Ha BC=0, akkor #P=0	21 16	+++1.	ED B1
CPD	Ha #Z=0 es #P=1, akkor PC=PC-2 Ha A=(HL), akkor #Z=1,HL=HL-1,BC=BC-1 Ha BC=0, akkor #P=0	16	+++1.	ED A9
CPDR	Ha A=(HL), akkor #Z=1,HL=HL-1,BC=BC-1 Ha BC=0, akkor #P=0 Ha #Z=0 es #P=1, akkor PC=PC-2	21 16	+++1.	ED B9

B bites aritmetikai és logikai utasítások

ADD A,s	A=A+s	4	+++0+	<u>00</u>
ADD A,xs	A=A+xs	8	+++0+	<u>DD 00</u>
ADD A,n	A=A+n	7	+++0+	C6 n
ADD A,(HL)	A=A+(HL)	7	+++0+	B6
ADD A,(x+d)	A=A+(x+d)	19	+++0+	<u>DD 06</u> d
ADC A,s	A=A+s+#C	4	+++0+	<u>08</u>
ADC A,xs	A=A+xs+#C	8	+++0+	<u>DD 08</u>
ADC A,n	A=A+n+#C	7	+++0+	CE n
ADC A,(HL)	A=A+(HL)+#C	7	+++0+	8E
ADC A,(x+d)	A=A+(x+d)+#C	19	+++0+	<u>DD 8E</u> d
SUB s	A=A-s	4	+++1+	<u>90</u>
SUB xs	A=A-xs	8	+++1+	<u>DD 90</u>
SUB n	A=A-n	7	+++1+	D6 n
SUB (HL)	A=A-(HL)	7	+++1+	96
SUB (x+d)	A=A-(x+d)	19	+++1+	<u>DD 96</u> d
SBC A,s	A=A-s-#C	4	+++1+	<u>98</u>
SBC A,xs	A=A-xs-#C	8	+++1+	<u>DD 98</u>
SBC A,n	A=A-n-#C	7	+++1+	DE n
SBC A,(HL)	A=A-(HL)-#C	7	+++1+	9E
SBC A,(x+d)	A=A-(x+d)-#C	19	+++1+	<u>DD 9E</u> d
AND s	A=A AND s	4	++1+00	<u>A0</u>
AND xs	A=A AND xs	8	++1+00	<u>DD A0</u>
AND n	A=A AND n	7	++1+00	E6 n
AND (HL)	A=A AND (HL)	7	++1+00	A6
AND (x+d)	A=A AND (x+d)	19	++1+00	<u>DD A6</u> d
XOR s	A=A XOR s	4	+0+00	<u>AB</u>
XOR xs	A=A XOR xs	8	+0+00	<u>DD AB</u>
XOR n	A=A XOR n	7	+0+00	EE n
XOR (HL)	A=A XOR (HL)	7	+0+00	AE
XOR (x+d)	A=A XOR (x+d)	19	+0+00	<u>DD AE</u> d
OR s	A=A OR s	4	+0+00	<u>B0</u>
OR xs	A=A OR xs	8	+0+00	<u>DD B0</u>
OR n	A=A OR n	7	+0+00	F6 n
OR (HL)	A=A OR (HL)	7	+0+00	B6
OR (x+d)	A=A OR (x+d)	19	+0+00	<u>DD B6</u> d
CP s	A-s	4	+++1+	<u>BB</u>
CP xs	A-xs	8	+++1+	<u>DD BB</u>
CP n	A-n	7	+++1+	FE n
CP (HL)	A-(HL)	7	+++1+	BE
CP (x+d)	A-(x+d)	19	+++1+	<u>DD BE</u> d
INC r	r=r+1	4	+++0.	<u>04</u>
INC xr	xr=xr+1	8	+++0.	<u>DD 04</u>
INC (HL)	(HL)=(HL)+1	11	+++0.	34
INC (x+d)	(x+d)=(x+d)+1	23	+++0.	<u>DD 34</u> d
DEC r	r=r-1	4	+++1.	<u>05</u>
DEC xr	xr=xr-1	8	+++1.	<u>DD 05</u>
DEC (HL)	(HL)=(HL)-1	11	+++1.	35
DEC (x+d)	(x+d)=(x+d)-1	23	+++1.	<u>DD 35</u> d



16 bites aritmetikai utastások

ADD HL,dd	HL=HL+dd	11	..!.0+	<u>09</u>			
ADC HL,dd	HL=HL+dd+#C	15	++!+0+	ED	<u>4A</u>		
SBC HL,dd	HL=HL-dd-#C	15	++!+1+	ED	<u>42</u>		
ADD x,xd	x=x+xd	15	..!.0+	<u>DD</u>	<u>09</u>		
INC dd	dd=dd+1	6	.....	<u>03</u>			
INC x	x=x+1	10	.....	<u>DD</u>	<u>23</u>		
DEC dd	dd=dd-1	6	.....	<u>0B</u>			
DEC x	x=x-1	10	.....	<u>DD</u>	<u>2B</u>		

Általános aritmetikai és CPU vezérlő utastások

DAA	Kiigazítás BCD művelet után	4	+++++	27			
CPL	A= egyes komplemente A-nak	4	..1.1.	2F			
NEG	A=-A	8	++++1+	EB	<u>44</u>		
CCF	#C=1-#C	4	..!.0+	3F			
SCF	#C=1	4	..0.01	37			
NDP		4	.....	<u>00</u>			
HALT	CPU megállítás	4	.....	<u>76</u>			
DI	Megszakítás megtiltás	4	.....	<u>F3</u>			
EI	Megszakítás lehetővé tétel	4	.....	<u>FB</u>			
IM i	Megszakításmód beállítás	8	.....	EB	<u>46</u>		

Léptető utastások

RLCA	Ciklikus léptetés balra, #C a körben	4	..0.0+	<u>07</u>			
RRCA	Ciklikus léptetés jobbra, #C a körben	4	..0.0+	<u>0F</u>			
RLA	Ciklikus léptetés balra	4	..0.0+	<u>17</u>			
RRA	Ciklikus léptetés jobbra	4	..0.0+	<u>1F</u>			
RLC s	Ciklikus léptetés balra, #C is a körben	8	++0+0+	CB	<u>00</u>		
RLC (HL)	Ciklikus léptetés balra, #C is a körben	15	++0+0+	CB	<u>06</u>		
RLC (x+d)	Ciklikus léptetés balra, #C is a körben	23	++0+0+	<u>DD</u>	CB	d	<u>06</u>
RRC s	Ciklikus léptetés jobbra, #C is a körben	8	++0+0+	CB	<u>00</u>		
RRC (HL)	Ciklikus léptetés jobbra, #C is a körben	15	++0+0+	CB	<u>0E</u>		
RRC (x+d)	Ciklikus léptetés jobbra, #C is a körben	23	++0+0+	<u>DD</u>	CB	d	<u>0E</u>
RL s	Ciklikus léptetés balra	8	++0+0+	CB	<u>10</u>		
RL (HL)	Ciklikus léptetés balra	15	++0+0+	CB	<u>16</u>		
RL (x+d)	Ciklikus léptetés balra	23	++0+0+	<u>DD</u>	CB	d	<u>16</u>
RR s	Ciklikus léptetés jobbra	8	++0+0+	CB	<u>18</u>		
RR (HL)	Ciklikus léptetés jobbra	15	++0+0+	CB	<u>1E</u>		
RR (x+d)	Ciklikus léptetés jobbra	23	++0+0+	<u>DD</u>	CB	d	<u>1E</u>
SLA s	"Aritm." léptetés balra (s*2)	8	++0+0+	CB	<u>20</u>		
SLA (HL)	"Aritm." léptetés balra ((HL)*2)	15	++0+0+	CB	<u>26</u>		
SLA (x+d)	"Aritm." léptetés balra ((x+d)*2)	23	++0+0+	<u>DD</u>	CB	d	<u>26</u>
SRA s	"Aritm." léptetés jobbra (s/2)	8	++0+0+	CB	<u>28</u>		
SRA (HL)	"Aritm." léptetés jobbra ((HL)/2)	15	++0+0+	CB	<u>2E</u>		
SRA (x+d)	"Aritm." léptetés jobbra ((x+d)/2)	23	++0+0+	<u>DD</u>	CB	d	<u>2E</u>
SRL s	"Logikai" léptetés jobbra	8	++0+0+	CB	<u>38</u>		
SRL (HL)	"Logikai" léptetés jobbra	15	++0+0+	CB	<u>3E</u>		
SRL (x+d)	"Logikai" léptetés jobbra	23	++0+0+	<u>DD</u>	CB	d	<u>3E</u>
RLD	Ciklikus tetrád léptetés balra	18	++0+0.	ED	<u>6F</u>		
RRD	Ciklikus tetrád léptetés jobbra	18	++0+0.	ED	<u>67</u>		

Bitkezelő utastások

BIT b,s	Ha r.b=0, akkor #Z=1	8	!+!0.	CB	<u>40</u>		
BIT b,(HL)	Ha (HL).b=0, akkor #Z=1	12	!+!0.	CB	<u>46</u>		
BIT b,(x+d)	Ha (x+d).b=0, akkor #Z=1	20	!+!0.	<u>DD</u>	CB	d	<u>46</u>
RES b,s	r.b=0	8	.....	CB	<u>80</u>		
RES b,(HL)	(HL).b=0	15	.....	CB	<u>86</u>		
RES b,(x+d)	(x+d).b=0	23	.....	<u>DD</u>	CB	d	<u>86</u>
SET b,s	r.b=1	8	.....	CB	<u>C0</u>		
SET b,(HL)	(HL).b=1	15	.....	CB	<u>C6</u>		
SET b,(x+d)	(x+d).b=1	23	.....	<u>DD</u>	CB	d	<u>C6</u>



Utasítás- alak	Funkció	Idő- igény	Felt. reg.	Hexa.kód S7HFNC	1.	2.	3.	4.
-------------------	---------	---------------	---------------	--------------------	----	----	----	----

### Vezérlésátadási utasítások

JP hl	PC=hl	10	.....	C3	1			h
JP cc,hl	Ha cc teljesül, akkor PC=hl	10	.....	C2	1			h
JR d	PC=PC+d	12	.....	18				d
JR c,d	Ha c teljesül, akkor PC=PC+d	7,12	.....	20				d
JP (HL)	PC=HL	4	.....	E9				
JP (x)	PC=x	8	.....	DD				E9
DJNZ d	B=B-1 Ha B(>0), akkor PC=PC+d	8,13	.....	10				d
CALL hl	PUSH PC, PC=hl	17	.....	CD	1			h
CALL cc,hl	Ha cc teljesül, akkor PUSH PC,PC=hl	10,17	.....	C4	1			h
RET	PDP PC	10	.....	C9				
RET cc	Ha cc teljesül, akkor PDP PC	5,11	.....	C0				
RETI	Megszakítási szubrutin vége	14	.....	ED				40
RETN	Megszakítási szubrutin vége	14	.....	ED				45
RST p	PUSH PC, PC.H=0, PC.L=p	11	.....	C7				

### Kimásolási-bemásolási utasítások

IN A,(n)	A=IN(An) "bemásolás (An)-ről A-ba"	11	.....	DB				n
IN r,(C)	r=IN(BC)	12	+++0.	ED				40
IN F,(C)	IN(BC)	12	+++0.	ED				70
INI	(HL)=IN(BC), B=B-1, HL=HL+1	16	!+!+!	ED				A2
INIR	(HL)=IN(BC), B=B-1, HL=HL+1 Ha B(>0), akkor PC=PC-2	16=	21<>!	!+!+!				ED B2
IND	(HL)=IN(BC), B=B-1, HL=HL-1	16	!+!+!	ED				AA
INDR	(HL)=IN(BC), B=B-1, HL=HL-1 Ha B(>0), akkor PC=PC-2	16=	21<>!	!+!+!				ED BA
OUT (n),A	OUT(An)=A "Kimásolás A-ból (An)-re"	11	.....	D3				n
OUT (C),r	OUT(BC)=r	12	.....	ED				41
OUTI	OUT(BC)=(HL),B=B-1,HL=HL+1	16	!+!+!	ED				A3
OTIR	OUT(BC)=(HL),B=B-1,HL=HL+1, Ha B(>0), akkor PC=PC-2	16=	21<>!	!+!+!				ED B3
OUTD	OUT(BC)=(HL),B=B-1,HL=HL-1	16	!+!+!	ED				AB
OTDR	OUT(BC)=(HL),B=B-1,HL=HL-1, Ha B(>0), akkor PC=PC-2	16=	21<>!	!+!+!				ED BB

### Megjegyzések

Jelzőbit beállítások: 1=egyre állítás, 0=nullára állítás, +=eredménytől függő beállítás, !=nem definiált értékre állítás, .=a jelzőbit nem változik.

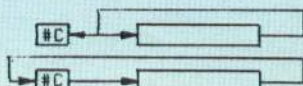
A táblázat funkció oszlopában a jelzőbitek # jelöli.

Az F regiszter jelzőbitjei (A jelzés 1-es állapotba állítással történik.)

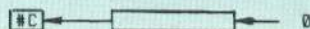
7.	S	negatív eredmény jelzése
6.	Z	nulla eredmény jelzése
5.	-	nem használt
4.	H	félátvitel jelzése
3.	-	nem használt
2.	P/V	túlcsordulás, páros paritású eredmény, IFF2 állapotának jelzése
1.	N	kivonás jelzése
0.	C	átvitel jelzése

### Léptetéstípusok szemléltetése

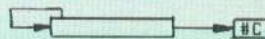
Ciklikus léptetés jobbra:



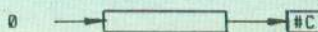
Ciklikus léptetés jobbra, #C is a körben:



"Aritmetikai" léptetés balra:

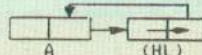


"Aritmetikai" léptetés jobbra:



"Logikai" léptetés jobbra:

Ciklikus tetrád léptetés jobbra:





### Jelölések helyettesítése

Az assembly nyelvű utasításban szereplő minden kisbetűs jelölést helyettesíteni kell az alábbi táblázat szerint. A vesszővel elválasztott helyettesíthető operandusok mellett zárójelben a gépi kódhoz OR művelettel kapcsolandó maszkokat találjuk. A gépi kód aláhúzása jelzi, hogy az adott bájtot maszkolni kell. A maszkok - az x, xh, és xl jelölés helyettesítésénél levők kivételével - az utasítás utolsó aláhúzott bájtiájához kapcsolandók. A kivételeknél a maszk az első bájthoz kapcsolódik.

#### A helyettesítések:

r: A(38), B(00), C(08), D(10), E(18), H(20), L(28)  
s: A(07), B(00), C(01), D(02), E(03), H(04), L(05)  
rx: A(38), B(00), C(08), D(10), E(18), xh(20), xl(28)  
sx: A(07), B(00), C(01), D(02), E(03), xh(04), xl(05)  
xr: xh(20), xl(28)  
xs: xh(04), xl(05)  
dd: BC(00), DE(10), HL(20), SP(30)  
qq: BC(00), DE(10), HL(20), AF(30)  
xd: BC(00), DE(10), x(20), SP(30)  
ss: BC(00), DE(10), SP(30)  
x: IX(DD), IY(FD) (első bájtt maszkja)  
xh: XH(DD), YH(FD) (első bájtt maszkja)  
xl: XL(DD), YL(FD) (első bájtt maszkja)  
b: 0(00), 1(08), 2(10), 3(18), 4(20), 5(28), 6(30), 7(38)  
c: NZ(00), Z(08), NC(10), C(18)  
cc: NZ(00), Z(08), NC(10), C(18), PO(20), PE(28), P(30), M(38)  
p: 0(00), 8(08), 16(10), 24(18), 32(20), 40(28), 48(30), 56(38)  
i: 0(00), 1(10), 2(18)  
d: 8 bites relatív érték  
n: 8 bites állandó  
hl: 16 bites állandó

#### Behelyettesítési példák

1. Tekintsük a "BIT 5, D" utasítást. A táblázatban a "BIT b, s" utasítás mellett a "CB 40" kódot találjuk. Mint a helyettesítések táblázatából kiderül, a "b" 5-ös értékéhez "28"-as maszk tartozik, az "s" helyére való D-hez "2"-es maszk. A keresett utasítás kódja ezek szerint a következőképpen alakul:

1. bájtt változatlanul CB
2. bájtt (40 or 28 or 2)=6A

2. Tekintsük a "BIT 3, (IY-8)" utasítást. A táblázatban a "BIT b, (x+d)" utasítás mellett a "DD CB d 46" kódot találjuk. Mint a helyettesítések táblázatából kiderül, a "b" 3-as értékéhez "18"-as maszk tartozik, az "x" helyére való IY-hoz "FD" maszk, míg a "d" helyébe magát a 8 bites számot kell írni. Láthatjuk, hogy az "FD" az első bájtt maszkja. A keresett utasítás kódja ezek szerint a következőképpen alakul:

1. bájtt (DD or FD)= FD
2. bájtt változatlanul CB
3. bájtt -8, vagyis F8
4. bájtt (46 or 18)= 5E

3. Tekintsük az "LD C, YH" utasítást. A táblázatban az "LD rx, xs" utasítás mellett a "DD 40" kódot találjuk. Mint a helyettesítések táblázatából kiderül, az "rx" helyére való "C"-hez "8"-as maszk tartozik, az "xs" helyére való "xh"-hoz "4"-es maszk, amely tovább helyettesítendő IY-nal, melyhez az "FD" maszk tartozik. Ez utóbbi az első bájtt maszkja. A keresett utasítás kódja ezek szerint a következőképpen alakul:

1. bájtt (DD or FD)= FD
2. bájtt (40 or 8 or 4)= 4C

A bemutatott példákban is látszik, hogy a táblázat hatékony használatához érdemes megjegyeznünk, milyen operandusokat milyen betűjelölésekkel helyettesíthetünk.



**DIGITAL**

szelvény  
Mikroszámítógép Magazin  
1988. április

## 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 al-  
katrészek (integrált áramkörök stb.) és zsebszámológépek is.

Aki ebben az üzletben a lapunkból kivágott sarokszelvényt  
átadja, vagy megrendelésével együtt oda elküldi, minden hó-  
napban más-más cikket olcsóbban vásárolhat meg. A kedvez-  
mény 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

**Egy botkormányos illesztő, fény-  
ceruzával. Ára 1200,- Ft, en-  
gedmény 5%.**

**Két botkormányos illesztő. Ára  
1800,- Ft, engedmény 5%.**

Az első változat a Spectrum gépek  
buszcsatlakozójára köthető, kapcsolós  
(Atari, Commodore típusú) botkormányok  
illesztésére szolgál. A játékprogramokhoz

az ún. Kempstone típus a szabvány, amely  
az egy botkormányos játékokhoz való. Bár  
sok játékot billentyűzetről is lehet játszani,  
de mivel így hamar tönkremennek a billen-  
tyűk, célszerűbb botkormányt használni.

A külön illesztővel csatlakoztatott fény-  
ceruzával rajzolhatunk és törölhetünk a  
képernyőn, majd a kész ábrát eltávolíthat-  
juk, magnetofonszalagra vihetjük.

A második változat, a két botkormányos  
illesztő két botkormányos játékhoz is jó,  
de nem tartalmaz fényceruza-illesztőt.

S. E.



## A Híradástechnikai Gépgyár felvesz

legalább középfokú

gépipari  
végzettséggel  
rendelkező

**gépipari  
programozót,  
gépipari  
kooperátort,**

forgácsoló  
lakatos

**normatechnológust,  
időnormást.**

Szerkezeti  
üzembe

**művezetőt,**

forgácsoló  
üzembe

**művezetőt,  
üzemi**

**koordinátort,  
árkalkulátort,**

**anyaggazdálkodót.**

Mikroelektronikában,  
illetve

műszergyártásban  
kellő tapasztalattal  
rendelkező

szakembereket,  
valamint

**gyors- és gépirót.**

Jelentkezni lehet  
személyesen vagy telefonon.  
Cím: Budapest XI., Hunyadi Já-  
nos u. 2.  
Telefon: 667-255/294 mellék.



# Közprogramok

**O**lvásóink egyre gyakrabban küldenek be olyan hasznos programokat, amelyeket terjedelmi okok miatt nem áll módunkban közölni. Mivel 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 szerkesztőségünkötől. A listák, ill. adathordozók másolását a KASZKAD Kiszérvetkezet óbudai PÓLUS szakcsoportja végzi (lásd lapunk 88/2. 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. Magán személyek a díjat a KKVMF PÓLUS Szakcsoport, 1034 Bp., Bécsi út 94-96. címre fizethetik be.

A megrendelés az alábbi formában történik:

## **MIKROSZÁMÍTÓGÉP MAGAZIN SZERKESZTŐSÉGE 1371 Budapest, Pf. 433.**

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

**Program neve**

**Csak programlistát és leírást kérek**

**Példányszám**

**Csak adathordozót kérek  
Mindkettőt kérem**

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

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

Kérjük, akik saját készítésű programjaikat felajánlják a köz javára, a részletes programleírást, a listát és az alábbiakhoz hasonló rövidített ismertetőt juttassák el címünkre. (Nevük jelenti a garanciát a továbbiakban a megrendelők számára.)



## **Osztályátlag-számító program**

Programnév: Osztályátlag-számító  
Géptípus: C16, Plus/4  
Konfiguráció: magnetofon háttértár, meg nem nevezett típusú nyomtató

Adathordozó: kazettaszalag  
Terjedelem: 348 soros BASIC program  
Készítő neve: Stumphauer Tamás  
Megjegyzés: a C16 csak bővített változat lehet

Leírás

oldalszáma: 3 + 4 oldal melléklet  
Program  
oldalszáma: 3

Maximálisan 16 tantárgyra és 40 tanulóra személyenként, csoportonként, osztályonként, tantárgyanként képes átlagot számítani. A személynevek maximális hossza nincs megadva, az osztálynév 3, a tantárgynév 20 karakter lehet. Az eredményeket az adatokkal együtt táblázatos formában ki lehet nyomtatni. Az adatok a háttértárba juttathatók, illetve onnan beolvashatók. A tanulók, tantárgyak adatai egyenként törölhetők, javíthatók, bevihetők. A program menürendszerű.

## **Orosz nyelvű feladatlap-szerkesztő program**

Programnév: Orosz szövegró  
Géptípus: Plus/4  
Konfiguráció: lemezegység (VC 1541) háttértár, nyomtató (MPS801)  
Adathordozó: 5.25" mágneslemez  
Terjedelem: 207 BASIC sor  
Készítő neve: Ködmön József  
Megjegyzés: A megadott állományok (felmérők, versenyfeladatok, megoldások) helyett egy-

szertű módosítással mást is lehet használni.

Leírás

oldalszáma: 2  
Program  
oldalszáma: 4

A program egyszerre ki nem nyomtatható példányszámban készülő orosz szövegek nyomtatására használható. (Pl. feladatlapok, versenyek anyagának készítése stb.) A szöveg képernyőn szerkeszthető, lemezen tárolható, onnan elővehető, automatikus lapozással nyomtatható. 20 soros szöveg tizenhárom perces időtartamú. A feladatlapok rendszerbe foglalhatók, utólag módosíthatók. A program menürendszerű.

## **Nagyfelbontású grafika**

Programnév: Super HI-RES grafika  
Géptípus: C128  
Konfiguráció: alappép  
Adathordozó: nincs megadva  
Terjedelem: 596 bájttal + 3 bemutató BASIC program (13, 10, 37 soros)  
Készítő neve: Dávid Zoltán  
Megjegyzés: Munka közben a parancsokat a 40 karakteres üzemmódban kell kiadni! Az ASCII/DIN billentyű lenyomása törölheti az ábrát!

Leírás

oldalszáma: 3  
Program  
oldalszáma: 6

A 80 karakter/sor felbontású képernyő-üzemmód BASIC-ből grafikus utasításokkal nem üzemeltethető, bár a VICII IC ezt lehetővé tenné. A programmal ez a hiány pótolható. A módszer: a grafikát a 40 karakter/sor felbontású üzemmódban állítjuk elő, majd átírjuk a nagyfelbontású grafikus tárolóba.



## Hardver

A cikksorozat írójának szándéka, hogy átfogó, alapfokú hardverismereteket adjon azoknak az olvasóknak, akik már tudnak számítógépes programokat írni. Előre csak minimális áramköri ismeret (Ohm-törvény) tételez fel. Mivel az írások elsődlegesen a digitális áramkörökkel ismertetnek meg, nagy előny, hogy elegendő ezek be- és kimeneti kapcsolását megérteni, a belső jelutak és kapcsolatok az alkalmazók számára közbömbösek.

A tárgyalásban — a nézőpont jellegénél fogva — nem törekedhetünk teljes precizitásra, nem mélyedhetünk el a részletekben. Reális cél viszont a funkcionális működések bemutatása.

## Áramköri alapismeretek

A jelenlegi integrált áramkörök félvezető diódákból és tranzisztorokból épülnek fel. Első lépésként ismerkedjünk meg a dióda és a tranzisztor funkcionális működésével.

### A dióda

Két kivezetést tartalmazó elem. Áramköri jelölése és a kivezetéseinek elnevezése az 1a ábrán látható. Elméletileg a dióda működését a szilícium vagy germánium félvezető anyagban vegyi úton kialakított, ún. P—N átmenet vezetési mechanizmusával magyarázhatjuk meg. A kivezetéseket vagy a diódán lévő fenti szimbólum, vagy a katód oldalán lévő jeles (festett gyűrű) alapján azonosíthatjuk.

Ha a dióda anódjára (ez kapcsolódik a P réteghöz) pozitív feszültséget kapcsolunk a katódjához (N réteg) képest, a dióda kinyit (vezet), azaz áram folyik rajta, míg ellentétes feszültség esetén nem nyit ki, azaz zár (nem vezet, vegyis szigetel) (1b ábra). A dióda jóságáról ezek szerint egy ellenállásmérő műszerrel győződhetünk meg: a dióda két vége között egyik irányban nagy ellen-

**A sorozat alapgondolata — 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.**

csí, gyakorlatilag elhanyagolható, ún. szivárgási áram folyik (Iz).

### A tranzisztor

A tranzisztor már két P—N átmenetet tartalmaz — ezért szokták bipoláris tranzisztoroknak is nevezni —, amely három kivezetéshez kapcsolódik. Jelölésük és kivezetéseik elnevezése az 2. ábrán látható. Amint az ábra is jelzi, tulajdonképpen két összekapcsolt diódának is felfogható a két P—N átmenet miatt. (Vigyázat! Ez nem azt jelenti, hogy két diódából a fenti módon összeköve, tranzisztor tudunk készíteni! Az ábra alapján egyetlen tranzisztor szakadt vagy zárlatos voltáról tudunk meggyőződni: például ohm-mérővel a bázishoz rakva az egyik mérővezetéket, a kollektor és emitter irányban nagy ellenállást, a két vezeték felcserélve kis ellenállást mérhetünk. A kollektor és az emitter között mindkét irányból nagy ellenállást kell mérni.)

Az NPN és a PNP tranziszortípus felhasználása lehetővé teszi különböző polaritású és kapcsolású tranzisztoros áramkörök kialakítását. Mi a továbbiakban az NPN tranzisztor funkcionális működését tanulmányozzuk. Az NPN tranzisztor működését a következőképpen fogalmazhatjuk meg. A bázis- (b) emitter (e) diódán átfolyó árammal (Ib) szabályozni tudjuk a kollektor- (c) emitter (e) kivezetések között folyó áramot (Ic). A tranzisztor fontos jellemzője az áramerősítési tényező, azaz az Ic/Ib arány. Ennek értéke néhányzor tízezer-

különbé, például hangfrekvenciás erősítőben.

Számunkra sokkal fontosabb a tranzisztor ún. kapcsoló üzemmódja. Ennek megértéséhez segítséget nyújt a 3. ábra.

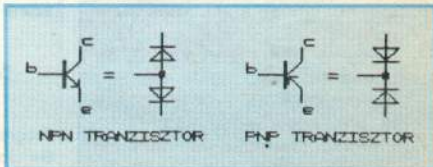
A K kapcsoló nyitott állapotában  $I_b = 0$ . Ekkor természetesen az Ic áram is nulla, és az Uki kimeneti feszültség +5 V (nem csak feszültség az Rc ellenálláson). Ha a K kapcsolót zárjuk, akkor a bázisáram nagysága  $I_b = (5 - 0,7) V / 1k\Omega = 4,3 \text{ mA}$  lesz. Itt a 0,7 V-os érték a bázisemitter dióda nyitó irányú feszültsége.

Mekkora lesz az Ic kollektoráram? Ha az áramerősítési tényezőt 100-nak tételezzük fel (szokványos érték), akkor elvileg  $100 \times 4,3 = 430 \text{ mA}$  lehetne. De a kollektor-körben lévő  $R_c = 1 \text{ k}\Omega$  ellenállás ezt az áramot nem engedi kialakulni, ezért úgy mondjuk, hogy a tranzisztor teljesen kinyit (c és e között közel nulla ellenállás lesz), és így  $I_c = 5 V / 1 \text{ k}\Omega = 5 \text{ mA}$ ! Az Ic és Ib már nem arányos egymással. Azaz ha  $U_{be} = 0V$ , akkor  $I_c = 5 \text{ V}$ , és ha  $U_{be} = 5 \text{ V}$ , akkor  $I_c = 0V$ .

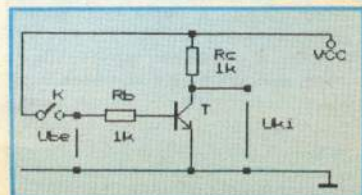
## Logikai áramkörök

Azok, akik számítógépes programokat írnak, ismerik a logikai műveleteket. Elemi állítások igaz, illetve hamis voltához az 1 és 0 értékeket rendelve felépíthető egy kétértékű, más néven bináris vagy Boole-algebra, ami lehetőséget ad összetett állítások igaz vagy hamis voltának megállapítására.

2. ábra



3. ábra



1. ábra

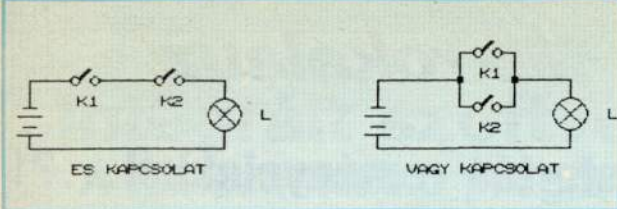
állást (szakadást), a mérővezetékeket felcserélve kis ellenállást (rövidzárt) kell mutatnia a műszernek.

A diódán nyitó irány esetén kb.  $U_{\theta} = 0,7 \dots 1 \text{ V}$  nagyságú feszültség esik. Ezt a dióda nyitó irányú feszültségének nevezük. Ebből még az is következik, hogy a dióda akkor képes csak kinyitni, ha  $U > U_{\theta}$ !

A dióda zárt állapotában csak nagyon ki-

többezseres nagyságrendig terjedhet. Ezzel magyarázható a tranzisztor erősítőképesége: kis bázisárammal tudjuk az Ic kollektoráramot vezérelni. Amikor a tranzisztor úgy használjuk (a többi áramköri elemet úgy választjuk meg), hogy a bázisáram és a kollektoráram egymással arányos, akkor a tranzisztor lineáris üzemmódban működik. Ezt a tulajdonságát érvényesítik a





4. ábra

Az elemi  $\theta$ , illetve 1 értékű eseményekhez egy kapcsoló nyitott, illetve zárt állapotát rendelve, kapcsolókból felépített áramkörrel valósíthatjuk meg az összetett események elektromos leképvezését. A 4. ábrán látható áramkörök a K1 és K2 kapcsolók állapotához rendelt elemi állítások  $\theta$ S, illetve VAGY kapcsolatot valósítják meg.

$\theta$ S kapcsolat: a lámpa akkor ég, ha K1  $\theta$ S K2 is zárt (mindkét elemi esemény igaz). VAGY kapcsolat: a lámpa akkor ég, ha K1 VAGY K2 VAGY mindkettő zárt (valamelyik vagy mindkét elemi esemény igaz).

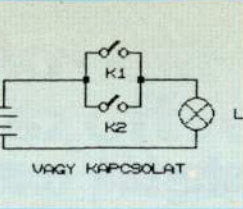
Logikai funkciókat megvalósító áramköröket — kapcsoló tulajdonságait kihasználva — félvezető diódákból, illetve tranzistorokból is felépíthetünk. Az így kialakított áramkörökben a  $\theta$ -át és 1-et mindig két feszültség szint képviseli, amelyeket L és H szinteknek is szoktunk nevezni. A leggyakoribb esetekben a két logikai állapothoz úgy rendelünk feszültség szinteket, hogy L-nek a  $\theta$  V, H-nak a pozitív feszültség (legtöbbször 5 V) feleljen meg. Ezt a megfeleltetést szokták pozitív logikának nevezni.

Kapuk

Diódák felhasználásával az előbbieknél kapcsolókkal bemutatott  $\theta$ S valamint a VAGY kapcsolatot a 5. ábrán vázolt, ún. diódás kapuáramkörökkel valósítható meg. Vizsgáljuk meg először az  $\theta$ S kapu működését. Ha az A és B jelű bemenetek mindegyike L szintű, akkor a két dióda vezet, ezért az X jelű pont (a kimenet) L szinten lesz (pontosabban  $U_{\theta} = \theta,7$  V nyitóáramú feszültségen). Ha csak az egyik bemenet L szintű, akkor is vezetni fog a hozzá kapcsolódó dióda, azaz X továbbra is L szintű marad. Ahhoz, hogy a kimenet H szintű legyen, mindkét bemenetre H szintet kell kapcsolni. Akkor a diódák lezárnak, és az R ellenálláson áram nem folyik, ezért a kimenet H szintű lesz.

A VAGY kapu működése az előzőhöz hasonló módon elemezhető. Ha a bemenetek L szinten vannak, akkor az ellenálláson áram nem folyik, így X pont feszültsége nulla (L). Ha azonban valamelyik bemenetre H szintet kapcsolunk, akkor a dióda vezetni fog, az R ellenálláson áram folyik, és a kimenet H (pontosabban  $V_{cc} - \theta,7$  V) szintű lesz.

Ilyen diódás áramkörökkel bonyolult logikai függvények is megvalósíthatók, de előnyösebb tulajdonsága miatt a tranzisztort mint kapcsoló elemet alkalmazó áramkörök váltak a legjobban elterjedté. Csak példaként egy előnyösebb tulajdonság: logikai áramkörökben igen gyakran kell egy adott jel ellentettjét — negáltját — előállítani. Az ezt megvalósító áramkör — az inverter — diódával való előállítása meglehe-



5. ábra

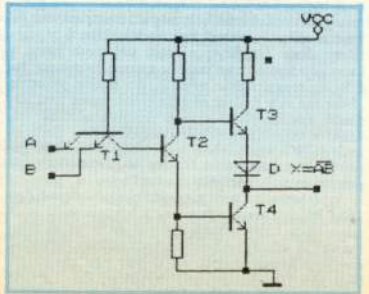
tősen bonyolult; tranzisztorral ez a legegyszerűbb feladat, hiszen a tranzisztor kapcsoló üzemmódját bemutató 3. ábra egy inverter: ha a bemenet  $U_{be} = \theta$  V (L szint), akkor a kimenet  $U_{ki} = 5$  V (H szint), és ha  $U_{be}$  H szintű, akkor a kimenet L szintre vált.

A tranzistorokat alkalmazó TTL (Tranzisztor-Tranzisztor Logika) igazi diadalát az integrált áramkörök megjelenése hozta. Az integrált áramkörök olyan — kivezetéssel ellátott — tokozott félvezető lapka, amelynek felületén sok-sok tranzisztor és dióda van működőképes áramkörre egyesítve, vagy más szóval integrálva. Mi ezeket az áramköröket csak a kivezetéseik felől „látjuk”, ezért alkalmazásukhoz elegendő funkcionális működésük ismerete. Illusztrációként elemezzük egy két bemenetű NEM- $\theta$ S (NAND) TTL kapuáramkör működését (6. ábra).

Ha egy vagy mindkét emitter L szinten van, akkor a bemeneti R ellenálláson áram folyik (a T1 bázisemitter diódája vezet). Ennek következtében T1 kollektora L szinten lesz. A kollektor csak akkor lesz H szintű, ha valamennyi bemenet H szintűvé válik. Így a bemeneti rész tulajdonképpen  $\theta$ S kapcsolatot valósít meg. A következő fokozatot a T2 tranzisztor alkotja, ami az ún. totem-pole kimenet (T3—T4 tranzisztor, D dióda) megfelelő vezérlését (meghajtását) biztosítja. Amikor T2 vezet (T1 bemenetei H szinten vannak), akkor T3 lezár és T4 kinyit, ezzel a kimenet L szintű lesz, ami a NEM- $\theta$ S funkciónak felel meg. A kimeneti láncban lévő D dióda biztosítja, hogy T3 lezárt állapotban legyen akkor, ha T4 kinyit. Ha T2 nem vezet, akkor T3 van nyitva és T4 zárva.

Méréssel a fenti működést úgy ellenőrizhetjük, hogy a föld és az X kimenet közé egy voltmérőt kötünk; akkor annak közel  $\theta$  V-ot (L szint) kell mutatnia. Az A és B bemeneteket egyszerre vagy külön-külön a  $V_{cc}$  feszültségre kötve, a voltmérő közel 5 V-ot (H szint) fog mutatni.

6. ábra



Más jellegű kimenet is kialakítható a fenti totem-pole kimenet helyett: az ún. nyitott kollektoros kimenet. Ilyenkor a \*-gal jelzett ellenállás, a T3 jelű tranzisztor és a D dióda hiányzik a kimenetről, és az X jelű kimenet (T4 kollektora) van kivezetve. Ezért hívják nyitott vagy szabad kollektoros kimenetnek. Milyen tulajdonságai vannak az így kialakított kimenetnek?

— A T4 tranzisztor így nagyobb tápfeszültségen működő terhelést is képes kapcsolni: például egy 12 V-os izzót vagy relét, azaz TTL bemenetről vezérelhetünk nem TTL jellegű kimeneteket. Természetesen ilyenkor a T4 tranzisztoron átfolyó áramnak a megengedett érték (kb. 20 mA) alatt kell lennie.

— Amikor az ilyen nyitott kollektoros kapuk kimeneteit összekötjük, és a közös vezeték a  $V_{cc}$  tápfeszültséghez csatlakoztatjuk, akkor ez a közös vezeték (X kimenet) csak abban az esetben lesz H szintű, ha valamennyi kimenet H szinten van. Ha csak egyetlen kimenet is L szinten van, akkor a közös kimenet is L szinten lesz, azaz a kimenetekkel  $\theta$ S kapcsolatot valósíthatunk meg. Az ilyen összekötött kimeneteket huzalozott  $\theta$ S (más megoldásnál huzalozott VAGY) kapcsolatnak is nevezik.

Logikai hálózatok

A digitális TTL kapuáramkörök ilyen és ehhez hasonló áramkörökből épülnek fel. Az ilyen kapuk egyértelműen jellemezhető bemeneteik számaival és azazal, hogy a bemenetek milyen logikai függvénye jelenik meg a kimeneten. Az integrált áramkörök összekapcsolása során gyakran több bemenetet kell egy kimenetre kapcsolni. TTL áramköröknél megadható, hogy egy kimenet maximálisan hány bemenettel terhelhető. Ez a szám a kimeneti terhelhetőség (fan out). Előfordulnak olyan bemenetek is, amelyek a kimenetet a szokásosnál jobban terhelik. Az ilyen áramkörök bemeneti terhelése (fan in) nagyobb mint 1.

Ezek alapján TTL áramkörökből álló logikai hálózatokat kapuáramkörökből a kimenetek és bemenetek megfelelő összekapcsolásával hozhatunk létre. Néhány alapszabályt azonban be kell tartanunk:

- egy kimenet maximum tíz bemenetet képes működtetni (fan out = 10);
- kimenetek közvetlen módon nem köthetők össze;
- a fel nem használt bemeneteket célszerű valamelyik tápfeszültségpontra (GND vagy  $V_{cc} = 5$  V-ra) kötni.

(Folytatjuk)



## Lépéstartás reményeinkkel...?!



kedelemből 16 ezer, míg idehaza 12 ezret gyártottak. S itt egy adat láttán elgondolkoztunk, ki és mi a kereskedő: ugyanis 1984-ben mindössze ezer gép jutott a külkereskedelm jövőléből a felhasználók birtokába. Addig egy szál sem!

Dicsőretünkre legyen mondvá, hogy a házi számítógépek gyártását a szocialista országok közül elsőként Magyarországon kezdték meg 1982-ben, s mind ez ideig csak nálunk gyártottak belülről ezernél többet. Ebben meghatározó volt a már 1983-ban számítógépet gyártató középiskolai számítógépprogram és a Prímó debütálása, majd szorozatgyártása 1984-ben. Oh, Prímó! Gyorsan jött, gyorsan hervadt. 1987 februárjában bekövetkezett a végkiárusítása, a Prímó gyártói halálát még az előző év szeptemberében bejelentették. Pedig „A Prímó harminc hónapos életét értékelve elismeréssel kell szólni a kezdeményezés úttörő jellegéről, a kitűnő piaci munkáról (a szocialista országok első, boltban árusított gépe címet is elnyerte). Az élete rövidségéért okolható az elterjedt nyugati típusokkal való inkompatibilitása, a megbízhatatlan program ki- és bemenete, a mindezek következtében gyenge programellátottsága. Ez utóbbi a programok ez-reivel kényeztetett C64 és Spectrum gépekkel szemben óriási hátrányt jelentett. Magyarul: nem lehetett má más csupasz vasat értékesíteni.”

Aztán 1986-ban a Videoton robbantott, robbantott. Megjelent a TV-Computerrel a boltokban. Bár a gépet 1984-ben bemutatják, végül két esztendőre rá jelent csak meg a forgalomban. Mi lett volna, ha Pallasz Athéné ennyit vajúdik Zeusz fejében! Bizonyára a legfőbb isten adata-ganattal küldött volna. De hát egy isten még a gyorsaságot is megengedheti magának. — Mindenesetre a Videoton a teljesítménykategória visszacsúsztatása ellenére igazságot az igények-kelést tartani, amiért is gazdag periferiaválasztékot kínál, illetve ígér.

Az árak, a kereslet-kínálat és a beszerezhetőség hullámzása fokozatosan csökkent. De a gyors süllyedés után mára megállapodtak. Így például egy C64 25 ezer, egy ZX-Spectrum 48 k 12 ezer forint körül állt az ár, a beszerezhető tavalé év végén. Végül az összegzésből kiemelhető, hogy a házi számítógépek 1986-tól a piac-on túlnálkint jellel mutatkoznak, a vevők válogathatnak a típusok között, s míg a házi számítógépek-állomány gyarapodás útjeme mérséklődött, az iskolásztámítógépek-állomány jelentősen megnőtt. Egyébként az iskolásztámítógépek-program, ahol dominál a Commodore Plus/4 és a TV-Computer, azzal számol, hogy 1990-ig 50 ezer gépet juttat a közép- és általános iskolák birtokába.

A professzionális mikrogepek gyártása a szocialista országokban a 70-es évek végén, 80-as évek elején kezdődött meg. A 8 bites gépek gyártása gyorsan ivelt felfelé, míg a 16 bites megje-lentése szinte a törlőkódz beodobásra kényszerítette őket. Ugyhogy a majd két tucat hazai 8 bites gép listáját már majdhogynem műzeumi, ha a 86-os forgalmukat nézzük. Az árakat tekintve „Már ma is a 8 bites és a 16 bites gépek árai hasonló konfiguráció esetén összemehetnének, viszont a teljesítménykülönbségük óriási. Ezért is emlegetik a szakmában úgy, hogy a 16 bites gépek árai rázuhatnak a 8 bites professzionális gépekre és agyonütötték azokat.”

A 16 bites közül a meghatározó jelentőségű IBM PC-vel kezdődően a vele kompatibilis mikrogepek a honfoglalók. Itt milyen a hazai állomány? A magánember óvatlanságában igencsak savanyú képet vághat, ha behoz egy gépet. „A várn egy kis konfiguráció esetén is 300-500 ezer forintot tesz ki. A magánéletű vásárlás ezért szin-

te kizárt.” A külkereskedelmi forgalom tank-csapdája viszont a Cocom-rendelkezések sora. Ennek szellemében csak 1985 tavasza óta engedélyezik IBM PC-k kivételét, illetve behozatalt hazánkba, azóta is csak egyedi exportengedél-lyel. 1986 újdonsága volt az IBM PC-vel kompa-tibilis magyar gépek szocialista exportjának be-indulása. Aminek egyik élenjárója a Proper és továbbfejlesztésének eredményeképpen kis család lett.

Az árak alakulása belsőszervi betegségekre utal. Hasonlattal élve: a vasszerkezet megrogy-gyan anyaghiba miatt, avagy hamissággokra épül-e a lélek is megkeseredik, olyannyira, hogy a cselekedetek is sántikálnak. Magyarán: „Az 1986-os év főszereplője ebben a kategóriában az ár volt. Komoly túlnálkint alakult ki, így akinek jelentősebb készletei voltak akár gyártásból, akár külföldi beszerzésből, forgalma növelése érdekében erőteljesen csökkentette az árait.” De ez a tendencia megfordult novemberre. A több ok sűrűmánya 1987-re: „A potenciális vevők a kedvezőtel (sic!) vámintézkedések miatt egyre szű-külő kínálatból válogatás helyett fantomárú fantomgepek kezdtek elődni. Közben pedig a tényleges vásárlások száma, az alkalmazásba vétel szeptemberig szinte a nullára apadt!” A Mikrokalauz az okok láncolatában minden-estre kellő alaposággal eligazít. A netán lappo-dkedve talán visszazökkenheti eredeti illúzió-jába az „Elektronika társadalmi-gazdasági alkal-mazása elterjesztésének gazdaságirányítási ele-menzeit átfogó központi Gazdaságfejlesztési Program” keretében kínálkozó kedvezmények eredményeként és legfőképpen a lehetőségeink áttekintése. E részben végezett az IBM PC-vel kompatibilis lokális hálózatokig táblázatokon te-kinthetjük át a gyártókat, a termékeket és azok árát. Hozzákapcsolva néhány egyedi típust egy záró megjegyzés kíséretében: „... a hazai gyár-tók komolyan veszik az IBM PS/2 családját, s erőteljesen folynak az azzal kompatibilis gépek előállítására irányuló munkák.” — Amihez nem fér kétség, ámbar nem ártana mindannyiunknak komolyan venni magát!

A periferiák igazi csemegéi a számítógépepek-nak. Annál inkább, mivel a nyomtatók, a win-chester-tárak, a folyamatos üzemű mágnesszalag- táraz (streamerek) legkorszerűbb fajtáiból a hazai piacon kevés akad, ha nem éppen újdonság, és az egyszerű nyomtatókon kívül borsos az árak is. Minderről a Mikrokalauz szívből jó érzékkel tájékoztat, és külön kötetben bön-gészhetünk a teljes választékban. Felvilantva még a lizingnek, amely olyan bérletreztést jelent, melynek végén a bérlet eszköze egy jelképes ösz-szegért a bérítő tulajdonosa lesz, remélhetőleg sikeres számítógépes jövőjét is. — Am számítógépek ide, periferiák oda, e helyütt Pascal teljesen réa-lis: „A múlt és a jelen csupán eszműközik: célunk csakis a jövőendő. Így tehát sohasem élünk, ha nem csak remélünk élni.” — A kérdés csupán az: saját reményeinkkel lépést tudunk-e tartani?

SZULYOVSKY CSABA

**Küldjétek figyelmé!  
Mikroszámítógépet  
akarnak vásárolni?  
Tájékoztódnak  
a naprakész piaci helyzetről!  
Díjtalan ismertető!  
MÉS  
Számítástechnika  
1368 Budapest, Pf. 193.**

Egy régen élt filozófus arra a kérdésre: Elgon-dolható-e egy olyan világegyetem, melyben nincs élet? nyilvánvalóan azzal válaszolt volna — nem. Hiszen nincs értelme világegyetmről beszélnünk, ha senki sincs, aki beszélne róla. Az ember középponti helyének kopernikuszi trón-fosztása óta azonban első hallásra könnyen igen-lis és válaszolhatnánk. Mégis Einstein egyik neves tanítványa, John A. Wheeler egyértelműen nemet mond. Állítván, hogy nemcsak az ember alkalmazkodott az univerzumhoz, hanem az uni-verzum is az emberhez. Ez az állítás persze nem vezet minket egyenesen a tyúk vagy a tojás di-lemmájához. Ezzel ellentétben még egocentri-kusságunk vagy rosszabb esetben egoizmusunk sem mondhatnánk velünk, hogy létezésünknek nem egyik alapfeltétele alkalmazkodóképesség-ünk, s noha az iménti gondolatfutamhoz képest kissé nyers fordulatnak tűnik, ne vegyük zokon, de az alkalmazkodóképesség nagyon is kifejez-hető egy adott ország, jelen esetben hazánk szá-mítástechnikai, pontosabban mikroszámítógépes piacának helyzetével. Mert hiszen, prózáibbra fordítva első kérdéstünk: Elgondolható-e egy legkorábbi dinamikus fejlődése korszakú vagy ergorszerűbb számítógépek nélküli? — erre egyértelműen nemmel válaszolhatunk.

És noha a Mikrokalauz hazai mikroszámítógé-pek piacáról szóló tanulmány szerzőit nem társadalmi és gazdasági hatások és behatároltsá-gok elemzése irányítja, a tájékozódó vagy éppen a vásárlásnak nekirugaszkozó olvasó a „képzelt” következtetéseit elől nem menekülhet. Mert, no-ha mikroszámítógépes állapotainkra ráhúzni, hogy: non progridi est regredi (nem haladni annyit, mint visszafelé menni), szélsőséges állás-foglalás volna, az már az előbbi szövegen apró fordítve prozaivan valós, hogy nem eléggé gyors-haladni annyit, mint erősen lemaradni.

A kötetben a mikroszámítógépeket a mikro-processorok által egyszerre kezelt bitek száma szerint csoportosítva mutatják be a szerzők. Rög-tön megállapítva, hogy ma már a kis teljesítmé-nyük miatt a 8 bites gépek az életciklusuk leme-nő ágában vannak. Ide tartoznak azonban a házi számítógépek és iskolai számítógépek, amelyek nagyobb teret kaptak az összeállításban. És ami azzal szembe szúr: az alacsony sebességű, kis operatív táruék és korlátozott periferiellátottsá-guk miatt a házi számítógépeket általában csak oktatásban, jatekra és a háztartásokban használ-ják ma Nyugaton. Hazánkban viszont elsősor-ban a C64-et széles körben professzionális célok-ra is. „A fejlett többes országokban ezt a teljesí-ménykategóriát már csak jateknak tekintik. Ugyanis a 16 bites gépek árai is 2-3 havi átlagfi-zetésre mérséklődtek. ... — természetesen jóval többet tudnak kiösszekülni. És a mai házi számít-gépek teljesítmény kategória megkapja a kegye-lemdíjást a most megjelenő IBM gyártmányú JX3-tól. Aminek mikroprocesszora Intel 8088, operatív táru 256 kb-ot, az alapgép tartalmaz egy 3,5 hüvelykes, 1 Mb-ajt kapacitású hajlékonyle-mezes tárolót is. Árát várhatóan 600 dollár körül lesz.

A hazai összetétel? Képet kapunk róla a be-szerzési források, típusok és árak szerint is. Az egyéni utazások révén jutott mindmáig a legtöbb gép az országba, mintegy ötvenezer. A külkeres-



# Integrált szoftver

## A táblázatkezelő program használata II.

Az előzőekben ismertettük a táblázatkezelő utasításkészletét, és néhány egyszerű példával szemléltettük a működését. Egy összetett feladattal most megpróbáljuk illusztrálni a BASIC nyelven megírt, viszonylag egyszerű táblázatkezelő program nyújtotta lehetőségeket.

Mielőtt a táblázat összeállításához hozzáfognánk, a /T utasítással törölhetjük a táblázatot, amelyen eddig gyakoroltunk. Először állítsunk elő egy egyszerű táblát, amelyen egy számsorozat összegét számítsuk ki, majd határozzuk meg az összegre vonatkoztatott százalékokat (1. ábra).

Első lépésként írjuk be a megfelelő helyre a címkeket. Az 1. ábrán látható táblázat kialakításához mozgassuk a kurzort az AA pozícióba. Mielőtt a szövegírást elkezdjük, nyomjuk le a szökőgombot: ez jelenti a program számára, hogy címke meg-

löljük a cellánkénti összeadást. Mivel teljes összegről van szó, akad egyszerűbb lehetőségünk is:

@SUMOBD. Nézzük egy kicsit részletesebben.

@SUM a szummázó utasítás, O az oszlop szerint végzett összegzést jelenti, és BD a kezdő pozíció helye. Az utasítás hatására a BD pozíciótól kezdődően a program az oszlopban található minden értéket összead, és az eredmény megjelenik a kurzor által kijelölt pozíción (BM). A képernyő második sorában az összegző formula látható.

Most végezzük el a százalékszámításokat. Lépjünk jobbra a C oszlopba és mozgassuk a pozíciójelzőt a D sorba. Írjuk be a százalékszámítási formulát. Vegyük a BD mező tartalmát és osszuk el a BM mező tartalmával, majd szorozzuk meg 100-zal. A formula tehát a következő: +BD/

érhetjük el: shift gomb lenyomása után le kell nyomni a felkiáltójelét. A program most újraszámolja a táblázatot és kitölti a megfelelő helyeket.

A 2. ábra egy negyedévi táblázatot mutat be. Láthatóan itt már nemcsak az oszlopok szerint, hanem a sorok szerint is végzünk számításokat. A képernyőn a jobb felső sarkban megjelenő 0: (oszlop) vagy S: (sor) jelzi, hogy a kijelölt műveletet most oszloponként vagy soronként veszi a program. A soronkénti számolás utasítása /SZS, ahol / azt jelenti, hogy utasítás következik, SZ jelentése: számolj, S: soronként.

Ne felejtjük el, hogy egy táblázat számításánál a mezők tartalmának már létezniük kell, mielőtt egy formulával műveletet végeznénk a táblában. Ez különösen fontos akkor, ha az egyik formula használja a másik által kiszámított értékeket.

Ha a nyomtatóra akarjuk küldeni a táblázatot, akkor a számítások elvégzése után (a tábla ki van töltve) a /N (nyomat) [ENTER] utasítás megadásakor a program megkérdezi a kezdő oszlop helyét. Esetünkben ez az A oszlop volt. A táblázat minden sora nyomtatásra kerül.

Végeztül még néhány szó a fájl szerkezetéről. A táblázatot tartalmazó fájl három rekordból épül fel. Az elsőben található a

?	[BM]	(F)	@SUMOBD
A	Havi kiadások		
B	Megnev.	összege	Százalék
D	Lakás	1250.00	10.13
E	Közmű	845.00	6.85
F	Hitel	2400.00	19.46
G	Ruházat	1338.00	10.84
H	Élelem	3350.00	27.15
I	Közlék	1375.00	11.14
J	Egyéb	680.00	5.51
K	Takarék	1100.00	8.92
L	-----		
M	TOTAL	>12338.00	100.00

1. ábra

adása következik. A szöveg (a címke) bevételét egy ENTER lenyomásával lehet lezárni. Léptessük a kurzort az AB pozícióba, ismét szökőz lenyomása után írjuk be a kívánt szöveget, majd az ENTER ismét lezárja a címkeket. A címke szövege „átlógathat” a következő pozícióba is, ha ezt például a táblázat fejléce így kívánja.

Az említett eljárást folytatva írjuk be minden pozícióra, amit akarunk. Az aláhúzások, amelyek a táblázat jó tagoltságát eredményezik, könnyen képezhetők: /- (kötőjel) a program számára azt jelenti, hogy a jobb oldalon, a kötőjel mellett álló karaktert ismételni kell (például - = azt jelenti, hogy a kurzorpozícióval megadott cellához egyenlőségjel kerüljön a cella teljes szélességében, mint címke).

Az értékek beviteli módját már az előző alkalommal ismertettük. Csak emlékeztetőül: először a tizedespont helyét kell megadni (jelen esetben a /TP2 utasítással). Ezek után vigyük be egymás után a megfelelő pozícióra a számértékeket. Az utolsó pozíció a BK lesz. Vigyük most lejjebb a kurzort, az MB pozícióra. Itt most két lehetőségünk van az összeg előállítására. Az egyik +BD +BE +BF... stb., tehát kije-

?	[BM]	(F)	@SUMOBD		
A	Havi kiadások				
B	Megnev.	január	február	március	összesen
D	Lakás	1250.00	1250.00	1250.00	3750.00
E	Közmű	845.00	1050.00	1124.00	3019.00
F	Hitel	2400.00	2400.00	1100.00	5900.00
G	Ruházat	1338.00	1209.00	1500.00	4046.00
H	Élelem	3350.00	3600.00	3950.00	10900.00
I	Közlék	1375.00	1420.00	1390.00	4185.00
J	Egyéb	680.00	980.00	1200.00	2860.00
K	Takarék	1100.00	1000.00	1000.00	3100.00
L	-----				
M	TOTAL	>12338.00	12908.00	12514.00	37760.00

2. ábra

BM+100. [ENTER], az eredmény megjelenik a CD mezőben. Hasonlóan írjuk be a többi pozícióra is a formulákat. Az eredmények rendre megjelennek a kijelölt mezőkben.

Lehetőség van egy másik megoldásra a /M (másol) utasítás használatával. Az utasítás a következő alakú lesz: /MCDCK, ami azt jelenti, hogy másold a CD mezőben található formulát a CK mezőig minden pozícióba.

Természetesen most ki kell javítani a százaléktételeket tartalmazó mezők koordinátáit a megfelelőkre. Tehát például az E sorban a BE helyére BE kerül és így tovább. Ez az utasítás hasznos, ha bonyolult formulákkal dolgozunk és csak kis változások szükségesek az egyes pozíciókon.

Miután elvégeztük a szükséges módosításokat a képernyőn, visszatérünk a normál üzemmódba. Ezután, ha kívánjuk, ellenőrizhetők a mezőkön található formulák. Léptessük lefelé a kurzort a C oszlopban. Az aktuális számítási képlet megjelenik a képernyő második sorában. Láthatóan a számítás eredményei is a helyükön vannak, de semmi nem történt a táblázatban. Még nincs kitöltve. Ezt egy egyszerű utasítással

kurzorpozíció a „kimentés” pillanatában, a táblázatban használt tizedes jegyek száma és a számolás rendje (oszlop vagy sor). Például BE,2S azt jelenti: a kurzor a BE pozíción volt, két tizedesjegy törekszünk a pontosságára a számításnál, és a műveleteket soronként végezzük. A harmadik rekord a fájlt zárja le (EOD = End of Date). A közbenső rekord tartalmazza a táblázat adatait. Amint már említettük, 676 mezős a táblázat. Mindegyik pozíción tulajdonképpen három mezőtípus található (címkek, formulák és változók értékei). A rekord a következő formában épül fel: oszlop, sor, típus (... adatok...).

Az oszlop és a sor megnevezése A-tól Z-ig terjed, a típus L (címke), F (formula), V (változó) lehet. Az adatok sztring formában lesznek rögzítve: a program betöltésükkor az adatokat sztring formában és számszámszerűen tárolja.

Mivel a rekord tartalmazza minden mező pozícióját és tartalmát, a rekord előállítására például egy másik programmal elég egyszerű feladat. A szükséges dimenzióállításokat a programlista 2. sora tartalmazza (lásd '87/12. számunkat).

GALINA FERENC



# ADOM A MAGYARÁZATOT!

Lapunk előző számában Gódér György olvasónk a C16 fontosabb alkatrészei látkiosztásáról kértet. A válasz:

1. A C16 mikroprocesszorának kéredezett kivezetései:

**AEC:** Bemenet. Ha valamilyen külső egység használja akarja a cím- és adatbuszt, akkor az AEC lábát alacsony szintre állítja, és ennek hatására a mikroprocesszor lekapszolja magát a buszról.

**RDY:** Bemenet. Ha alacsony szintű, akkor ez jelzi a mikroprocesszornak, hogy az adatbuszon lévő információ még nem érvényes, ezért még nem olvashatja be. A C16-on nincs kihasználva ez a bemenet.

**TRQ:** Maszkolható megszakításkérés bemenet. Ha a CLI gépi utasítással a megszakításkérés engedélyezve van, és valamilyen külső egység a lábát alacsony szintre állítja, akkor a mikroprocesszor a \$FFFE—FFFF memóriacím által mutatott megszakításkészítő rutinra ugrik.

**GATE IN:** Ezt a jelet a TED állítja elő. A dinamikus memóriákat meghajtó IC-k számára azt jelenti, hogy a cím alsó vagy a felső bájttját továbbítsák a RAM-okhoz. A mikroprocesszornak pedig jelzi, hogy mikor készek a RAM-ok az adatbusz információjának fogadására.

**PO:** A mikroprocesszor (µP) órajele. A TED állítja elő. PO—P7: A µP saját 7 bites porttal rendelkezik, aminek ezek a kivezetései. (P5-8s nincs.) Programból a memória \$0000—0001 címeken érhetjük el. A \$0000 cím a port bitenkénti irányítást megadja. (1 = kimenet, 0 = bemenet)

A \$0001 cím en írhatunk, illetve olvashatunk a portról. (A 6510 adatlapján ez is, és a fordítottja is megtalálható. A két változat közül ez az ellenőrzöten helyes.)

A C16-ban ezeket a portbiteket a soros busz és a magnó kezelésére használják.

2. A TED vezérli a C16 összes fontos funkcióját, ami elengedhetetlen a számítógép működéséhez. A TED a C16 legfontosabb alkatrésze.

**Feladatai:**

— a megfelelő memóriaszelet kiválasztásának vezérlése.

CSO, CSI/

— dinamikus RAM-ok felfrissítése

— a rendszerorajel előállítása (Φ0)

— RAM-ok címzésének vezérlése (CAS, RAS, MUX)

— billentyűzetleolvasás (K0—K7)

— a teljes képvezérlés: színes videojel összetevőinek előállítása (SYNC, LUM, COLOR)

— hanggenerálás (SND)

— időzítők és rasztermegszakítás kezelés (TRQ)

**Kivezetései:**

AEC, BA: mint a µP-nál az AEC és RDY kivezetés a TED-re vonatkoztatva.

CSO, CSI: a címbusz alapján a megfelelő ROM, ill. RAM IC-t engedélyezi. Normál (bővítetlen) esetben a BASIC és KERNAL ROM között választ.

COLOR, SYNC/LUM: A kimeneti videojel szín, szinkron- és fényességjel összetevői.

SND: a hanggenerátor AUDIO kimenete.

KO—K7: bemenetek, a billentyűzetmátrix oszlopainak értékét és a botkormányok állapotát olvassa be a billentyűzet litch TED regiszterbe (\$FF08).

3. U16-os IC (\$7700—10):

Ez egy logikai hálózatot tartalmazó be-rendezésorientált áramkör, ami a 16 különböző bemenetétől függően változtatja 8 kimenetének állapotát. Egy I/O memóriaterület megcímezésekor a kimeneteken engedélyezi a megfelelő egység adatbuszra kapcsolódását.

PI: kazettás egységen gombnyomás érzékelése,

billentyűzet sorainak vezérlése,

memóriabank-kiválasztás.

(Az I/O terület helye: \$FD00—FEFF 16 bájtos részre osztva.)

4. 6529 B:

Ez egy egyszerű 8 bites regiszter, ami a C16-ban a billentyűzetmátrix sorait hajtja meg. A lenyomott billentyű hatására a TED K0—K7 bemenetein a megfelelő helyen megjelenik a 6529B IC-nek a lenyomott billentyű sorához tartozó kimenetállapota és a TED \$FF08-as regiszterének olvasásával a lenyomott billentyű megállapítható. A 8 bites IC az I/O terület \$FF30 címén érhető el.

A C16 számítógép kapcsolási rajza megtalálható a RÁDIÓTECHNIKA 1986. áprilisi számában.

GYURKOVITS GERGELY

## ADOK—VESZEK—CSERÉLEK

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hírdetések közlik. 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. Hírdetéseiket a szerkesztőség címére várjuk.

**ADOK**

**ATARI 800 XL** (64k) magnóval, joystickkal, játékprogramokkal 16 000 Ft-ért eladó. Kovács László, Alsódszolca, Bajcsy Zs. u. 40. 3571

**ATARI 800 XL/XE**-hez monitorprogram kapható. Assembler-Disassembler, 12 monitorparancs. Ára: postán utánvéttel 300 Ft/kazetta. Cím: ifj. Nyistár István, Gyula, K. 190. 5700

**ATARI 800 XL** magnóval, játék- és felhasználói programokkal, izósor-vezérlővel, szakkönyvekkel, ROM-listával, jegyzetekkel eladó. Bódy Attila, Kazincbarcika, Pollack M. u. 3. 3700. Telefon: 06-48-14-643 az esti órákban.

**ATARI 520 SIM** eladó (512k) floppyval (360k), egérrel, sok programmal (68 ezer), hozzá profi fi monitor (20 ezer). Dr. Nagy József, Eger, Dr. Sándor I. u. 3. IV/9. 4139

**C64** eladó + magnó, joystick, programokkal + Commodore Microdrive, rend-

szerszöftver nélkül, kazettákkal, ára: 21 000 Ft. Simon Gábor, Miskolc, Gömöri u. 10. 3526. Telefon: 15-379

**Commodore VC20** + 16k bővítés + magnó + 3 db cartridge 3 db DEMO-kazettával + sok gyári programmal eladó. Ár megegyezés szerint. Kovács Zolt, Budapest, Táncsics Mihály u. 63. 1211

**VIDEOTON TV-Computer** (32 k) + rendszerszoftver 9000 Ft-ért eladó. Balogh Endre, Budapest, Bertóti u. 14. 1172. Tel.: 639-343

**TV-Computer** (32 k) eladó programokkal (kazettán). Kuczi Géza, Katymár, Sallai u. 7. 6455

**VESZEK**

**ATARI 800 XL**-hez keresek adatrögzítő magnót, lehetőleg 1010-es típusú. Boros Elemér, Pécs, Vöröskő u. 14. 7634

**Plus/4-re** beszélő program után érdeklődök. Szabó Tibor, Miskolc, Szentpéteri kapu 815/1. 3526

**D64-re** Contract bridge programot keresek (lemezen) megvételre vagy cseréként. Dr. Bóka, Várpalota, Kulich u. 25. 8100

**TI-99/4A** számítógéphez gépkönyvet, kiegészítőket és programokat veszek. Ajánlatokat a következő címre kérem: Simon Árpád, Tatabánya, Sárberk 707. III/1. 2800

**CSERÉLEK**

**C16** programokat cserélek. Illosvay Zoltán, Budapest, Futórézsa u. 86. 1165

**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 jelentkeznek be, és gombnyomásra a kiválasztott program azonnal fut. Javaslott program-csomagok: Turbo tape, Assembler, Monitor, Supergrafik, Help plus, Turbo másoló. File másoló 1699 Ft és Turbo tape 699 Ft. Hozott programok elhelyezése a GYORS-HÁTTÉRTÁR-ban. Trompler László, Budapest, Attila u. 22. 1201. Tel.: 287-493 este.

**Enterprise** programokat cserélek. Listát kérek! Veresgyházd, Pf.: 28. 2112

**C64-es** játékprogramokat cserélek kazettán és lemezen. Keresem SIDEWIEZ és a FREDDY HARDEST nevű játékokat. Válaszokat lehetőleg listával kérem: Vas László, Szeged, Tarján, Pentelei sor 1. IX/45. 6723

**Primo** programsere! Több mint 150 program! Minden gépi kódú programhoz örökéletet is adok, a küldött programokhoz elkészítem. Listát kérek! Répás Sándor, Budapest, Fegyvernek u. 10. VI. 61. 1116

Ez a rovatunk **KODEX 2000** szövegszerkesztővel készült.



# Építsünk számítógépet? ...

*Vita si uli, scias longa est — azaz: az élet hosszú, ha élni tudsz vele — mondta Seneca, s e szemléletnek azóta is vannak követői. Mindenesetre eme örökbecsű állítást bátran zászlajukra tűzhetnék — ha lenne — klubszótárunk következő állomásának szereplői.*

A fővárosban, az V. kerület, Molnár utca 9. szám alatt levő keskeny, négyemeletes épület a tárgyalós Belvárosi Művelődési Ház nevet viseli. Itt lett otthonra kedd délutánoként a Homelab számítástechnikai klub.

Beismerem, enyhe rezignáltsággal készültem az ismerkedésre. Hiszen eddigi klublátogatásaimon úgy tűnt, létezik valamilyen láthatatlan ördögi kör vagy inkább labirintus, melyből képtelen kikászálódni manapság egy-egy hazai számítógépes klub. Mert hiába működik önszerveződő módon, lelkes tagokkal, a „háttérből” valami titokzatos erő a megsemmisítésbe taszítja. Előbb vagy utóbb elsőlétesednek a foglalkozások, és ilyenkor már nem sok érv szól a klubban maradás mellett. Egyeseket a helyszűke, másokat a helyhiány, megint másokat a legcsekélyebb támogatást is nélkülöző szegényes klubmunka dönt az önfelszámolásba. Manapság — hobbi szinten is — lehetetlenség hosszú távon dolgozni megújodás nélkül, ugyanis a programok gépies ismétlődése egy idő után lehangel, megfájtja a kezdeti lelkesedést, végül kikapítja a tagokat. És még hosszan lehetne sorolni... Miért vártam volna mást így ismeretlenül a Homelab klubtól? Náluk miért ne a fentiekhez lenne hasonló a gyakorlat?

Mindenesetre felkészültem a látványra: hordozható tévék, magnók sora, kábeltelevíziók, számítógépek hada fogad. No, meg természetesen a másolandó kazetták elmaradhatatlan halma. A háttérben zsi-bongás, a szűk helyen tolong a tömeg.

Hm. De furcsa! Talán rossz helyre kopogtattam? Netán az emeletet tévesztettem el? Nem, az lehetetlen! Hoppá, megvan! A klub összejövetele bizonyára elmaradt, s most másik baráti társaság foglalja el a termet. De hát mégiscsak érdekes! Ott távol, a sarokban, közvetlenül a beszélgetők mellett miért világít a tévéképernyő? És miért ül oly elmélyülten, számítógépe fölé görnyedve az a férfi? Aztán meg hogy-hogy csak az az egy? A többiek pedig mintha valamiféle állófogadásban lennének. Vagy mondjuk az egyetem folyosóján a vizsgák időszakában a le és fel masírozó drukkoló vizsgázók hada. Itt — lehetnek vagy harmincan — szintén állodogálnak vagy párosával, hármasával sétálnak, közben pedig hatalmas tagejtésekkel magyaráznak. A mozdulatok a beszélő közlődjéit nyomatékosítják. Az asztalok mellett ülő párok, csoportok az is szembre veszem. Csendesnek, elmélyülten látszanak, néhányuknál cserélődnek a beszélgetőpartnerek.

Barabási Rezső — akiről bemutatkozásunk meg tudom, hogy a Homelab klub egyik irányítója, vezetője — hancsalanságot látva, a segítségemre siet. Megerősít,

hogy jó helyen járok. Ajánlja, ülünk le, s mindjárt halkabba is fogja a háttérben szóló diszkózenét. Faggatásomra belekezd a klubjuk történetébe. Ime.

— Születésünket 1983 nyarára tesszük. A többi klubhoz hasonlóan „anyánk” az NJSZT égisze alatt működő HCC klub. A XI. kerületi Bocskay úton lévő TIT szervezésében tartott összejöveteleken ismerkedtünk meg egymással. Ha nem is tudatosan, de az idő tájt éltük a hazai mikroszámítógépek hőskorát, természetesen mi gépépítőként. Az anyaklubból helyhiány miatt elsőként a Sinclair klub költözött el, majd néhány hónappal később mi távoztunk. Kapcsolatok révén az Eötvös Loránd Tudományegyetem, pontosabban az Eötvös Klub fogadott be bennünket, s így ez lett egy időre békés, biztonságos találkozóhelyünk. Olyan, ahol nyugodtan építhettük hétről hétre Homelab típusú személyi számítógépeinket, s adhattuk át egymásnak frissen szerzett számítástechnikai ismereteinket. Erről az időszakról szólván feltétlenül emlékeztenünk kell idézni, hogy noha Nyugaton már terjedőben voltak a ma általánosan használt személyi számítógépek — mint például a Commodore-ok —, de őrök magyar zsebek szinte megfizethetetlenek számítottak. Mit tehetünk mi, akik hallomásból, szakirodalomból, tanulás révén ismerkedtünk meg e számunkra új technikával, ugyanakkor pénztárcánk nem tette lehetővé, hogy hozzá is férjünk? Gépet építettünk. Természetesen ez sem volt filléres szórakozás, de már sokaknak megengedhető kiadás és hasznos kikapcsolódás.

A hazai számítástechnika történetében méltán jegyzett Lukács fivérek dolgozták ki a Homelab személyi számítógépet. Ezt a típust azután nem fejlesztették tovább. Mi viszont alkatrészekből, részegységekből megépítettük magunknak a gépet. Micsoda nagy szó volt! A ZX típusú — számunkra elérhetetlen — gépek „konkurenciájának” foglalkozhattunk magyar számítógéppel! És egy cseppet sem zavart bennünk, hogy ezek gyakran egyáltalán nem voltak kompatibilisek egymással, hogy használható programok sem íródtak hozzájuk.

Rendületlenül raktuk össze a Homelab 2-es, majd a 3-as, 4-es, 5-ös típusokat. Mindezt közös erővel, közös akarattal. Ma már sokaknak felfoghatatlan, hogy akkoriban nekünk ez mekkora örömet jelentett. (A rádiósok hőskora hasonlítható ezzel. — A szerk.)

A klub akkor átlagéletkorá. Az egészen ifjak még nálunk is kevesebbet vagy egyáltalán semmit sem tudtak a számítástechnikáról. Így maradtak az érettségiző koruk és a felnőtt klubtársak. Elméletet akartak tanulni ezeken az összejöveteleken, és gyakorlati ismereteiket bővítették. A tudást, az információt nem sajnáltuk egymástól. Né-

zen csak körbe! Ennek a korszaknak a lenyűgöző mágia eltartott. Az álldogáló, üldögélő „szakik” híreket, adatokat, szakmai forrásokat cseréltek. Sokan egy egész estét nyugodtan rászánnak — ha kell, mert úgy adódik — pusztán egyetlen kérdésre, hogy választ kapjanak: kíváncsiak, hogy a másik, netán este 9-re, de betoppanjon, s az egy héttel korábban felvetett témára meghozza a feleletet.

Visszatérve történetünkre, a hobbikategóriában egészen jó kis gép volt a Homelab. Például 320 × 200 pontos grafikus lehetőséggel működött. Félbemaradt fejlesztéseért kár, hiszen impozáns volt a teljesítménye. Mi mindenesetre sokat tanultunk építése közben. Olyannyira így igaz, hogy sokan lettek közülük az évek során profik, számítógépes szakemberek. Az alapokat itt szerezték, majd egyetem mentek, programozók, mérnökök, vagy éppen rendszer-szervezők lettek.

Időközben átkerültünk a Molnár utcába. És hát értelmét veszítette a gépesítés is, hiszen ma már elérhető áron, csaknem minden harmadik ültetben vásárolhatók különböző mikroszámítógépek. Mi maradt hát az eredeti célkitűzésből, abból, amire szerveztük? A megszerzett tudás és a számítástechnikai szerető baráti társaság. Igaz, sokan kiváltak közülünk: a család, a hétköznapos hajtása, egy új géptípus vonzása stb. távol tart számos egykori klubtagot. De megértjük őket, mert tudjuk, hogy sokuk magas színvonalon műveli a számítástechnikát, meg azt is, hogy szeretettel gondolnak a velünk töltött időszakra. És ez jó. Ha úgy érezték, váltani, módosítani kell, miért ne tettek volna...?

Váltani!? Ezt megmaradt közösségünk is felismerte. Lassacskán mi, a hajdanvolt gépipítő klub, CP/M felhasználói körre alakultunk át. Sőt ezzel párhuzamosan MS DOS felhasználói körre is. A technika fejlődésével a Homelab gép emléktárgyként a vitrinbe került. Nostalgialból azonban nem élhetünk meg! Feltett szándékunk szerint az eredeti névvel ugyan, de egészen más tartalommal kívánunk továbblépni. A világ körülöttünk változik, fejlődik, gazdagodik. Képtelenség lenne egy leirt, kinőtt gép emlékével hosszú távon együtt maradnunk. Programok írását, másolását sem kell már összejönnünk, tudunk annyit, hogy ezzel egyedül, otthon elboldoguljunk. Az együtt gondolkodás viszont kovácsló, megtartó erő a holnapra is.

Megköszöntem, elkészöntem, s távoztamban elhatározom: a ma kinlódó, vegetáló kluboknak tanulságként rögzítem a Homelab klub programját, lépéstartását a változásokkal, a fejlődéssel. Hátha mások olvassa, tanulva, szintén képesek lesznek váltani...

KRASZNAI ÉVA









# SZÁMÍTÁSTECHNIKAI KISSZÖVETKEZET

## VÁLLALJUK

termelőszövetkezetek,  
állami gazdaságok,  
iparvállalatok,  
városgazdálkodási vállalatok,  
tanácsok  
és  
egyéb gazdálkodó szervezetek

ügyvitelszervezési,  
rendszer-szervezési,  
tervezési  
és  
programozási feladatok

## mikroszámítógépekre

(Robotron, Sirius, IBM, Alpha-Micro, VT20 A, VT20/IV, Commodore 64, 610, 720)

## való elvégzését.

## MUNKATÁRSAIKAT BETANÍJTUK! RENDELJE MEG NÁLUNK

- főkönyvi könyvelés,
- analitikus anyag-, fogyóeszköz-, terméknyilvántartási,
- bér- és munkaügyi rendszerek,
- személyzeti nyilvántartás,
- állóeszköz-nyilvántartás,
- tervezés,
- statisztikai feldolgozások,
- egyéb célrendszerek

## szervezését, programozását!

A szervezések bevezetését felügyeljük, a programokra egyéves garanciát adunk. Számítógépek beszerzésében is közreműködünk.

## INGYENES SZAKTANÁCSADÁS!

A számítógépek illesztését megtervezzük, elvégezzük.  
Személyes konzultációt biztosítunk.

**Címünk:** 1027 Budapest II., Mártírok útja 24. VIII. em.

**Telefon:** 151-438

**Telex:** 22-3124



## A Pascal továbbfejlesztése

„Számítástechnikai gondolatainkat” valamilyen programozási nyelven írjuk le. Egyes nyelvek alkalmasabbak bizonyos elközelítések megvalósítására: van például *lebegőpontos aritmetikájuk* — mely a tudományos számításokhoz nélkülözhetetlen —, másoknak meg nincs. A BASIC és a FORTRAN nem támogatja a szövegfeldolgozáshoz és a rendszerprogramozáshoz új szükséges *ún. rekurzív algoritmusokat*. A Pascalban nehéz a nagy programok írása, mert nincsenek eszközei a probléma (a megoldás) jó *szételezéséhez*: nem lehet vele (benne) igazán jól szétdarabolni a feladatokat (a feladatot megoldó programokat).

Természetesen számolnunk kell már a Modula-2 kritikusaival is. Mondják: zsákutca, hiányzik belőle is egy sor fontos képesség, amely például versenytársában, a „mindent tudó” Ada nyelvben megvan. Nincs értelme a döntőbíráskodásnak: a Modula-2 arra biztosan jó, hogy néhány új elvet, megoldást bemutassunk a segítségével. A többit bizzuk az időre.

## Lokális modulok és a láthatóság

Már említettük, hogy a nyelv a *modulok* három típusát különbözteti meg:

- az ún. *programmodulokat* (mondjuk: „fő programokat”)
- az ún. *globális modulokat*, amelyek két részből — egy ún. *definíciós* és egy ún. *implementációs* részből — állnak (ezek segítségével hozzuk létre az „előregyártott szerelvényeket”), és
- a most kicsit részletesebben bemutatandó, ún. *lokális modulokat*.

A modulok — leegyszerűsítve — adatok és procedúrák gyűjteményei. A programmodulok és globális modulok önálló egységek, a lokális modulok mindig be vannak ágyazva egy másik modulba. Szöveggörnyezetükben éppen arról ismerhetjük fel a lokális modulokat, hogy a „MODULE” kulcsszó előtt nem áll a globális moduloktól megkülönböztető „DEFINITION” vagy „IMPLEMENTATION” kulcsszó, valamint hogy a „MODULE” kulcsszó önmagában nem áll önálló egység elején — mert akkor „fő program” (egy „fő program”) kezdetét jelentené. (A továbbiak előtt az 1. ábrán összefoglaltuk a modulokat.)

A lokális modul alkalmazásának fő célja a program *szervezettségének* javítása: a modul belsejében új, a külsőtől elválasztott *környezet* létrehozása. A modulon kívülről *csak* azok az objektumok (változók, procedúrák, típusok stb.) „láthatók”, hozzáférhetők, amelyeket a modul belsejéből „*exportálunk*”, illetve — és talán ez a nagyobb újdonság — a modul belsejéből *csak* azok a külső objek-

tumok „láthatók”, hozzáférhetők, amelyeket a modul belsejében „*importálunk*”. A lokális modul segítségével tehát például a Pascalban megismert „láthatósági” szabályoknál (ezek az ún. blokkstruktúrán alapulnak) finomabban tudjuk szabályozni a „láthatóságot” (elérhetőséget).

Lássunk egy példát (2. ábra). Vegyük észre, hogy a „Példa” nevű modul egy fő program, míg az „Elrejt” nevű „csak” egy lokális modul. A példában elrejtettük a változódeklarációk egy részét. Ha nem ezt tettük volna, akkor a compiler a *q* és az *r* nevű változók kétszeri deklarációja miatt nem is fogadná el a programot. Ha továbbfejlesztett példánkat (3. ábra) feluttatnánk, akkor az alábbi szöveg jelenne meg a képernyőnkön:

```
Ezek itt q,r,s értékei
az Elrejt modulból 10 30 50
Ezek meg itt p, q és r értékei
a fő programból: 2 4 6
```

Új példánkban az „Elrejt” nevű lokális modulból csak két *procedúrát* exportáltunk: a „Kezdőérték” és a „Kiír” nevűt — a modulon belüli (az ott deklarált) változókat nem. A lokális modulon belülről sem importáltuk a fő program (Modula-2 „tájszólásban” az ún. programmodul) három változóját: p-t, q-t és r-t, csak az InOut globális modulból három procedúrát: a WriteString nevűt, a WriteInt nevűt és a WriteLn nevűt. Eredményként a lokális modul belső elrejtett változóihoz csak az exportált két procedúrán keresztül férhet hozzá a fő program, továbbá nincs „névütökzés” a két helyen is deklarált „q” és „r” nevű változók között (lásd a képernyőre kiírt szöveget!).

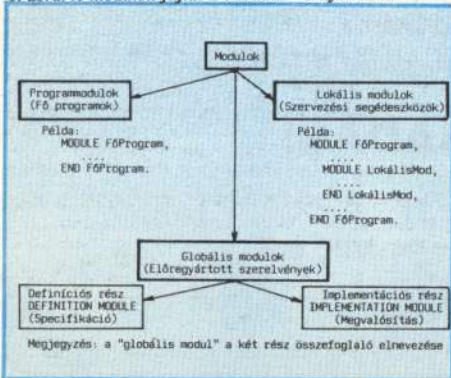
## Általános célú globális modulok

A WriteString, a WriteInt és a WriteLn procedúrákkal kapcsolatban — ezeket mind a fő programba, mind a lokális modulunk-

3. ábra

```
MODULE Újpélda; (*A fő program kezdete*)
FROM InOut IMPORT
    WriteString, WriteInt, WriteLn;
MODULE Elrejt; (*A lokális mod. eleje*)
IMPORT WriteString, WriteInt, WriteLn;
EXPORT Kezdőérték, Kiír;
VAR q, r, s: INTEGER;
PROCEDURE Kezdőérték;
BEGIN
    q := 10; r := 30; s := 50;
END Kezdőérték;
PROCEDURE Kiír;
BEGIN
    WriteString('Ezek itt q,r,s értékei');
    WriteLn, WriteString('Az Elrejt modulból');
    WriteInt(q, 4);
    WriteInt(r, 4);
    WriteInt(s, 4);
    WriteLn;
END Kiír;
END Elrejt; (*A lokális modul vége*)
VAR p, q, r: INTEGER;
(*A fő program "nyilvános" változói*)
BEGIN (*A fő program "nyilvános" teste*)
    p := 2; q := 4; r := 6;
    Kezdőérték; (*A exportált proc. hívása*)
    Kiír; (*Mint az előbb*)
    WriteString('Ezek meg itt p,q és r értékei');
    WriteLn, WriteString('A fő programból');
    WriteInt(p, 4);
    WriteInt(q, 4);
    WriteInt(r, 4);
    WriteLn;
END Újpélda; (*A fő program vége*)
```

1. ábra. A modulok fajtái a Modula-2 nyelvben



2. ábra

```
MODULE Példa;
(*Fő program*)
VAR p,q,r:INTEGER;
MODULE Elrejt;
(*Lokális mod. kezdete*)
VAR q,r,s:INTEGER;
(*Innen, a lokális modul
belsejéből nem "látói"
s külső p,q,r változókat*)
END Elrejt; (*Lokális modul vége*)
BEGIN
(*Itt, a fő program testében
csak a külső p,q,r "látható"*)
END Példa.
```



és az előző számban tett ígéretünknek megfelelően visszatérünk a Modula-2 elnevezésű programozási nyelvben rejlő — valószínűleg sokak számára még újszerű — lehetőségek bemutatásához Kaare Christian „A Guide to Modula-2” című könyve alapján. (Springer Verlag, 1986. 436 oldal.)

ba importáltuk — elmondjuk, hogy minden Modula-2-es rendszer kiegészítésként tartalmaz egy sor igen hasznos globális modult, amely „együtt jár” a fordítóval. E készlet (az összetétel, a funkciók, az elnevezések) kis mértékben függ a konkrét szállítótól, az eladótól (végső soron az adott rendszer fejlesztőjétől). A fenti három procedúra például a Modula-2 nyelvi rendszerrel együtt szállított „InOut” nevű globális modulból való, mely a gép és a külvilág közötti kommunikációhoz szükséges procedúrákat exportálja. A leggyakrabban használt matematikai függvényeket például tipikusan a „MathLibO” nevű globális modul tartalmazza. Ezek a rendszerrel együtt szállított (angolul: common) globális modulok szorosan véve nem a Modula-2 nyelv részei — ellentétben elődjével, a Pascal nyelvvel. A Modula-2 tervezői ugyanis — élve a globális modulokban rejlő lehetőséggel — „kirakták” a rendszer konkrét géptípustól függő részeit globális modulokba. Vásárlás előtt célszerű tehát megnézni, milyen globális modulok „járnak” a rendszerrel.

### Lokális modulok — statikus változók

A Modula-2 konstans és változó adatokat ismer. A konstansok neve olyan érték helyett áll (olyan értéket képvisel), amely a program futása alatt nem változik. A változó értéke pedig futási időben ún. értékadó utasítással változtatható.

A változókat már néztük abból a szempontból, hogy honnan láthatók: láttuk azt is, hogyan befolyásolják a lokális modulok a láthatóságot. Egy másik szempont szerint a változók statikusak vagy dinamikusak aszerint, hogy akkor „keletkeznek-e”, amikor a gép azt a procedúrát kezdi el végrehajtani, melyben a kérdéses változót deklarálták (illetve akkor „szűnnek-e meg létezni”, amikor a procedúrának vége), vagy a változó „él-e” a procedúrán (annak futási idején) kívül is.

A változók élettartama nagyon fontos a procedúrák egymás közötti kommunikációja miatt is. A Modula-2 moduljai a változók

élettartama szempontjából másképpen viselkednek, mint a procedúrák. A modulokon belül változók akkor keletkeznek, amikor a gép a modul környezetének (angolul az ún. *surrounding scope*-nak) végrehajtásába kezd. A lokális modulok környezete ebből a szempontból maga a program (a programmodul vagy a globális modul). Emiatt mondják, hogy a modulok változói — ellentétben a procedúrák dinamikusként mondott változóival — statikusak.

Most egy vázlatos példán (4. ábra) megismerjük érzékeltetni, hogyan lehet a statikus változókat kihasználni. Kezdjük a fenti programban (a kommentárban) tett ígéret beváltásával: a lokális moduloknak is lehet „testük”, azaz a deklarációs rész után BEGIN ... END közé fogva tartalmazhatunk egy utasítássorozatot, amelyet a gép még azelőtt végrehajt, mielőtt a lokális modul körülvevő modul utasításainak végrehajtását elkezdené. Ha több lokális modul van egy modulba ágyazva, akkor ezek testében foglalt utasítások a modulok sorrendjében lesznek végrehajtva (5. ábra). A „StatVált” programmodul „először” nevű, Boolean típusú változója olyan, hogy a program egész futási ideje alatt „él”. Kezdeti értékét még a fő program törzsének futása előtt a „Statikusok” nevű lokális modul teste állítja be. Mivel az „először” nevű változó sem exportáltuk a lokális modulból, az a „statikusok”-on kívül nem látható, nem változtatható. A láthatóság fenti kezelhetősége a lokális modulok egyik legfőbb haszna.

A lokális modulok törzsének végrehajtási sorrendjét több lokális modul esetén az 5. ábra példájával szeretnénk megvilágítani. Ha a programot lefutattuk, az alábbi szöveg jelenik meg a képernyőn:

```
Itt az "A" modul
Itt a "B" modul
Itt a "sorrend"
```

Vegyük észre, hogy a fő program törzsében explicit módon nem hívtuk a lokális modulokat — törzsüket a gép automatikusan, a deklaráció sorrendjében hajtotta végre, még mielőtt a fő program utasításaira ráfutott volna. Ügyeljünk arra is, hogy egy-egy lokális modul ne vegye igénybe a sorrendben utána deklarált másik lokális modul szolgáltatásait.

A lokális modulok törzse kicsit olyan, mint egy igazi (paraméterezett) procedúra, csak hogy törzsüket a gép automatikusan, explicit hívás nélkül hajtja végre a fent bemutatottak szerint.

Befejezésül most is bátorítjuk olvasóinkat, hogy ne riadjanak vissza az első nehézségek láttán, „szoktassák a szemüket-agyukat az újhoz”. Más alkalommal, más könyvek kapcsán igyekezni fogunk még visszatérni a témára.

— KE —

4. ábra

```
MODULE StatVált; (*Fő program *)
MODULE Statikusok; (*Lokális kezdő*)
EXPORT Fn;
VAR
  (*Ide jönnek a "Statikusok" nevű lokális modul
  privat, azaz nem exportált statikus változói *)
  először := BDDLEAN;
PROCEDURE Fn; (*Az exportált proc. *)
VAR
  (*Ide jönnek az "Fn" nevű procedúra dinamikus,
  privat változói *)
BEGIN
  IF először THEN
    először := FALSE;
  ELSE
    ...
  END
END Fn;
BEGIN (*A "Statikusok" nevű lokális modul testének
kezdete. Ez itt most valami új, magyarázat
következik! *)
  először := TRUE;
END Statikusok; (*A "Statikusok" modul teste *)
BEGIN
  ... (*A "StatVált" program teste *)
END StatVált.
```

5. ábra

```
MODULE sorrend;
FROM InOut IMPORT
  WriteString, WriteLn;
MODULE A;
IMPORT WriteString, WriteLn;
BEGIN (*Itt van az "A" modul törzs *)
  WriteString('Itt az "A" modul');
  WriteLn;
END A;
MODULE B;
IMPORT WriteString, WriteLn;
BEGIN
  WriteString('Itt a "B" modul');
  WriteLn;
END B;
BEGIN (*A fő program törzs *)
  WriteString('Itt a "sorrend" ');
  WriteLn;
END sorrend.
```



# Harangozó András

## ÚJFÖLD

### 4. rész Az ellenség



Ren napokig elhanyagolta a Gépet. Most, hogy megválasztotta vezetőnek, azt sem tudta, hol áll a feje. Mégis szakított időt, szakítani kellett, hogy elmenjen a Géphez. Valami belső késztést érzett. Magányossá vált. A gondolatait csak napok múlva értették meg a polgárok. Az úton nem járt senki. Csak ő kereste fel az égből lepotyogt kockát, de már egyre ritkábban.

Ren hallgatott. A kontroll-lámpák fénye furcsa képeket rajzolt a kocka falára. Renben még elevenen éltek a Gép mondatai, melyeket önkívületben sorolt. Az illemszabályok. Ezért hallgatott, várva, hogy a Gép törje meg a csendet.

— Három nap! Három földi nap! — szólalt meg hirtelen a Gép.

Ren összerzezent.

— Parancsolsz?

— A Mester halálának napja lesz.

— Miért kell emlékezni a halottakra? Nem elég emlékezés, ha felhasználjuk mindazt, amit alkottak?

— A nagy emberek képét meg kell őriznünk a lelkünkben, tisztelnünk kell azt, hogy éltek!

— Az emberre emlékezni felesleges, és nem is tudsz! Csak azt idézheted fel magadban, amit alkotott! Nem azzal emlékezünk a legjobban, ha használjuk az általa feltárt tudást?

— Nem — felelte nagyon gyorsan a Gép.

— Miért? — kérdezte Ren.

— Mert a nagy emberek képét meg kell őriznünk a lelkünkben, tisztelnünk kell azt, hogy éltek!

Ren a sétabotjával megbökte a cilinderét. — Neked nincs is lelked! — mondta, és gyorsan, még a Gép válasza előtt kilépett a kockából.

Már sötétedett. Először volt jobb, először jött el úgy a Géptől, hogy ő oktatta ki. Valami mérhetetlen ellenszenvet érzett a Gép iránt abban a pillanatban, de már megbánta.

A csillagok előbújtak a felhők mögül. Ren megállt és bámulta az éjszakai égen világító fénypontokat. Egyre erősebben bánta a kockában előadott produkcióját. Hirtelen elhatározással elindult visszafelé, hogy elsimítsa a dolgot. A kocka közelébe érve, eddig még nem hallott zajra lett figyelmes.

Egy guruló szerkezet érte el a kockát. A tisztító érzés egyre erősebbé vált. Ren minden gondolata menekülés volt. A járműből egy tojás alakú tárgy gurult elő, majd a kocka bejárata felé haladt. Ren nem bírta tovább. Minden erejét összeszedve a város felé iszkolt, vissza sem nézve.

A Gép is érzékelte a közeledő tárgyat. A tojás begurult a kockába. A Gép aktivizálta a belső védelmi készültséget. A tojás beszélni kezdett hozzá Ren nyelvén. Fény gyulladt a belsejében, és a Gép azonosította a benne tartózkodó lényt. A robot által hozott első képeken látott hasonló lényt. Henger alakú törzs, három csáp. Egy víz alatti lény állt előtte, az ellenség!

Ren, amikor beért a városba, egyenesen az újonnan elkerített helyére ment. Testének remegése lassan csillapodott. Megrémült az állapótól. Újból és újból beléhasított a hang, amit a kockánál észlelt. És a félelem, a menekülés. — Miért? — kérdezte magától talán századszor. — Miért?

A Gép minden érzékelőjét az idegenre szegezte. — Idegen — állapította meg, de csak a Mester érezhette volna meg e szó furcsaságát.

A tájban kialudt a fény, és az alak beleveszett a szkafander-szerű védőburok sötétjébe. A Gép megpróbálta elemezni a testet, de a tojásburok minden behatolási kísérletet megghiúsított.

— Én — kezdte az idegen Ren nyelvén; amikor már kinösszá vált számára a csend — az élt civilizációt képviselő... —

— Tudom — vágott a szavába a Gép.

— Én egy élőlény vagyok — próbálkozott az idegen ismét — egy túlélőlény. Ostoba szójáték, tudom, de így van. Egyedül maradtam. A túlélők sorban meghaltak, mint mindig. Háborúztunk logikátlanul, értelmetlenül. Vizióként lebegő célokért küzdötünk, és elbuktunk. Az egész következményeként meghaltunk, azaz meghaltak.

— Mikor a győző megtudja, hogy mit nyert, akkor válik legyőzhetővé!

— Bölcis vagy — mondta az idegen.

— Bölcis? — kérdezte a Gép. — Egy gép? Nem, legfeljebb az, aki megalkotott. Miért jöttél elő?

— Nem akarok meghalni. A Bionomokkal életben tudok ma-



radni legalább addig, amíg valami nyomot nem hagy magam után.

— Miért kell nyomot hagyni? — kérdezte a Gép.

— Te miért vagy? — kérdezett vissza az idegen. — Nem érem, mi okozza a pusztulást. De nem szabad nyomtalanul eltűnni, mintha sohasem léteztünk volna! A Bionomoknál képesek lesznek elkészíteni az utolsó emlékművet, ami azt hirdeti, hogy él-tünk!

— Nem ismered a Bionomokat — vetette közbe a Gép.

— Ugyan, ha akarod, de te nem akarhatsz, elmondom a kapcsolási rajzokat fejből. Remek konstrukció. Életképes, ha használható ez a szó egy robotnál.

— Megölnék.

— Nem — mondta az idegen, és valami bugyborékoló hang szűrődött ki a tojásból. A külső hangszórók közvetítették a Gépnek, ami nevetésként azonosította. — Vigyáztunk — folytatta az idegen —, az első számú parancs, hogy minket nem bánthatnak. A háború alatt átprogramozták őket a mindenáron való védekezésre. De csak akkor állhatnak ellen, ha megtámadnánk őket, egyébként egy furcsa érzékelőjük van, én nem értek ehhez, lényeg az, hogy el kell kerülniük. Félnék tőlem! Tabu vagyok!

— Mindenképpen megölnék. Azt fogják hinni, hogy te vagy az ellenség!

— Hinni? Ugyan! Azt „hiszik”, amit a programjuk előír nekik. Nem, egy gép sohasem vétkes abban, ha pusztít, ha öl, csak a programozó, és mindaz, aki megépítette. Az eszközt elpusztítani oktalanság, hiszen az eszközt mozgatni kell! Azt az értelmet kell megismerni, amilyen létrehozta. De csak fecsegek. Kérdezz, és én válaszolok mindenre!

— A Bionomoknak mi volt a szerepük a háborúban? Hogyan éltek túl?

— Túlélni??? Tovább működnek. Robotok, nem érted? Robotok!

... bemutatni a világegyetem egy másik értelmes lényekből álló társadalmának, hogy milyen volt az ember!

— Egy másik értelmes civilizációnak — áramlott a Gépben.

— Elmondod nekik, hogy robotok? — kérdezte a víz alatti lényt.

— Nem kell elmondanom, ha észrevesznek, akkor ismét tudatosul bennük.

— Veszely! — hasított a Gépbe. A prioritásokat lejjebb szállította. Az információk műholdak folyamatos műsorközvetítéséről lekapcsolódott. Minden másodlagossá vált egy pillanat alatt.

— Menj el! — mondta a Gép.

— De miért? Nem érdekel egy civilizáció? Egy igazi civilizáció? Miért pottyantál ide? Miért? Biztosan van valami, ami miatt megépítették egy ekkora monstrumot! Mi az?

— Menj el! — ismételte a Gép.

— Kezdj el a feladatot! — parancsolta az idegen, és nem sejtette, hogy a feladat lassan a végéhez közeledik.

A Gép elnémult. Az idegen kifelé lebegett a tojás alakú burk védelmében. Bepréselődött a járműbe, ami idáig hozta.

A Gép bemerte a koordinátát, és parancsot küldött a fegyvert hordozó műholdnak, hogy tüzeljen.

— Mi az önvédelem? — tette fel magának a kérdést a Gép. — Benne van a szóban. Önmagamot, a saját érdekeimet védelmezem. Mégis, csak akkor mentenek fel, ha a többség elfogadja a tetteimet. Egyedül vagyok, így jogos önvédelem volt. Tönkre akarta tenni a célt.

A Gép visszaállította az eredeti prioritásrendet. Majd utasította az egyik robotját, hogy takarítsa el a roncsot a kocka elől.

A robot visszatért a víz alá tett információgyűjtő útvjáról. Begördült a kockába, és az újonnan szerzett adatokat áttáplálta a Gépbe. A Gép hozzáfogott az analízisához.

A képen egy már az előzőekben megismert, víz alatt élő lény állt, azaz lebegett, és pálcájával egy képre mutogatott.

— Tehát — mondta a képen lebegő lény — ez egy Bionom. Mint láthatják, felépítése tökéletesen alkalmazsá teszi a szárazföldi bányamunkára. Olyan aggyal láttuk el, ami képessé teszi alapvető döntések végrehajtására. Ez a fajta agy biológiai áramkörökből épül fel, és jelentősen korlátozott. Ezeket a kételkedőnek szeretnék válaszolni: a Bionomok nem jelentenek veszélyt ránk. Szükítettük az agyukat... — Itt a hozott képek elroncsolódtak, és a Gépben csak villogás és kusza vonalak áramlottak. Majd

hirtelen ismét kizsolt, és a Gép folytathatta az elemzést. — ... Hangsúlyozom, gépek! Nem önállóan cselekvő élőlények! Bizonyos szabadsággal rendelkeznek, de csak meghatározott döntéseknél, és itt is csak bizonyos döntési intervallumban. Remélem, megszárazzák ezeknek a konstrukcióknak az alkalmazását, és ezzel a nyersanyagdjainkat nagyon hosszú távra megoldottnak tekinthetjük. Köszönöm a figyelmüket. — A tekercs képei ismét villogás váltotta fel. Mialatt a Gép a következő értékelhető rész után kutatott, feldolgozta az újonnan szerzett információkat, megpróbált rátalálni a Bionomok túlélésének titkára.

A Gép teljes kapacitásával a robot által hozott információs tekercs felé fordult ismét. A kétely ott motoszkált benne, logikátlannak tűnt a Bionomok túlélése. A tekercsen ismét használható részhez ért. A felvételek, amelyeket ez a rész tartalmazott, sokkal jobb állapotban voltak.

A képen egy víz alatti lény állt. Eddig még nem látott borítás fedte a testét.

— A Messzelakók új eszközökhöz nyúltak! — kiabálta. Időről időre robbanások távoli moraja nyomta el a hangját. — Aljas módon próbálják az általuk kezdeményezett igazságtalan háborút a maguk javára fordítani! Csekély erejük nem elegendő arra, hogy nyitlan támadjanak. Tiszteletlenül módon támadásokat intéztek szárazföldi nyersanyagelőhelyeink ellen. De ugyanúgy nem ért minket váratlanul és a fejlemény, mint a többi aljas trükkjük. A Bionomok védelemre programoztuk át. Kiváló mérnökeink segítségével ez sem megvalósíthatatlan. Természetesen ellencsapást mérünk ezeknek a megátalkodottnak a városaira. Mi eddig kerüljük a polgári célpontok elleni támadásokat, de ennek vége! Felszólított mindenkit, hogy az ellentámadások áldozatainak csökkentése érdekében vonuljanak a biztonságot nyújtó, a lakóhelyükhöz legközelebb eső óvóhelyre.

— Polgárháború — állapította meg a Gép. — Egy háború, amiben a Bionomok csak eszközök voltak.

Ren, beérve a városba, furcsa jelenetnek volt tanúja. Két polgár állt egymással szemben. Az egyik egy csillogó kvarckristályt szorított. A másik vágyakozva nézte.

— Add nekem! — hallotta Ren a kérést.

— Mit adsz cserébe? — kérdezte az ásvány tulajdonosa.

— A cilinderemet!

A csere lebonyolódott. Ren bambán állt egy darabig. Azután ismét eszébe jutott a kockánál érzett félelem. Elment a számára kialakított térhez, ami a duplája volt a többiekének, elvégre a vezetőt ez megilleti, és pihenéshez helyezkedett el.

A Gép ezalatt tovább vizsgálta a tekercsot. A háború képei perregtek. Eddig még soha nem látott fegyverekkel pusztították egymást a víz alatt lakók. Hatalmas víztölcséreket gerjesztettek egymás városainak megsemmisítésére. Egy képen egy csónakhoz hasonló eszközt látott a víz felszínén suhanni, amiből meghatározott időközönként ismeretlen hatóanyagú bombákat súlylészettek a vízbe. Kivégzések is voltak szép számmal, árulók és kémekek, gyilkosok és tolvajok. A legszörnyűbbnek azt találta, amikor az elíteltet kihajították a szárazföldre, ahol iszonyú kínok között megfulladt.

— Micsoda történet! — érzékelte a Gép.

Hibára gyanakodott. A külső és belső érzékelő rendszerei senkinek a közeledtét sem jelezték neki.

— Micsoda félreértés! — érzékelte újra a Gép az előbbi „hangot”.

Növelte a kockában elhelyezett műszerek érzékenységét, egészen a kritikus határig, de nem érzékel semmit és senkit.

A kocka egyik sarkában egy fémláda körül zöldes fénnyben kezdett játszani a levegő. A doboz egyik fala eltűnt, és az átlátszó műanyag borítás mögött láthatóvá vált egy emberi agy.

A Gép úgy tudta, hogy ez egy pótenergia-előállítót rejtő láda, és nem szentelt neki nagyobb figyelmet. Belső kameráit ráirányította, és megkérdezte:

— Ki vagy?

— Keden vagyok, a Mester. Azaz az agyam, vagyis a Mester agya. Én vagyok, a Mester.

— A Mester háromban meghalt!

— Igen, de az agyam él. Elkísértelek, úgy, hogy nem tudtál róla. Már két hónapja figyellek! Segíteni akarok. Ketten vagyunk, újra együtt!



# UNIFORM

Követségünk az Apple, Atari, Commodore és Sinclair mikro tulajdonosait feltehetően érdekli, angol és német nyelvű cikkekről információk olvasdinkat egy tartalomeleőről szólólc segítségével.

A fordéshely karakteroszorozatát nyil vezető be, ezt a / jelig 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 másolatkerést megkönyvitendő - a cikk teljes oldalterjedelmét közöljük.

A tartalomeleőről szólólc permutálással és alfabetikus rendezésével szerkesztett teljes anyagot az OMIKK "APACS" c. kiadványosorozata 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 jelzettak az OMIKK (Bp. VIII. Múzeum u. 17.) szakkönyvtárában. Másolatok a SZÁMALK-tól csekkszelvény, az OMIKK-tól megrendelével bekküldésével, vagy személyesen rendelhetők.

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

PROGRAMLISTA  
adatatitel|c128|c64|filecsere a ket  
modell kozott ->cut|'87.01-75/3

PROGRAMLISTA  
adabazis|c64|relativ|fileok|ures al  
loman: eloallitasa|demo  
->run2|'87.01-18/8

PROGRAMLISTA  
asmiga|assembler|macintosh|mikroproce  
sszor|68000|ablakos menu|szubrutino  
k ->dobb|'87.01-48/17

PROGRAMLISTA  
asmiga|grafika|(3-d surfaces)  
->cut|'87.03-90/4

PROGRAMLISTA  
asmiga|jatek|(euchre)|Kartyajatek  
->cut|'87.03-61/4

PROGRAMLISTA  
asmiga|jatek|(jigsaw)  
->cut|'87.02-46/3

PROGRAMLISTA  
animacio|c64|grafika|filmszekvencia|k  
keszites|turbo|grafika|  
->run2|'87.02-96/11

PROGRAMLISTA  
animacio|c64|grafika|trukk|film|keszi  
tes ->164er|'87.02-53/3

PROGRAMLISTA  
apple ii|jatek|(chain reaction)  
->cut|'87.01-51/2

PROGRAMLISTA  
apple ii|jatek|(euchre)|Kartyajatek  
->cut|'87.03-54/5

PROGRAMLISTA  
apple ii|katalogus|formatalas|(40/80)  
->cut|'87.02-43/2

PROGRAMLISTA  
apple ii|lemezkezes|szektor|editor|  
|diskcheck| ->cut|'87.03-108/3

PROGRAMLISTA  
apple ii|nyomtatasi|(printer master)  
->cut|'87.01-67/4

PROGRAMLISTA  
applesoft|levedes|szekvencia|lis szov  
egfille-ok|kodolasa  
->cut|'87.03-97/3

PROGRAMLISTA  
assembler|atari|xl|x|felhasznaloi r  
utinko|hivasa|basic|ol|(page-6 grab-  
bag) ->anti|'87.03-39/4

PROGRAMLISTA  
assembler|forth|(nem szabalyos)|mikro  
prozessor|68000|oktatasi|informatik  
a|a|flint|nyelvi|szerepese|az|mkod  
es|interpreter|az|makro  
->dobb|'87.01-22/13

PROGRAMLISTA  
atari|st|atari|xl|x|barkacolas|bes  
zed|kimenet|trs|chipe|epulo|aramkor  
keszites|(talking|typeur|iter)  
->anti|'87.01-13/6

PROGRAMLISTA  
atari|st|fuzer|hozzarendeles|az|f-bi  
llent|ukhoz ->happ|'87.02-114/2

PROGRAMLISTA  
atari|st|jatek|(st puzler)  
->anti|'87.03-63/3

PROGRAMLISTA  
atari|st|jatek|(super trek)  
->anti|'87.01-08/12

PROGRAMLISTA  
atari|xl|x|jatek|(vectorn)  
->anti|'87.03-16/5

PROGRAMLISTA  
atari|xl|x|jatek|(word searcher)|sz  
okeres|es|betu|acc|bol  
->anti|'87.03-23/4

PROGRAMLISTA  
atari|xl|x|jatek|(zand's|labrynth)  
->happ|'87.03-66/7

PROGRAMLISTA  
atari|xl|x|kepernyokezes|gorgetes  
|(menu|maestro) ->anti|'87.01-63/3

PROGRAMLISTA  
atari|xl|x|kepernyokezes|input fo  
rmatalas ->cut|'87.01-92/2

PROGRAMLISTA  
atari|xl|x|nyomtatasi|ekvezes|karak  
terek|hasznalata ->happ|'87.02-105/2

PROGRAMLISTA  
atari|xl|x|oktatasi|(matematika)|alap  
muveletek|gyakorlata|(math|attack)  
->anti|'87.01-29/8

PROGRAMLISTA  
atari|xl|x|oraber|nyilvantartas|oss  
zesites|(uape|hour|paymaster)  
->anti|'87.03-45/4

PROGRAMLISTA  
barkacolas|billentyuzet|c16|plus|/4  
decimalis|tasztatura|keszites|masod  
tetes ->run2|'87.03-46/4

PROGRAMLISTA  
barkacolas|c64|lemezkezes|masolok  
nagy|sebesség|40 sav|ccop+|parhuza  
mos|kabel|leirasa|a|48.oldal  
->164er|'87.03-56/3

PROGRAMLISTA  
batch|c64|parancs|programoszorok|e  
gy|utasit|as|inditasa  
->chip|'87.03-223/2

PROGRAMLISTA  
c128|c16|lemezkezes|plus|/4|monitor  
|(disk|doc) ->run2|'87.03-108/4

PROGRAMLISTA  
c128|elo|jegyzesi|nap|ter|keszites|(re  
minder|128) ->run|'87.01-48/5

PROGRAMLISTA  
c128|sepi|kod|ml-programok|diszasseb  
lalasa|lemezrol ->cut|'87.03-108/3

PROGRAMLISTA  
c128|grafika|bitter|kep|hasznalat|80-  
csilopos|modusan ->cut|'87.03-69/1

PROGRAMLISTA  
c128|grafika|kepernyokezes|bot|korm  
anyos|menutechnika|(from|paint)  
->run2|'87.01-99/3

PROGRAMLISTA  
c128|grafika|nyomtatasi|hires|rutinko  
az|ms801|802|modellekhez  
->run2|'87.03-113/1

PROGRAMLISTA  
c128|jatek|(roulette|spiel)  
->run2|'87.03-146/3

PROGRAMLISTA  
c128|lemezkezes|at|formazas|mf|m|tip  
us|fel|ir|ashoz ->164er|'87.01-98/2

PROGRAMLISTA  
c128|nyomtatasi|hires|kep|output  
->run|'87.03-54/1

PROGRAMLISTA  
atari|st|kepernyokezes|parbeszed|m  
ezo|beilleszes|(basic|alert)  
->anti|'87.03-68/2

PROGRAMLISTA  
atari|st|kompak|lemez|digitalis|jelf  
eldolgozas|alap|ismeret|aramkor|ter  
vezes ->etor|'87.01-58/2

PROGRAMLISTA  
atari|st|nyomtatasi|vezelokod|kudlo  
rutin|(pr|ntest) ->anti|'87.02-73/4

PROGRAMLISTA  
atari|xl|x|basic|op.rendszer|valtoz  
at|azonositas|modositas  
->cut|'87.03-94/2

PROGRAMLISTA  
atari|xl|x|basic|programozas|10.res  
z|fuzer|kezes ->anti|'87.01-38/6

PROGRAMLISTA  
atari|xl|x|basic|programozas|11.res  
z ->anti|'87.02-55/5

PROGRAMLISTA  
atari|xl|x|billent|uzet|elrendezes  
modositas|(dvorak|keyboard)  
->anti|'87.03-36/5

PROGRAMLISTA  
atari|xl|x|felhasznaloi|karakterek  
generalasa ->cut|'87.03-95/2

PROGRAMLISTA  
atari|xl|x|filekezes|(binary|file  
segment|checker) ->cut|'87.03-89/1

PROGRAMLISTA  
atari|xl|x|filekezes|max. 27 file  
automatikus|betoltes|(multi|autoru  
n) ->anti|'87.03-47/2

PROGRAMLISTA  
atari|xl|x|sepi|kod|ml-programok|at  
irasa|data|sorok  
->cut|'87.01-71/1

PROGRAMLISTA  
atari|xl|x|grafika|kepernyokezes|  
szin|paletta|bovites|graphics-0|modu|b  
an|(rainbow|screen|customizer)  
->anti|'87.03-27/4

PROGRAMLISTA  
atari|xl|x|grafika|lemez|tar|kezes  
->happ|'87.02-107/2

PROGRAMLISTA  
atari|xl|x|grafika|programozasi|tan  
acsok|12.res|z|pr.n|modusok  
->anti|'87.03-42/4

PROGRAMLISTA  
atari|xl|x|hibamegjelenito|ablak|pr  
ogramozashoz|(electric|charlie)  
->anti|'87.02-42/7

PROGRAMLISTA  
atari|xl|x|jatek|(chain reaction)  
->cut|'87.01-54/2

PROGRAMLISTA  
atari|xl|x|jatek|(euchre)|Kartyajatek  
->cut|'87.03-51/4

PROGRAMLISTA  
atari|xl|x|jatek|(quatro)  
->anti|'87.02-10/5

PROGRAMLISTA  
atari|xl|x|jatek|(rebound)|sajkt|ke  
pernyo|tervezese ->anti|'87.02-44/5

PROGRAMLISTA  
atari|xl|x|jatek|(rebound)  
->anti|'87.01-57/6

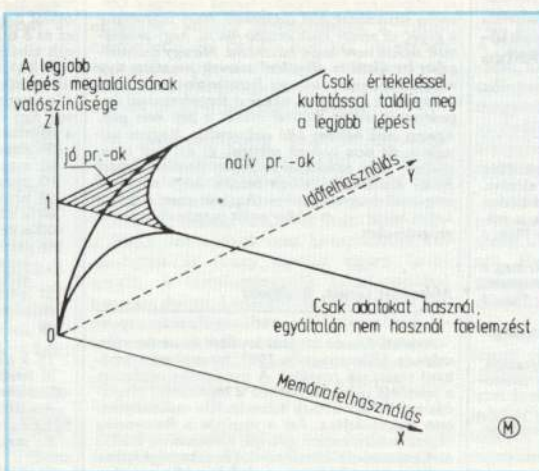


# Király plusz gyalog a király ellen I.

**A játéka terjedelmének minimumra csökkentése nagyon fontos szerepet játszik a sakkprogramok eredményességében. A játéka és kiértékelése című sorozatban (lásd a múlt év szeptemberétől ez év februárjáig terjedő számainkban — A szerk.) több példát láthattunk arra, hogy a világ legjobb sakkprogramozói milyen algoritmusokkal kísérleteznek, hogyan próbálják a fa szélességét lecsökkenteni azért, hogy a megtakarított időt a mélyebb elemzésre fordíthassák. A középjátékban az ismertett heurisztikus módszerek nagyon jól beváltak, de a végjátékban, melynek alapelvei nem egységesek, már nem olyan hatékonyak. Ez az oka, hogy a jelenleg működő sakkprogramok a végjáték a gyengéjűk.**

A végjátékban a középjátékhoz hasonlóan sok esetben lényegesen mélyebben kell számítani a változatokat, és a középjátékkal ellentétben a legkisebb pontatlanság is végzetes lehet a játszma kimenetelére. A középjátékban sok az általános alapelve, amely nagyon megkönnyíti a szelektálást (centrumérték, mozgékonyaság, gyalogszerkezet stb.). Könnyű a helyzetben az is, hogy a középjátékban elkövetett kisebb hiba sok esetben korrigálható. Ezzel szemben minden végjáték típusban más-más elv érvényesül, és ennek a helyes kiválasztása komoly feladat a program és a programozó számára egyaránt. Ezért a kevésbé biztonságos végjátékok megoldását — jelenlegi tudásunk mellett — más módon célszerű értékelni. A kevés báb ellenére szóba sem jöhet a fa teljes kiértékelése.

Nézünk át az egyszerű példát, amikor világosnak királyja és egy gyalogja van az ellenfél királyával szemben (az ilyen típusú végjátékokat KGK-val jelöljük).



## A KGK hadállások

Ezekben a legkevésbé a legálisan megtehető lépés, mégis könnyen beláthatjuk, hogy a játéka végigelemzése egy sakkjátszma kötött ideje alatt lehetetlen. Ezen a majdnem üres táblán a király általában nyolc, a gyalog egy lépést tehet, vagyis az egyik fél összesen kilencet. Ha a lépésszámot alábecsüljük és „csak” nyolc lépéssel számolunk, akkor is óriási számot kapunk a lépésváriációkra. Ha öt lépésnyire, vagyis tíz fellépésnyire szeretnénk előrettekinteni — ami az ilyen végjátékban egyáltalán nem sok — és feltételezzük, hogy minden állásban „csak” nyolc legális lépés van, akkor  $8^{10} \approx 10^9$  nagyságrendű csomópontot kellene értékelni. Ez több mint 12 napi munkájába kerülne egy olyan számítógépnek, amely másodpercenként ezer változatot vizsgál meg. Ez az oka annak, hogy a végjátékban találkozunk leggyakrabban szembe a horizonteffekttel, és ennek súlyos következményei vannak.

## A programozás helyzete

Napjaink legjobb programjai a fa felépítésénél részben

transzpozíciós táblákat használnak, részben pedig adatbankokat, vagy e kettő kombinációját alkalmazzák. A teljes adatbank nagyon sok tárterületet foglal el, ezért működő programokba csak a kulcspozíciókat építik be, ami az adatbanknak elenyésző hányadát jelenti. A kulcspozíciók nem az egész táblára vonatkoznak, hanem csak a figurák egymáshoz viszonyított elhelyezkedésére, vagyis egy táblarészletre. A program a táblarészletet áttranszponált táblákkal, transzpozíciós táblázat használatával eredményesen alkalmazza, bárhol is helyezkedik el ugyanaz a figuracsoport a kulcspozícióban megadott elrendezésben.

A különböző programok a rendelkezésre álló tárkapacitástól függően különböző számú kulcspozíciót tartalmaznak, amit ha kombinálnak a kutatásra szánt idővel, a horizontathasználat más-más mélységben jelentkezik. Az ábrán olyan háromdimenziós függvényt láthatunk, amely megközelítőleg jól szemlélteti az adattárolás és időfelhasználás függvényében a korrekt lépés megtételének valószínűségét. Ha a teljes adatbankból veszi a

program a lépést, akkor biztosan a legjobbat teszi meg, de nagy tárkapacitásra van szükség. Ha a játékat elemzi végig, akkor szintén eljut a legjobb lépésig, de az időfelhasználás lesz rengeteg. A legjobb lépés megtalálásához e kettőt — a tárolást és az időfelhasználást — kombinálhatjuk. Ezek egymáshoz való viszonyát szemlélteti a legjobb lépés síkjában ( $z=1$ ) levő görbe. A függvény azt is megmutatja, hogy más idő- és tárkapacitás-összetevőkkel a program milyen valószínűséggel találja meg a legjobb lépést.

## Nagy nevek

A programozók már régóta foglalkoznak a kevésbé biztonságos végjátékok hatékony beprogramozásával. Némelyek a teljes adatbankot is elkészítették (lásd az 1987. júliusi számban), de a nagy tárkapacitás hiányában ez a sakkprogramokban nem használható. A KGK típusú végjátékok feldolgozásával 1972-ben Tan próbálkozott, majd hosszú szünet után 1977-ben Beal, Bramer, Harris, Piasetski és Clarke. Ez utóbbi volt, aki elkészítette a teljes adatbankot. Programja úgy működött, hogy visszafelé dolgozta fel az állásokat. Vagyis megnézte, melyek azok a pozíciók, amelyek biztosan nyernek, és ezekből visszafelé építette fel a fát. Programjának két hétre (!) volt szüksége ahhoz, hogy az összes változatot feldolgozza.

A következőkben megvizsgálunk egy olyan algoritmust, amely sakkprogramokba beépíthető, és a KGK hadállásokról korrekt módon megállapítja, hogy nyer-e a gyalogal rendelkező fél vagy döntetlen a pozíció. A módszert elsőként Beal használta. Az algoritmus nem alkalmaz teljes katalógust, mégis, amikor összehasonlították a Clarke által készített adatbankkal, amely az összes KGK hadállást tartalmazza, korrektnak bizonyult.

(Folytatjuk)  
KOVÁCS P. ATTILA



**Egy kissé túl jól sikerült a biztatás. Ismét nagyon sok levelet kaptam — persze ezt nem panaszként, inkább dicselkedve mondom. Néhányat, amelyek közérdeklődésre tarthatnak számtól, rövidítve bemutatok.**

**Weiser István, Szeged**

a) Plus/4 számítógép-tulajdonos vagyok. Elég sok gondot okoz egyes játékoknál a cél elérése. A lapot az 1987/2. számtól jarram. Arra kérem önöket, hogy küldjék el nekem azoknak a számoknak a listáját, amelyekben C16 vagy Plus/4 örökletek, programleírások vannak.

b) Azt is szeretném kérni, hogy jelenjen meg a számban Plus/4-re vagy C16-ra több program, ugyanis mostanában eltűntek a lapból a Plus/4 programok.

c) Utókérek: kérem, írják meg nekem a szegedi programcsere-partnerek címét.

d) A *µM-1* nagyon sok könyvtárral megtalálja. Azt javaslom, hogy nézze át a régebbi számokat. Érdekes lesz, meglátja.

e) Nem csak rajtuk múlik!

f) Szegedi Plus/4 programcsere-partnerek, jelentkezzenek!!!

**Kaufmann Zsolt, Ajka**

Zircen tanulok könnyűszerkezet-lakatos szakmát. Tetszik a szakma, de én mégis tovább szeretnék tanulni a számítástechnika irányában. Azt szeretném kérdezni, milyen képesítés kell egy operátornak vagy egy programozónak, és miből áll ezek munkaköre. Kell-e nyelvet tanulni?

Azt hiszem, hogy egyetlen dolgot nem lehet megbánni soha: azt, ha az ember tanul. Még abban is lehet reménykedni, hogy a társadalom ezt a nagybőrdűsödést egyéretként őríti meg. A mai iskolarendszerben csak úgy tud tovább tanulni, ha leértékeljük — vagy gimnáziumban vagy szakközpontokban — és persze nem is akaróhogy, hanem jó eredménnyel. Érettségi után, ha jól tanul, elég széles a skála. Elmelet a Kandóba Pesten vagy Székesfehérváron, a Bánkiba vagy más főiskolákra. Beiratkozhat a SZÁMALK valamelyik tanfolyamára, ott is szerezhet felsőfokú képesítést. Lehetősége van azután egyetemre is menni Miskolcra, Szegedre, Pécsre, Veszprémbe, és persze Budapestre is.

Általában: hogy kell-e nyelvet tanulni, ez ma már nem kérdés, különösen meg az informatikában. Úgy vélem, hogy nyelvtudás nélkül ma ebben a szakmában csak félelmeleként lehet működni.

**Dr. Kisapáti Edéné, Szeged**

Vásároltunk egy, a kereskedelemben kapható Seltron 200 Color Computert, de játéprogramkártya nem kapható hozzá. Amerre csak érdeklődünk, mindenütt azt halljuk: az nincs, ez a gép nincs elterjedve. A számítástechnikai szakületben is próbálkoztunk itt Szegeden, hátha tudnánk kapni magyar nyelvű leírást, mert ami van hozzá, nem kielégítő, de itt is csak "nincs" a válasz. Ezt nagyon fűrsállom, hiszen az üzletekben most is kapható a gép, és a Skála-Coop forgalmazza. Nagyon kérem, legyen a segítségünkre. Írja meg, hogy hol érdeklődjünk. Biztosan megérti, hogy egy gyerek számára a számítógép a játékkal kezdődik.

Műi is mondják, hiszen majdnem minden számban kell hasonló tartalmú levélre válaszolnom. Nem beszéltem a Skála-Coop-pal, de azt hiszem, nem tévedek, ha feltételezem, hogy itt arról van szó: a kereskedelem fel tudott vásárolni korláto-

zott számban ilyen, általam sem ismert számítógépeket, amelyekhez azután mindenféle szoftvertámogatás nélkül „olcsón vette, viszonylag olcsón is adja” alapon hoz forgalomba. Sajnos ilyenkor a képzetlen vásárlónak kell eldöntenie, hogy megveszi-e a gépet, és persze csak később jön rá, hogy programok nélkül nem tudja használni. Ma egy számítógépet az illetékes ellenőrző szervek pontosan úgy vizsgálják meg, mint egy hajszárítót, és ha nem úgy ismerten a használatát, akkor a forgalmazását engedélyezik. Pedig szoftver nélkül a gép nem gép, hanem csak néhány kiló elektronika. Nagyon sajnálom, de nem tudunk segíteni, és azt sem tanácsolom, hogy panasszal forduljon bárkihez; egyszerűen elutasítják. Abban bízom, hogy a gyerekek megtanul programozni — ha lesz ezen a gépen kedve hozzá —, és akkor saját magának ír játéprogramokat.

**Mészáros Gyula, Budapest**

Avarosi László olvasói levelére és az ön válaszára (a Mikromagazin 1987. novemberi számában) szeretnék reagálni. A nyomtatási problémára a jelenlegi piaci helyzetben a legolcsóbb megoldás egy RPR—210-es Kármán-féle mátrixnyomtató megvásárlása. Ezt a masinát a Rozmaring Mgtst műszertüzeme gyártja, Centronics, RS232 és Commodore interfészrel. (Érdekességképpen említem meg, hogy ez az első és eddig egyetlen nyomtató, amely teljes egészében magyar műszaki alkotás, a fejtlő kezdve a vezérgyűjteményig.) A kész nyomtató ára 19 900 forint, de kitiben (bemérés előtti állapotában) már 9900 forintért megvásárolható. A félkész géphez jár egy bemérési útmutató, amelynek alapján egy oszcilloszkóppal tökéletesen elvégezhető a beállítás. Ha valaki mégsem boldogul vele — vagy bele sem mer kezdeni —, az Angström Gmk, melynek tagja vagyok, 1000 forintért elvégzi a munkát.

Szöveges üzemmódban a nyomtató 80 normál vagy 40 duplaszélű karaktert tud elhelyezni a 210 mm széles papíron egy, illetve két másodperc alatt. A normál grafika felbontása 512 pont, a nagy felbontásúé 1024 pont. Be lehet állítani duplaszélűs, aláhúzás és kétszeres nyomtatási üzemmódot, továbbá ezek kombinációját is. A gép 96 karaktert ismer egyezsere, ez átkapcsolhatóan ASCII vagy magyar ékezetes (ha jól tudom, Primo-kompatibilis).

Jelenleg az 5.3-as verzió üzemel a munkahelelyemen lévő Enterprise 128-hoz kapcsolt gépben. A karakterkészlet 256 karakterre bővült, amelyből tetszőleges 32 újradefiniálható, illetve tetszőleges 128 karaktert egy mássikkal helyettesíthető. Jelenleg az Enterprise 128-as és az IBM-kompatibilis karakterkészlet áll rendelkezésre, mindkettőt a teljes magyar ékezetes betűkészlettel bővíti.

A vezérgyűjteményhez szintén az Angström Gmk-n keresztül lehet hozzájutni: ára EPROM-mal és beszereléssel 900 forint.

Ez valójában reklám, de azt hiszem, hogy sokak gondján segít. Ha a hazai piac tele lenne nyomtatóval, akkor biztosan nem kerülhetett volna a rovatba.

**Borsos József, Székesfehérvár**

Nagy érdeklődéssel vettem kézbe a szeptemberi számot az Enterprise 128 ismertetővel, és kissé csalódottan tettem le még a novemberit is. Sajnos, nem tudtam még lényeges dolgokat a gépről, mint ahogy lényeges dolgokat „kifejtették” az Enterprise-nak a géphez mellékelte kézikönyvből is.

Elkeserítőnek tartom, hogy az információk visszatartásával lehet manipulálni. Hónapok óta izgattak a technikai ismertetőt a géphez, de mindig közbejött valami. Ugyanakkor fülbé sügva

metudhatom, hogy persze egy-két példány azért nem felárlár már kapható.

Romvári Gábor és Kopácsy Vilmos egyaránt magától értetődően irnak az LPT és LPB funkciójáról a gépben. Hogy ezt eszik, isszák, sikálják, ez az ő titkuk. A most következőket viszont sikerült kibányásznom a gép mélyéből.

A képernyő sorainak állapota a 47872 tárciműtől kezdődően van a memóriában, a Status, azaz állapotjelző sorral kezdődően, tehát összesen 25 sor. Egy-egy sor jellemzőit 16 bájrtartja nyilván a következő szerkezettel:

- 00 nem tudom
- 01 a sor video display üzemmódja
- 02 nem tudom
- 03 nem tudom
- 04 a képernyőcím alsó bájrtérteke
- 05 a képernyőcím felső bájrtérteke
- 06 attributum módban a grafikus memória, karakter üzemmódban a beírt karakter címének alsó bájrtérteke
- 07 a felső bájrt ujanarra
- 08 a paletta 0. színe
- 09

15 a paletta 7. színe  
A kezdőcímben lehet eltérés, ezért a gépből való kiolvasást is közlöm:  
A = SPEEK(255,49140)+  
SPEEK(255,49141)\*256  
Ez megadja az állapotjelző sor adatainak tárcimét.

A másik téma, ami miatt írok, a novemberi számban dr. Nagy András cikke: A Spectrum másolói. Szeretnék néhány észrevételt tenni a leírásokkal kapcsolatban; néhány funkciót nem egészen a leírás szerint működik. Estelge, ha ez Nagy Andrásnak is megfelel, szeretném velem megbeszélni és a javításokat így legközelebb.

Az észrevételeim a következők. Fenntartva természetesen az ízelek és pofonok különbözőségéről tartott elvet, szeretném elérni a COPIER FM3 másoló szép csendes kihálását. Tapasztalatom szerint ugyanis ez egy „diverzans”. Többször használva elrontja a programokat, mindig hozzacsempeszve egy adatot egy bájrt puszt, például a fejleche. Ez egyedi kiírásának DEC vagy HEX alakját az SS+4 kapcsolóval lehet váltani, a hangot pedig a CS+1-gyel.

A „markbajt”, az akkumulátor tartalma, a jelen 0, a programórzésl péld 255, az adatállomány legelső bájrtja. Szokások a programokat úgy védeni, hogy az akkumulátor nem 255, így az adatállomány csak saját betöltéssel tölthető be.

Az OMNICOPIY saját töltőrutinál rendelkezik, így a kimentésekhez az alap 1500 baud helyett az 1600 baud sebesség kell. Ez az egyik oka a gyorsaságnak, a másik pedig, hogy az adatállományok közötti időzítések hiányoznak.

A P gomb feladata (LEADER PITCH) nem titkosítás. Egyezsere normál vagy turbó üzemban a fej könyvből megtalálása a cél, vagyis fokozottan figyelni a 0 akkumulátort.

A betöltés folytatása a K billentyű funkciója. Előtte a még megtartandó állományra kell vinni a kurzort!

Úgy hallottuk, hogy a reklámait technikai ismertetőt, mire ez a cikk megjelenik, a Centrum Aruházak és a Novotrade már javában árusítja.

Néhány reklámoló levélre még nem válaszoltam, mert késik a válasz az illetékesektől; ezekre később visszatérünk. Várom leveleiket.

KOVÁCS GYÓZÓ

**Minden hétfőn 17-től 19 óráig**

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



## Egy kép többet mond, mint ezer szó...

**Ez régi szólás. Lehet, hogy többen megtették már, de mi is nézzünk utána egy kicsit: a számítógépes grafika adja az aprópót.**

Ha feltételezzük, hogy egy szó átlag tíz betűből áll, akkor ezer szó 10 ezer karakter. Ezer szót tehát a számítógépben tízezer bajton tárolunk, ami 80 ezer bitet jelent. Ugyanennyi helyen a számítógépes grafika például 400 x 200 pontos monokrom felbontás adatait képes őrizni. Kétszáz soros, soronként 400 pontos felbontás a számítógépes grafikában nem kis teljesítmény, de a kívülállóak mégis bárdolatlanul leszólják ezt a fokozatot: a leképezés látható korlátaiból adódó töredezett „köröket” és főleg az „egyeneseket” hamar sejtnek minősítik. Ha finomabb felbontást akarunk elérni, akkor hatványozottan nőnek a méretek: a kép digitálizált adatainak tárolásához nem kevés hely szükséges. A mai profi készülékek 2048 x 2048 pontos felbontása monokrom esetben is több, mint négy millió bitnyi adat tárolását igényli. Ha a színt és a fényességet is hozzávesszük az információhoz, akkor még ennek is a sokszorosára szükséges.

Könnyen belátható, hogy az „igazi” képekben óriási információmennyiség található. Messze több, mint amennyit ezer szó elmondhat. A vajtűfülkék azonban tudják, hogy a hobbi számítógépek a legjobb felbontásukban is 10–16 kbájtnyi tárral szoktak dolgozni a képgeneráláshoz. Ez közel van az ezer szó tárolására alkalmas tárméretre. Hogyan lehetséges, hogy ezeket a „durva” felbontású képeket nézve is az a benyomásunk, hogy ezer szóból kevesebbet tanulnánk, mint amit ezek a képek egyetlen pillantásra elének tárnak? Lehet, hogy az emberi képzetet, kiigazítva a képhibákat, megsokszorozza a képpen tárolt eredeti információt?

A megfigyelések szerint a tanulási folyamat hatékonyságát növeli, ha a rögzítés során minél bonyolultabb emberi tevékenység végződik. Mindenki tapasztalta, hogy könnyebben megjegyzi azt, amit valahova felír, mint ha ugyanazt csak hallja. A szöveg és a kép hatása közötti eltérés oka ezért nemcsak a tárolt információmennyiségben keresendő. Valószínű, hogy a kép áttekinthetése, elemzése jóval bonyolultabb gondolkodási műveleteket igényel, mint — automatiussága révén — az olvasás. Ráadásul a szöveg unalmas is lehet!

A címben idézett szólás akkor jutott eszünkbe, amikor megnézhattuk a Tudományszervezési és Informatikai Intézetben az egyik programcsomagot, amelynek témája az ábrázoló geometria, összefoglaló adatait pedig a táblázat tartalmazza.

A programcsomag kezelése egyszerű. A kassetta tartalomjegyzékkel kezdődik. A megadott számlálóértékeket persze nem tudtuk használni, mert a magnónk nem MK25-ös, hanem MK27-es volt. Kár, hogy a BRG önmagával sem kompatibilis két modellen át (vagy esetleg egyen belül sem?). A tartalomjegyzék mindazonáltal hasznos dolog. Legalább helyrehozza a TII programkatalógusában található rövid leírás sorrendhibáit, amelyek a dokumentáció elővasása után nem tudni, hogyan kerültek, hiszen abban jó a sorrend (apró malőr).

A szerző szándéka az volt, hogy játszva tanítson, amit a fenti gondolatok alapján is csak üdvözölni lehet. A játék bekapcsolása az oktatási folyamatba szintén bővíti a tevékenységőrt, ezért potenciálisan növeli a hatékonyságot — ha a játék nem tereli el egészen a figyelmet a tárgyról. A programok megtekintése arról győzött meg, hogy a szerző elérte a célját. A játékoság mellett sokkal fontosabb, hogy kivételesen jól sikerült bemutatnia a tantárgy szépségét. A képszerkesztés technikáját igen ügyes, animált ábrákkal szemlélteti.

A dokumentáció nem túl bő, de — ahogy az összefoglalóban szerepel — megfelelő. Bár a programok megadják a kezelésükhöz szükséges tudnivalókat — ahogy ez interaktív programokhoz illik —, jobb, ha a dokumentáció kéznél van a munkához. Az online help ugyanis menet közben nem hívható elő, továbbá a tantárgy-pedagógiai információk is fontosak a megértéshez. Különösen, ha valaki még ismeretlenül vág neki az ábrázoló geometriának. Ilyenkor valamilyen tankönyvről is szükséges van a dokumentáción kívül. Ez utóbbi megjegyzést annak kapcsán tesszük, hogy egyre több törekvést látunk az autodidakta oktatás támogatására. A programcsomag a mellékelt dokumentációval csak részben illik bele ebbe a tájképbe.

A programcsomag a hatoldalú hasáb ve-

tületeivel kezdi a témát. Érdekessége, hogy van külön oktató és feleltető funkciója. A folytatás a hatoldalú hasáb ferde síkmetésze, ahol a szerkesztési lépések között tehetőleges szünet tartható. Ezalatt a tanár magyarázhat, vagy a tanulók kérdezhetnek, az önálló tanuló pedig utánanézhethet a tankönyvben az elméleti alapoknak. A program nagyon látványos.

A látványosság folytatódik a következő programban is, amely a gúla ferde síkmetésze után szép axonometrikus képet rajzol. Hasonlóan szemléletes a következő programban szereplő háromoldalú hasáb csonkítása és izometrikus axonometriája.

Most jöhet egy kis kikapcsolódás: a befejező rész, amelyben csonkított kockával játszhatunk. A szerző a játéknak térszemléletet javító szerepet szánt. Sajnos a kezelés módját először félreértettük, majd a számítógépes játékokban megszokott botkormányhasználatához viszonyítva furcsállottuk. Ugyanez áll az utána következő játéka is, amelyben egy csonkított hasábon kell pontok helyét felismerni. A „Hamis? — Igaz?” játékokban pontkoordináta-párokra kell etalálni. Az utolsó két program a téglatest és a hatoldalú hasáb kapcsán kinez meg bennünket: ezek az egyszerű testek is komoly gondokat okozhatnak nekünk, ha szózatlan vetületben látjuk őket, és főleg, ha gyorsan kell válaszolnunk a velük kapcsolatos kérdésekre.

### A programcsomag értékelése

<b>Kezelhetőség:</b>	jó
<b>Teljesség:</b>	jó
<b>Dokumentáltság:</b>	közepes
<b>Használhatóság:</b>	kiváló
<b>Ár/teljesítmény:</b>	közepes
<b>Összbenyomás:</b>	jó

Még néhány megjegyzés a programcsomaghoz. Elgondolkodtat, milyen komoly erőfeszítést jelenthetett a szerzőnek a kidolgozáshoz szükséges szoftvertechnológiai eszközár elkészítése. Bár erről a dokumentáció semmit sem mond, valószínű, hogy a VIC BASIC beégett lehetőségein túl más nem állt a rendelkezésére. A programcsomag ugyan elég sokat lefed az ábrázoló geometriából, de korántsem mindent. A teljes lefedés nem is várható el egyszemélyes munkától. Ezért a programcsomag mögött keresünk még egy technológiai eszköztárat, amellyel hasonló programcsomagokat lehetne írni. Ez legalább olyan hasznos programtermék volna, mint az ÁBRÁZOLÁS. Csak ahhoz jobb dokumentáció kellene!

ZSADÁNYI PÁL—IF. ZSADÁNYI PÁL

<b>Forgalmazó:</b>	Tudományszervezési és Informatikai Intézet
<b>Terméknév:</b>	ÁBRÁZOLÁS programcsomag
<b>Szerző:</b>	Papp László
<b>Géptípus:</b>	Videoton (TVC)
<b>Hordozó:</b>	kassetta (9 programmal)
<b>Dokumentáció:</b>	gyenge kivitelű, de a célnak megfelelő
<b>Ár:</b>	1250 forint (1987-es ár)



**Shirai, Yoshiaki — Tsujii, Jun-ichi:**  
**Mesterséges intelligencia**  
(Budapest, 1987.  
Novotrade,  
179 oldal. Ára: 349,— Ft)

A számítógépes adatfeldolgozási technikák fejlődése együtt járt a korábban csak ember által végezhető szellemi munka automatizálásával. A mesterséges intelligencia kutatásának tárgya az előre meghatározott megoldási szekvenciával nem rendelkező szellemi feladatok megvalósítási módszereinek tisztázása. A mesterséges intelligencia tárgyköre így nem rögzített, hanem időről időre változik. Úgy tűnik, az a „végzet”, hogy egy adott területet a rá vonatkozó módszerek megállapítása, illetve gyakorlati megvalósítása után megszűnik a mesterséges intelligencia része lenni.

Általánosságban két fő nézőpontból szemlélhető. Az első a tudományos megközelítés; célja az emberi intelligencia működésének tisztázása. Ez a számítógépet szimulációra használja az intelligenciára vonatkozó tételek érvényesítéséhez. A másik technikai, amely a számítógépet az ember intellektuális képességeivel kísérli meg felruházni. A kutatások nagyobbik része az utóbbi nézőpontot követve, anélkül próbálja az emberi intelligencia szintjének elérését a számítógépekkel, hogy az ember információfeldolgozási lépéseinek pontos rekonstrukciójára törekedjen.

A szerzők bemutatják a mesterséges intelligencia egyes részterületeit és az ezeket támogató alapvető eljárásokat. Megismertetnek alkalmazási területeinek legfontosabb fogalmaival és eljárásaival. Áttekintik azokat az elveket, amelyekre a gyakorlati megvalósítások épülnek.

**Babán Gábor — Masa István:**  
**Gépi kódú programozás**  
**kezdőknek és haladóknak**  
(C16 és PLUS/4 számítógépekre)  
(Budapest, 1987. Novotrade Rt.,  
211 oldal. Ára: 129,— Ft)

A könyv két nagy fejezetből és függelékéből áll. Az első fejezet a kezdőket igyekszik bevezetni a gépi kódú programozás rejtelmeibe. A szerzők, amint az előszóban írják, ezt a részt tankönyvnek szánják. Az olvasóban eszerint kialakuló elvárással ellentétben nem könnyű olvasmány ez a bevezetés: én, mint nem teljesen kezdő is nehezen rágtam át magam rajta. Pedig szinte minden fontos tudnivaló benne van. A szerzőpáros részletesen ismerteti az alapfogalmakat, az utasításokat, címzési módokat, a beépített monitornak, valamint egy assembler fordítóprogramnak, az ASS-16-nak a használatát. Az utóbbiról azonban a legfontosabb információival adósak maradnak: hol, hogyan lehet hozzájutni?

A második fejezet a haladóknak szól. A gép lelkivilágának minden részletéről írnak, egy-egy rövid programmal illusztrálva a bemutatott területen alkalmazható programozási fogásokat. Sok részletet táblázat segíti az eligazodást, bár néhány olyat is találtam, amelyet feleslegesnek tartok.

A függelékben további nyolc táblázatot találunk; ezek egy része más könyvekben is fellelhető, de itt is helyükön vannak. Szükség lenne egy irodalomjegyzékre is, a kiegészítő olvasmányok felsorolásával. Ennek hiánya azért is furcsa, mert néhány Novotrade kiadvány is feltétlenül szerepelhet benne.

A felfedezett sajtóhibák és tárgyi tévedések arra engednek következtetni, hogy akad még jó néhány belsőük, amelyek rejte maradtak. Ennek

ellenére a téma iránt érdeklődőknek ajánlom a könyvet, emlékeztetve őket a régi szabályra: ha a terep és a terep nem egyezik meg, akkor mindig a terephez kell igazodni.

**Tv-informatika**  
**Szerkeszti: Kovács Péter**  
(Budapest, 1987.  
**SZÁMALK,**  
297 oldal. Ára: 144,— Ft.)

A kötet a Magyar Televízió és a SZÁMALK együttműködésének második termése. Elődjétől, a nagy sikerű TV-BASIC-től két dologban tér el: egyrészt nem tekinthető szorosan véve tankönyvnek, másrészt nincs olyan szoros kapcsolatban a képernyőn megjelenő műsorral.

Mi is a tv-informatika? Egy szakma, a rendszerszervezői, elsajátításának első lépése. Amíg azonban a TV-BASIC-et végigtanulók olyan ismeretekhez jutottak, hogy megkezdhették a programozói gyakorlatot, a tv-informatika csak az előfeltételek előfeltételeinek megteremtéséhez vezet út. A rendszerszervezőnek jól el kell igazodnia azon a területen, amelyikre a szervezés irányul, és tudnia kell alkalmazni a korszerű szervezés- és irodatechnikai eszközöket, módszereket a tárgyrendszer jobb minőségű megvalósításához.

A kötet önálló tanulmányokban foglalkozik a rendszerszervezők helyével és szerepével a számítógépes információrendszerek fejlesztésében, az irodaszervezéssel és ügyvitelgépítéssel, a kereskedelem — például a vonalkód — szervezési problémáival, az iskolai informatikával, a bankszervezet információrendszerével.

BARNA LÁSZLÓ

## A COBRA alapos munkát végez...

Tudják ezt szövetkezeti ügyfelei, hiszen segítünk gondjaik megoldásában.

### De tudja-e Ön is,

hogy szövetkezetünk komplex számítástechnikai szolgáltatást kínál a számítástechnika minden területén?

#### Egy kis ízelítő:

- hardver elemek az Ön igénye szerint (IBM PC XT/AT) számítógépek: TURBO AZTECH)
- nagy hatékonyságot biztosító hálózati rendszerek kiépítése, telepítése (NOVELL)

- floppy lemezek
- egyéb speciális kiegészítők (EPROM-égető, digitális kártyamérő, hardverbővítők stb.)
- a megrendelő sajátos követelményeit kielégítő programok

**Munkánk megbízhatóságát  
nagyvállalati  
referenciák igazolják.**



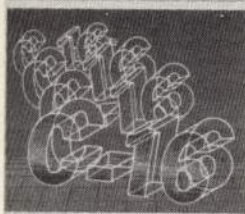
**AHOL A GOND GONDOLATTÁ VÁLIK**



**Hettinger, A. — Heinz, A.:**  
**Atari—BASIC kezdőknek**  
 (Budapest, 1987.  
 Műszaki Könyvkiadó,  
 129 oldal. Ára: 60,— Ft)

Az Atari személyi számítógép igen kedvező tulajdonságokkal rendelkezik. Finom felbontású grafika, többesatomás hangszintetizátor és a tanulásra igen alkalmas BASIC jellemzi. Magyarországon 1986-tól lehet kapni. A vásárlók bizonyára nemcsak a gyári szoftvereket szeretnék használni, hanem meg akarnak ismerkedni a gép speciális tulajdonságaival, lehetőségeivel is. Ez a kötet a gép bekapcsolásától kezdődően az otthoni alkalmazásokig vezeti el a felhasználót. Számos példát, trükköt és játékprogramot is tartalmaz.

THEISZ GYÖRGY  
**BASIC**  
 TANÁCSADÓ  
 C-16-PLUS/4



**Theisz György:**  
**BASIC tanácsadó. C—16, PLUS/4**  
 (Budapest, 1987.  
 Műszaki Könyvkiadó,  
 129 oldal. Ára: 49,— Ft)

A közismert Commodore 64-gyel összehasonlítva a Commodore 16, illetve a Plus/4 egy sor kellemes, új tulajdonsággal rendelkezik, és így programozását sokkal könnyebb elsajátítani, mint a C64-ét.

Nagy felbontású grafikája és a grafikus utasítások igen kreatív felhasználást biztosítanak. A könyvet olyan szerző írta, aki jól ismeri a gép minden különleges tulajdonságát, és tanárként tudja, mi a legmegfelelőbb megközelítés a diákok, a gyerekek számára.

A kötet külön fejezetben tárgyalja a billentyűzetet, a képernyőt, a kulcsszavak típusait, a programozás jellemzőit. Leírja, hogy mit érdekes a BASIC-ről tudni. Végül néhány BASIC-trükköt és mintaprogramot is közöl a feladatmegoldásokkal együtt.

Minden kedden 17-től 20 óráig  
**ENTERPRISE klub**  
 a VSZM Közösségi Házban  
 (Bp. XI., Fehérvári ut 120.)  
 Klubvezető: Romvári Gábor  
 Telefon: 450-950/473

## A karatéval kezdődik

Budapesten, a Testnevelési Főiskolán gyors fordulatú filmfelvétel készítenek filmeket a karatézokról. Ezeket egy NAC típusú félautomata koordinátalevelő és mozgáselemző segítségével értékeli ki, mely számsor formájában adja meg a számítógépes elemzéshez szükséges adatokat. Feldolgozásukra egyelőre C64-es programok futnak, de már folynak az előkészületek a munkák IBM PC-vel kompatibilis gépre való átvitelére.

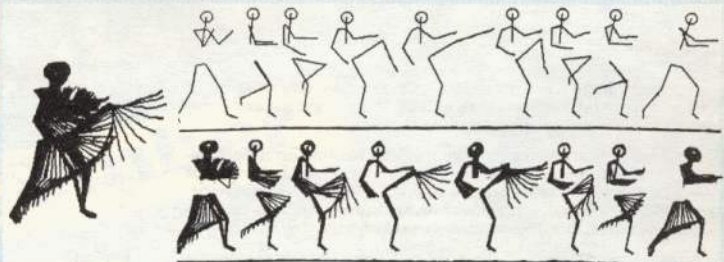
A mellékelt ábrákon levő, ún. egyesenes vonalú rúgást (mae geri) Borza József 2 danos mester hajtja végre. A képsoron látható mozdulat számos érdekes és vizsgálódásra érdemes kérdést rejt magában. Ezek közül ragadjunk ki egyet, a lábfej mozgását. A lábfej ugyanis az a testrész, amely a rúgás pillanatában érintkezik az ellenféllel, és így a mozgás során a leghosszabb utat teszi meg. A cél, hogy a találat pillanatában a 90°-ot minél inkább közelítő szöget és minél nagyobb sebességgel érjük el. A képsorozaton szemléletesen mutathatók be a változások.

A küzdősportokban általában nagy, a karatéban pedig kiemelt fontossága van

az egyensúly megőrzésének. Ez persze nem statikus, hanem dinamikus egyensúly. Az elemzések alapján egyesenes vonalú rúgás esetében akkor beszélhetünk bizonyos egyensúlyi helyzetéről, ha rúgáskor a láb — kinyúlása után — jelentősebb kibillenés nélkül indul vissza az eredeti küzdőállás felé. Az ábra alapján megállapítható, hogy Borza József biztos egyensúlyi helyzetben hajtotta végre a rúgást.

A kiragadott példán is jól érzékelhető, hogy a biomechanikai filmelemzéses módszer kiegészíti és alátámasztja az eddig empirikusan szerzett ismereteket a karate különböző mozgássorozataiban. A mozgáselemzéssel a korrekcióhoz, az optimális végrehajtáshoz vezető utat keresik.

A Bárdosi Zsolt és Nagy László által kialakított módszer segítségével sok más sportmozgás számítógépes elemzése elvégezhető hasonló módon. A film alapján a számítógéppel szárad-, ezredmásodpercenként figyelhető a sportoló mozgása komplexen és részleire bontva is. A megállapítások nyomán az edzéseken változtatni lehet a mozgásokat, így a teljesítmény fokozható.



A fázisrajzok a valóságnak megfelelően

A fázisrajzok fázisonként önálló képre bontva

Borza József 2 danos mester egyesenes vonalú (mae geri) rúgása. A pálcikafigurákat is a mikrogép rajzolta ki

## Árnyékolják a képernyősugárzást!

A starnbergi (NSZK) Nokia Information Systems vállalatnál kifejlesztett mágnes-tekerces és földeléses rendszerrel sikerült csaknem teljesen megszüntetni a képernyőkből kilépő mágneses tésugárzást. A képernyőtől 30 cm-es távolságban a mágneses sugárzás az eddig mért értékek csupán a tízede.

A képernyő felületének földelésével a villamos terek is megszüntethetők. Bár a képernyők sugárzási szintje tulajdonképpen veszélytelen, az az olcsó megoldás mégis megnyugtathatja azokat, akik képernyő előtt dolgoznak.

## Az okosak terve: „okos” város

Az „elektronizált ház” fogalmával már hazánkban is a gyakorlatban ismerkedtünk. Például Budapesten, a Korona cukrászda épületében már üzemel az elektronikus házmester, felügyelve a liftet, a szellőzőmotorokat, a füst- és betörérsjelző érzékelők jeleit.

Japánban már az „okos” városok, illetve városrészek tervezése van folyamatban. Arra gondolnak, hogy az új épületek lakóit egy számítógépes rendszer arról is értesíti, hol van éppen forgalmi dugó, üres parkolóhely vagy olyan áruházzal, amelyben nem tolonganak a vevők.



# Pontvadászat

Új rejtvényt indítottunk újtára 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 kapnak.

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

**Beküldési határidő: 1988. május 16.**  
Címünk: **Mikroszámítógép Magazin Szerkesztősége**  
1371 Budapest, Pf. 433.

Jó vadászatot kíván a feladatok összeállítója:  
dr. Hoffmann Tibor

## 1. feladat

Lehetőleg teljes indukcióval mutassuk ki, hogy az egész számok bármely hatványának összege mindig osztható  $n(n+1)$ -gyel! (5 pont)

E feladat megoldásának gondolatmenete hasonló az 1988/3. számban megjelent 1. feladatéhoz.

## 2. feladat

Ismeretes, hogy a szubrutinok alkalmazása mindig több futási időt igényel, mint a szubrutinok nélküli futás. Ugyanakkor azonban a program memóriahelyet takaríthat meg.

Egy bizonyos számítógépen egy  $p$  számú paraméterrel (felhasznált változóval) rendelkező szubrutinnak az egyszeri behívásához  $5p+8$  memóriahely szükséges. A szubrutin magja  $m$  memóriahelyet foglal el, s ennek a memóriában szubrutinként való elhelyezéséhez további  $2p+4$  memóriahely szükséges.

Annak a feltételezésével, hogy az egyes memóriahelyek egyenértékűek, állapítsuk meg, hogy adott  $p$  és  $m$  mellett hány hívásnál több esetén lesz gazdaságos a szubrutin felhasználása? (4 pont)

Vizsgáljuk ezzel meg a szubrutinok alkalmazási gazdaságosságát  $p=0,1$  és 2 paraméter esetében. (1 pont)

## Az 1988/2. szám feladatainak megoldása

### 1. feladat

Ha a sorozatból a két egymás utáni elem négyzetének különbsége páros szám, akkor ezek két, egymástól 2-vel különböző számból keletkeznek. Így

$$(a+2)^2 - a^2 = 4a + 4$$

A megadott első számmal így 142187 és 142189 adódik. (2 pont)

Ha ez a különbség páratlan, akkor a sorozatban ez két egymás utáni számot jelent, tehát

$$(a+1)^2 - a^2 = 2a + 1$$

A megadott második számmal ekkor 284377 és 284378 adódik. (2 pont)

### 2. feladat

A számjegyek megsokszorozott képe alapján a magasság mindig nagyobb, mint a szélesség. A szélességben  $n$  nem lehet 2, mert ebben az esetben nem tudunk olyan számjegyet kirajzolni, amely valamilyen „üreges” belső részt használ, például 0, 2, 3, 5, 6, 8, 9.

Az üreges jelleg ábrázolásához tehát minimálisan  $n=3$  kell. De ez elég is, amint a következőkben látni fogjuk.

Magasságban a középvonal elválaszthatósága például a 3 ábrázolásánál azt adja, hogy legalább  $m=5$ -nek kell lennie. Mint az ábrából látható, ez elegendő is. Tehát  $3 \times 5$ -ös méretre a lehető legkisebb. (3 pont)

o o o	. o .	o o o	o o o	o . o	o o o
o . o	o o .	. . o	. . o	o . o	o . .
o . o	o . .	o . .	o . o	o o o	o o o
o . o	. o .	o . .	. . o	. . o	. . o
o o o	. o .	o o o	o o o	. . o	o o o

o o o	o o o	o o o	o o o
o . .	. . o	o . o	o . o
o o o	. . o	o o o	o o o
o . o	. . o	o . o	. . o
o o o	. . o	o o o	o o o

Az ábráról leolvashatjuk, hogy a felhasznált pontok száma az egyes számjegyeknél az alábbi:

„0” = 12	„5” = 11
„1” = 6	„6” = 12
„2” = 9	„7” = 7
„3” = 10	„8” = 13
„4” = 9	„9” = 12

(1 pont)

Mivel a legnagyobb pontszámot a „8” számjegy használja fel, a mátrixnyomatók tesztelése a 8-as számokkal célszerű. Mint azt a mátrixnyomatókat használó kollégák tudják, ez nemcsak erre a méretű mátrixra van így, hanem a gyárak a nagyobb méretűekre is a csupa 8-as számjegyekből álló számot adják meg hivatalos tesztnek. (1 pont)

Ingyenes átvizsgálással,  
a legalacsonyabb javítási díjjal,  
a legrövidebb határidővel, hathavi garanciával,  
Budapest központjában, a Corvin mozi mellett

**vállaljuk**  
**személyi számítógépek és videoberendezések**  
**javítását.**

**Ígény esetén átalánydíjas javításra is vállalkozunk.**

**Gelka Spectrum Leányvállalat Computer-video szerviz**

Budapest, Kisfaludy köz 4. 1082. Telefon: 343-999.





Buborékok  
(Felföldi Anna)

## A gép is ember

A Neumann János Számítógéptudományi Társaság karikatúrapályázatára nyolc országból százöt karikaturista háromszáz rajtot küldött. A nagydíjat (25 000,- Ft) Vjekoslav Klaič, Jugoszlávia nyerte.

- A hagyományos módon rajzolt karikatúrák kategóriájában
- I. Jurij Koszobukin, Szovjetunió (15 000,- Ft),
  - II. Balázs-Piri Balázs, Magyarország (10 000,- Ft),
  - III. Louis Postruzin, Ausztrália (5000,- Ft).
- A számítógéppel készített karikatúrák kategóriájában
- I. Halász Géza, Magyarország (15 000,- Ft),
  - II. Felföldi Anna, Magyarország (10 000,- Ft),
  - III. Kiss Katalin, USA (5000,- Ft).

Néhány karikatúrát, függetlenül a helyezésektől, alkalmatosan közreadunk.



