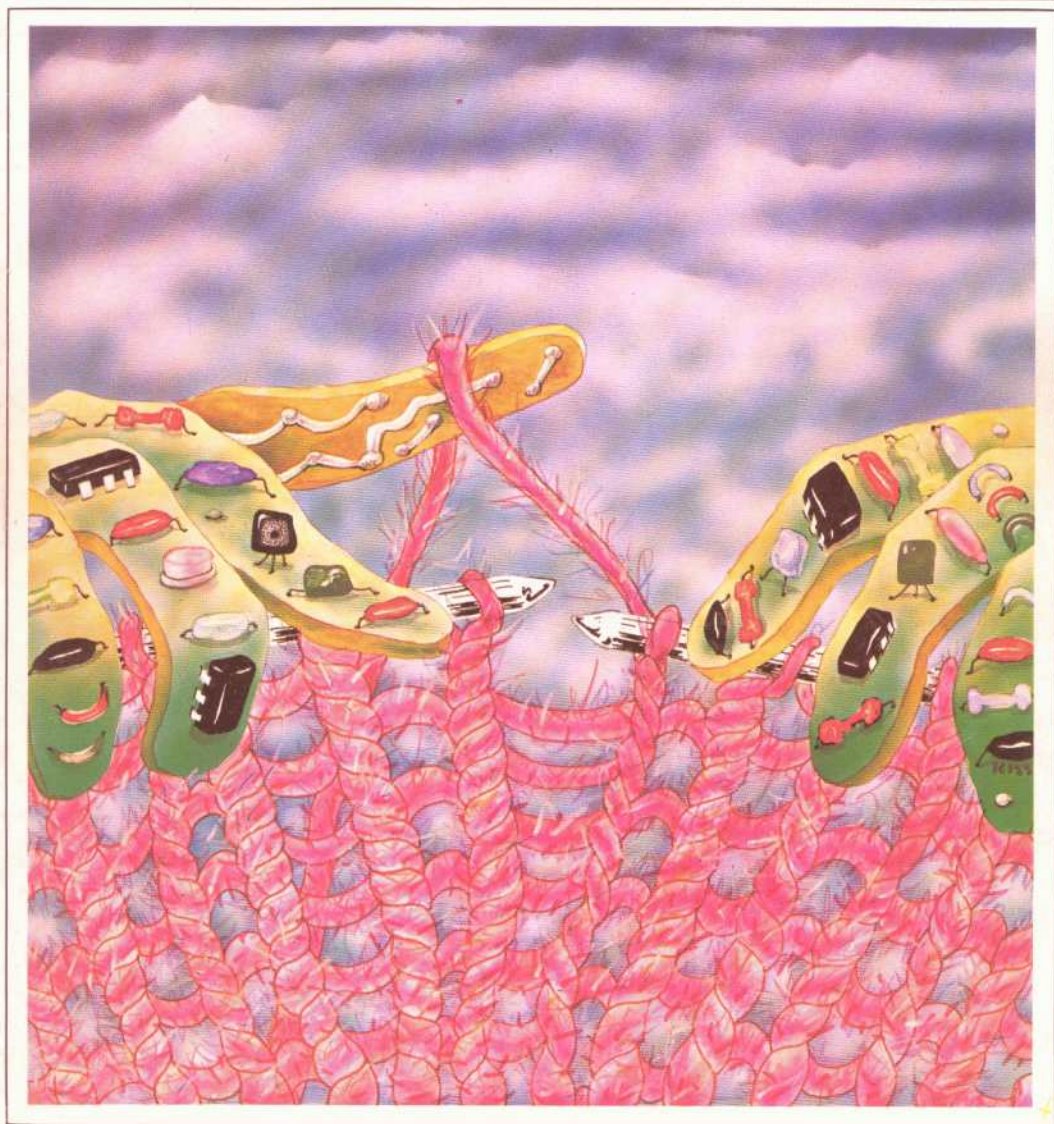


mikro

számítógép

magazin

Ára: 30 Ft





mikro számítógép magazin

A NEUMANN JÁNOS SZÁMÍTÓGÉPTUDOMÁNYI TÁRSASÁG LAPJA

A szerkesztőbizottság vezetője:
Kovács Győző

A szerkesztőség munkatársai:
Bakos Tamás
(programozástechnika)

Broczkó Péter
(hírek)

Kovács Győző
(levelezés)

Nagy Imre
(tanuljunk együtt)

Petróczy Judit
(könyvek)

Pinke György
(NJSZT, alkalmazások)

Soltészné Vizi Zsuzsa
(tervezőszerszám)

Simonyi Endre

Szebenszki Sándor
Szulyovszky Csaba

Tamásné Lakó Erika
Terebessy Ákosné

Címképpünk:
Kiss Ilona munkája

Felelős szerkesztő:
Könyves Tóth Pál

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

Levélcím:
1371 Budapest
Pf. 433

Kiadja:
MTESZ Neumann János
Számítógéptudományi Társaság
1054 Budapest, Báthori u. 16.

Levélcím:
1368 Budapest 5. Pf. 240

Telefon: 329-349

Felelős kiadó:
Tóth Istvánné
ügyvezető főtítkárhelyettes

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

Megjelenik havonta.
Egy szám ára 30.— Ft
Előfizetési díj:
egy évre 360.— Ft
fél évre 180.— Ft
Külföldön terjeszti
a Kultúra,
1389 Budapest, Pf. 149
és a Magyar Média
1932 Budapest, Pf. 279
88—1552



Szikra Lapnyomda
Budapest (88—1966)
Felelős vezető:
Csöndes Zoltán vezérigazgató

INDEX: 25 629
ISSN 0236-6088

TARTALOM

- 2 Technológizált tanulás
- 10 Feladatok – megoldások
- 19 Mintakötés nemcsak hölgyeknek
- 21 24 vonalas be-, kimeneti bővítő
- 22 Nem rejtjük véka alá
- 28 Rendszerfejlesztési eszközök
- 31 Merre tart a világ?
- 34 A Motorola 68000-es mikroprocesszor
assembly programozása
- 40 Olvastunk ...
- 44 Programtermék
- 46 Adok—veszek—cserélek

TANULJUK EGYÜTT!

3

- 3 A Pascal rejtelmel
- 6 Verseny a javából!
- 6 Hangos fejbeállítás
- 7 Ne bánts d a magnót!
- 8 Hova tovább?!

CSIPEGETŐ

12

- 12 Kilépés a sprite-térbe
- 12 Bájtsorozatok tömörítése
- 13 Örökélet
- 13 Képcsere — bitenként
- 14 Video-RAM
- 15 Az ugrálgatós
- 15 TOP-lista

PROGRAMOZÁSTECHNIKA

16

- 16 A számítógép motorja
- 18 Programozási fogások és melléfogások

ENTERPRISE

23

- 23 Átalakítás németről angolra
- 25 Megkérdeztük az Enterprise-ről
- 26 Dölnék a dominók
- 27 Mi a manó?

SAKK

42

- 42 Centrumkontroll

AZ OLVASÓ ÍRJA

43

KÖNYVEK – HÍREK – ÉRDEKESSÉGEK

45

PONTVADÁSZAT

47

 mikro számítógép
magazin



1989.02.04. és 07.23. 1989.02.04.

Nyugat-Európa konvergenciát és kiállítást látja, az ember egyre több olyan vállalkozással találkozik, amelyeknek a fő terméke a számítógépre készített tananyag, vagy ahogy sokan ismerik, a CAI (Computer Assisted Instruction), vagy magyarul: számítógéppel segített oktatási program. Az elnevezés csak akkor igaz, ha a tananyagot a tanár az előadott anyag illusztrálására használja. Ha olyan termék készül, amit a diák egyedül, tanári jelenlét nélkül is képes megtanulni, akkor jobb, ha a CAL (Computer Assisted Learning), vagyis a számítógéppel segített tanulás kifejezést alkalmazzuk. A kompromisszumokra hajlóknak van még egy harmadik lehetősége is, a CBT (Computer Based Training), azaz a számítógépre alapozott képzés, ami közelebb áll a CAI-hoz, de érthető rajta a CAL is. Én magam a CAL, illetve a CAL-t alkalmazó tanulási rendszernek, a távtanulásként (distance learning) a híve vagyok.

Mindez most arról jutott eszembe, hogy október elején Budapesten tartotta zo-

„Az oktatásban elért eredményeink közismertek, és ezek az eredmények igen jelentősek a legfejlettebb országokhoz viszonyítva. Mégis vigyázzunk a felső- és középfokú oktatás reformját. Miért? Először is azért, mert a modern társadalom új követelményeket támaszt az emberrel szemben. És ezenkívül azért, mert a pangás jelenségei hatással voltak oktatási rendszerünkre is: itt is túrtük az önelégültséget, beértük azzal, amit már elértünk, és ez rögtön megmutatkozott mindenben.”

(Mihail Gorbacsov: Átalakítás és új gondolkodás)

A számítógépes alkalmazások területén — véleményem szerint — a helyzet nem sokkal különbözik a szoftverrel kapcsolatban elmondottaktól. Ami pedig a CAL-t illeti, szerintem ezen a téren az elmaradásunk azért nem mérhető, mert nincs mihez viszonyítani. A courseware-fejlesztési technológia ugyanis nem régei jelent meg először az országban.

A budapesti megbeszélésen részt vevő cégek szakértői dr. Bert Camstrának a vezetésé-

kesztési trükkjei. Mindezeket vasos és kevésbé vasos könyvek foglalják össze azért, hogy az összetartozó tananyagok lehetőleg egységesek legyenek. Ezeket a szabványokat hosszú évek tapasztalatai alapján dolgozták ki, és bennük a szerzők nagyon sok pszichológiai, pedagógiai és módszertani tapasztalatot gyűjtöttek össze.

Említettem már, hogy a számítógépes tananyag nem tankönyv, amit képernyőre írtak. Ebből nyilvánvalóan az is következik, hogy csak olyan tananyagot szabad számítógépre vinni, amelynek a megértéséhez a számítógép alkalmazása feltétlenül szükséges. Néhány példa. Egy folyamatábrát könnyebb úgy elmagyarázni, hogy egy képernyőn lépésenként építem föl az egyes elemekből az egészet, animációval bemutatva az információ áramlásának irányát, és mindig csak annyi szöveget írva az ábra mellé a képernyőre, amennyi az adott lépés megértéséhez szükséges. Vagy egy másik példa. A legújabb szerzői rendszereket írt tananyagokhoz a tanulás bizonyos fázisaiban be lehet hívni előre elkészített, például szimulációs vagy alkalmazói programokat és így nem csak szimulált környezetben, hanem az eredeti alkalmazói rendszert alkalmanként behívva lehet például a kezeléshez szükséges ismereteket megtanítani.

Ennek az írásomnak nem az a célja, hogy a számítógépes tananyagkészítés technológiájáról részletes ismertetést adjak. De talán sikerült érzékeltetnem mennyire fontos számunkra, hogy ezen a területen is felzárkózzunk az európai élelményhöz, hiszen különböző esélyünk sincs arra, hogy a tananyagok formájában létrejött szellemi termékeinket legalább az európai piacon értékesíthessük. Ennek az írásomnak bevallott célja az is, hogy a hazai iskolaügyért felelősök, akik előtt különféle javaslataink már hosszú ideje hevernek, végre megmozduljanak, és tegyenek valamilyen technológiai távtanulási bevezetése érdekében. A virágözék minden virág elvének általában híve vagyok, de ebben az esetben nem. Talán az elmondottakból is látszik, hogy a hazai oktatási intézményekben egységes technológiát kellene bevezetni, és így az oktatási kormányzatnak sürgősen döntenie kellene ebben a kérdésben. Persze egyáltalán nem ígylem a művelődési minisztert, hiszen az iskolák meg vannak rakva a technológiai tananyag készítésére alkalmas hazai számítógépekkel, amelyeket előbb vagy utóbb nem kevés anyagi áldozat árán PC kompatibilis eszközökkel kell majd kiváltani.

Nem hiszem, hogy a döntés sokáig halogatható lenne, tudniillik, nem tudom elképzelni, hogy az oktatásügy ma már a diákság által is követelt reformját technológiai rendszerek bevezetése és az európai, illetve a nemzetközi élvonalhoz való felzárkózás nélkül meg lehetne valósítani.

Kovacs Győző

Technológizált tanulás

kásos évi összjevitelét a hollandiai központi Courseware Group, amelyhez eddig tizenegy ország tizenkét professzionális vállalata csatlakozott.

Az ülésen részt vett az osztrák székhelyű Hofbauer Group is, amelynek egyelőre hat vállalata van négy országban. Ennek a közel húsz cégnek a legfontosabb tevékenysége a számítógépes tananyag- (courseware-) fejlesztés, de a fejlesztés szoftvereszközeinek, például a szerzői rendszereknek a kidolgozása is. Ez a két csoport elsősorban Európában tevékenykedik, de a szervezetnek Európán kívül az USA-ban és Indiában van már képviselése, és a jelek szerint hamarosan újabb országok újabb vállalkozásai csatlakoznak a két csoporthoz.

Nem érdektelenül írok mindezekről, hiszen javaslatomra hamarosan a SZÁMALK a bécsi székhelyű Courseware Learnservice GmbH társtulajdonosoként tagja lesz ennek a nemzetközi szervezetnek. A csatlakozásnak az a célja, hogy elsősorban az informatikai képzésben nagyobb legyen a világra való tekintés, és lehetőleg legyen az aktív nemzetközi részvételre mind a tananyagok cseréjében, mind a fejlesztésben.

Ma már minden fórumon elismerik, hogy az ország a hardverfejlesztésben jócskán lemaradt, már nemcsak az ipariilag fejlett országok, de jó néhány volt „fejlesztő” ország, mint Tajvan, Dél-Korea vagy akár Thaiföld mögött is. Sokáig döngötték mellünk, hogy szoftver világhatalom vagyunk, de ez is ma már csak illúzió.

Tagadhatatlan, hogy vannak nagyon jó szoftvertermékeink, de ezek közül csak nagyon kevés az, ami az európai átlagot jóval meghaladja, és így kemény árunak tekinthető. Ezért nem engedélyezték korábban például a szoftverrel kompenzált külkereskedelmi özeteket, mondván, hogy a magyar szoftver — kemény áru lévén — minden piacon eladható. Azt ma már mindenki tudja, hogy messze nincs így. Ezért aztán sokan várjuk a Beck Tamás vezette, várhatóan megújult szellemi Kereskedelmi Minisztériumtól, hogy il-lúziók helyett a valóságos értékeket fogja védeni, és a jelenlegi korlátozások helyett az árforgalom, és így a szoftverforgalom liberalizálását is meg fogja teremteni.

vel vitakoztak a minőség, valamint az egyre igényesebb piac problémáiról. A vitában elhangzottak alapján megpróbálom összefoglalni a számítógépes tananyag és tananyagkészítés fontosabb jellemzőit.

A számítógépes tananyag — először is — pedagógiai, metodikai, szakmai és nem számítástechnikai produktum, azért azt pedagógusoknak, pszichológusoknak, az adott, a tanulandó témához értő szakembereknek és nem számítástechnikusoknak kell készíteni. Kívétel persze a számítástechnika tanúlé, de ott sem közömbös, hogy a tananyag készítőinek van-e pedagógiai érzékük és ismeretük.

A számítógépes tananyag éppen ezért — pedagógiai értelemben — nem számítógépes program. Sokkal közelebb áll a tankönyvhez, mint egy programhoz, de ugyanakkor a tananyag nem elektronikus tankönyv, mert akkor az már nem „courseware”.

A számítógépes tananyag készítéséhez, azért, hogy egy számítástechnikában képzelt, de például az irodalomhoz kiválóan értő pedagógus is képes legyen ilyet fejleszteni, nagyon magas szintű, könnyen megvalósítható és kezelhető eszközök, könnyen nyelvek vagy szerzői rendszerek kelljenek.

A szerzői nyelv utasításokkal írja le a képernyő tartalmát (szöveg, ábra, fotó, grafika, animáció, visszakerézés/teszt stb.). A szerzői rendszerekkel közvetlenül a képernyőn lehet dolgozni, legtöbbször ablaktechnikával, egérrrel vagy tablettal adva az utasításokat.

Ellentétben más alkalmazói rendszerekkel, a számítógépes tananyagot úgy kell elkészíteni, hogy ahhoz bárki hozzányúlhat, azt kiegészíthesse, törölhesse belőle, a „lapokat” átcsoportosíthassa, kurzusokat összehasonlíthasson vagy zétáradalhat.

A tananyagkészítésnek az őS-PLATO rendszer óta — a hatvanas években a Control Data Corporation munkatársai készítettek először ún. PLATO rendszerben tananyagot „nagy” számítógépre, ami végül is nagy pedagógiai, de igencsak mérsékelt gazdasági siker volt — megvannak a maga szabályai és szabványai, technikai és grafikai, szövegszer-



A PASCAL REJTELMEI

2. A Turbo Pascal 3. XX verziója

2.1 A szükséges rendszerfájlok

Jelenleg már a 4.0-es verzió is létezik. Ennek használata ugyan lényegesen kényelmesebb egy, a Pascalban járatos programozónak — például a formátum-, grafika-, szín-, fájlkezelést támogató nagyszámú beépített eljárása révén —, de kezelése sok tanulást, gyakorlást, első ránézésre „pilotávizsgát” igényel. Jellemző tény, hogy a Turbo Pascal 4.0 kézikönyve, amit a szoftverfejlesztő cég nyilván nem a kezdőknek szánt, majdnem 800 oldal terjedelmű. Ha az olvasók közül valaki a már meglévő — 3.XX verzió szerint készült — programjait később a 4.0-es verzióban kívánja futtatni, ennek általában nincs akadálya, mert a szükséges szoftvertámogatást megkapja a 4.0-es változat megfelelő rendszerfájljaitól.

A 3.XX verziók valamelyike a Pascal-lal most ismerkedők számára könnyebben használható eszköz. A változathoz több rendszerfájl tartozik:

- TURBO.COM (fordító)
- TURBO.MSG (hibaüzenetek)
- TINST.COM (installáló program)
- TINST.MSG (installációs üzenetek)
- TINST.DTA (az installáció eredményei)
- TLIST.COM (nyomtatóra listázó program)

A forrásprogramokat a rendszer .PAS kiterjesztésű fájlokban tárolja.

Egy program fordításához és futtatásához csak a TURBO.COM fájlra van szükség, tehát az ezt tartalmazó lemeznek feltétlenül a gépben kell lennie. Ha nincs fix lemezegységünk, a legegyszerűbb a fordítót ugyanazon a hajtékonylemezen (floppy) és ugyanabban a könyvtárban (directoryban) tárolni, ahol a forrásprogram is van. Ha hibaüzeneteket is kívánunk — márpedig ez a

tanuláskor elengedhetetlen —, a TURBO.MSG fájlra is szükség van. Az installációval — a Pascal üzembe helyezésével az adott képernyőtípusra és saját szerkesztési parancsaink beépítésével a rendszerbe — nem foglalkozunk, tehát a TINST fájlokra nincs szükség, bár a „számítástechnikai angolt” ismerők megpróbálkozhatnak a TINST.COM futtatásával.

A TLIST.COM fájl a forrásprogramokat listázza a nyomtatóra, ami főleg akkor hasznos eszköz, ha programjaink mérete 2-3 képernyőnél nagyobb, mivel így a bogarászás a nyomtatott listán sokkal kényelmesebb.

2.2. A Turbo Pascal indítása

Ha a szükséges rendszerfájlok aktivizálhatók — az MS/DOS hozzáfér az azokat tartalmazó könyvtárhoz —, az indítás a TURBO parancssal történik. Ennek hatására a fordító betöltődik és elindul. A képernyőn az 1. ábrán látható kép jelenik meg.

A Color display 80×25 üzenet a szerző gépén végzett installációból ered. A feltett kérdésre: Hibaüzenetek szükségesek? igennel (Y) válaszoljunk. Ha a TURBO.MSG fájl a rendszer számára nem érhető el, hibaüzenet keletkezik, ha elérhető, betöltődik a tárbá.

A válasz után az ernyőn a 2. ábrán látható parancs főmenü jelenik meg. A menüelemek nagybetűivel választhatók ki az egyes funkciók. Ha a TURBO promptja, a > jel és a villogó kurzor látható, a rendszer a parancsokat elfogadja. Más billentyűk lenyomására a — már megváltoztatott — főmenü jelenik meg

- L: aktív lemez megváltoztatása
 - A: aktív könyvtár megváltoztatása
 - W: a munkafájl nevének megadása
 - M: a főfájl nevének megadása
 - E: szövegszerkesztés
 - C: fordítás
 - R: futtatás
 - S: tárolás
 - D: a könyvtár tartalmának kiírása megadható maszk szerint (a maszkban az ún. joker karakter, a * is alkalmazható)
 - Q: kilépés a Pascal rendszerből. — Ha az éppen szerkesztett programot még nem tároltuk, a rendszer figyelmeztető üzenetet küld. Ha a feltett kérdésre: SAVE (Y/N)? Y-nal válaszolunk, a programot a kilépés előtt a Pascal tárolni fogja. Ha a válasz N, a kilépés tárolás nélkül megy végbe.
 - O: fordítási opciók (memóriába, .COM fájlba vagy összefűzhető, láncolható .CHN fájlba)
- Végül a felhasználható szövegmező és a szabad tárterület (Free) megadása következik.
- A felsoroltak közül kezdetben csak az E, a C, az R, az S és a Q parancsokat fogjuk használni. Ezek alkalmazásának részleteiről és a jellemző ernyőképekről, screenekről a későbbiekben — első programjaink megírásakor — bővebben lesz szó.

2.3 A TURBO EDITOR legfontosabb szerkesztési parancsai

A szövegszerkesztőbe az E parancssal léphetünk be. A szerkesztő a 3. ábrán látható képpel jelentkezik be.



3. Első programunk

A Line n és a Col n az aktuális kurzorpozíciót mutatja. A Line (sor) a sorok számát adja a fájlkezdettől, a Col (oszlop) a karakterpozíciót az ernyő bal oldali szélétől számítva. A szerkesztő alaphelyzetben Insert, beszúrás üzemmódban van (3.a ábra). Ez azt jelenti, hogy a pillanatnyi kurzorpozícióban billentyűzött karakter a sorba beillesztődik, a tőle jobbra levő karakterek jobbra tolódnak egy pozícióval. A szerkesztő alkalmas felülírásra: Overwrite üzemmódra is. Ilyenkor az aktuális kurzorpozícióban lévő karakter helyére a billentyűzött új karakter kerül (3.b ábra).

Az Insert/Overwrite átkapcsolás az Insert billentyűvel végezhető el. Az Indent az automatikus tabulációs üzemmód jelzője, ki-be kapcsolása a Ctrl-Q Ctrl-I kombinációval lehetséges. A sor végén a szerkesztett fájl neve olvasható, ábránkon: ALMA.PAS. A fájl neve előtt az aktív lemez jele van.

A szövegszerkesztő sok lehetőséget kínál a felhasználónak. Most csak azokat az alapvető parancsokat ismertetjük, amelyeknek a segítségével viszonylag kényelmesen állíthatjuk össze programjainkat a képernyőn. A parancsok egy, esetleg két billentyű vagy billentyűkombináció segítségével adhatók ki.

- Kurzormozgató: a kurzorvezérlő billentyűkkel
- Karaktertörlés: Backspace (kurzorpozíciótól balra) Delete (kurzorpozícióban)
- Egy szóval balra: Ctrl-A
- Egy szóval jobbra: Ctrl-R
- Sorkezdetre állás: Home
- Sorvégre állás: End
- Sortörlés: Ctrl-Y
- Sor beszúrása: Ctrl-N
- Egy lappal vissza: PgUp
- Egy lappal előre: PgDn
- Fájlkezdetre állás: Ctrl-PgUp
- Fájlvégre állás: Ctrl-PgDn
- Kilépés a szerkesztőből: Ctrl-K Ctrl-D

Azok, akiknek — például más géptípuson — némi nyelvismeretük van, az eddigiek alapján akár már meg is próbálkozhatnak a munkával.

A konkrét nyelvi ismeretek — Pascal-utasítások, -eljárások, -szabályok stb. — részletes leírását és példaprogramokon való bemutatását ezennel megkezdjük.

Hogy figyelmünket elsősorban a nyelvi problémákra koncentrálhassuk, válasszunk egy igen egyszerű feladatot. A billentyűzetről vigyünk be két olyan egész számot, amelyeknek abszolút értéke <= 255, végezzük el velük a négy matematikai alapműveletet, majd az eredményeket vigyük ki a képernyőre!

A feladat megoldásához — eddigi ismereteinken kívül — két standard: a bevitelt szolgáló **read** és a kivitelt szolgáló **write** eljárásra is szükség van. A két eljárás szintaxisa igen egyszerű:

read (fájlnev, változólista)
write (fájlnev, adatlista)

Ha a be/kivitel nem lemez fájlra, hanem az ún. standard perifériákra vonatkozik — többek között a billentyűzet és a képernyő ilyen —, a fájlneveket nem szükséges kiírni. Ha ugyanis az eljáráshívásokban lévő zárójeles rész deklarált változónevekkel vagy a Pascal számára felismerhető egyéb adatnevekkel

kezdődik, a fordító a fájlnevet nem keresi: **read** esetén a billentyűzetpuffert, **write** esetén a képernyőpuffert feltöltézi, mindkettőt 127 karakteres fizikai fájlként.

A **read** eljárásnevet követő zárójelben a változóneveket egymástól a , karakterrel elválasztva kell felsorolni. A bevitelkor minden egyes adat bevitelét ENTER-rel kell befejezni. A **write** eljárásnevet követő zárójeles rész adatlistája változóneveket, kifejezéseket, szám- és szövegkonstansokat egyaránt tartalmazhat. Az egyes listaelemeket itt is a , (vessző) választja el egymástól. A szövegkonstansokat ' (aposztróf) jelek közé kell tenni.

A programban egyébként felhasználtuk a Turbo Pascal egyik egyszerű, a képernyő törlését végző eljárását is, a **clrscr**-t. Célunk ezzel mindössze annyi volt, hogy a futtatási képet ne zavarják a

```

-----
TURBO Pascal system          Version 3.01A
                               PC-DOS

Copyright (C) 1983,84,85    BORLAND Inc.
-----

Color display 80x25

Include error messages (Y/N)?
  
```

1. ábra

```

Logged drive: C
Active directory: \PASCAL\PASCAL3

Work file:
Main file:

Edit      Compile Run   Save

Dir       Quit  compiler Options

Text:      0 bytes
Free:     62024 bytes
  
```

2. ábra

```

A Line 1 Col 1 Insert Indent C:ALMA.PAS
  
```

3. ábra

```

B Line 1 Col 1 Overwrite Indent C:ALMA.PAS
  
```




forrásprogram-lista maradványai és a fordításkor keletkező üzenetek.

Ezek után a 4. ábrán minden kommentár nélkül bemutatjuk a program listáját, ami a Turbo Pascal szövegszerkesztője segítségével készült (lásd a TUBRO EDITOR használatával kapcsolatban a 2.3 pontban már közölteket). A szerkesztésből a Ctrl-K Ctrl-D-vel kilépvé megjelenik a 2. ábrán bemutatott parancs főmenü. A programot az R parancssal futtathatjuk. A fordító a programot először lefordítja, majd elindítja.

Az adatok bevitelét után megjelennek az eredmények. A futásra jellemző kép az 5. ábrán látható.

Az ernyőkép formátuma rossz, nem tagolt, szinte áttekinthetetlen. Ha a read és a write helyett a readln és a writeln eljárásokat alkalmazzuk (az eljárásokban levő ln a line: sor szó jelzése), minden ENTER-rel terminált bevétel és minden writeln után CR/LF kerül végrehajtásra. A módosított programlista a 6. ábrán, a futtatási kép a 7. ábrán látható.

Azért, hogy továbbléphessünk, próbáljuk meg ezt a programot úgy futtatni, hogy mindkét bemenő adat abszolút értéke nagyobb, mint 255. Ha máshol nem is, de a szorzásnál helytelen eredményt kapunk. Ennek az a magyarázata, hogy két egybájtosnál „hosszabb” szám szorzata két bájtnban nem ábrázolható, azaz túlléptük az integer típusú változókra megengedett maximális 32 767 értéket. A megoldás igen egyszerű: az a és b változókat ne integer, hanem real típusként deklaráljuk.

A jobb ernyőkép elérésére alkalmazunk egy eddig még nem tárgyalt, a kurzorvezérlést szolgáló:

gotoxy (oszlop, sor)

szintaxisú eljárást is. Ennek eredménye, hogy az ernyőre írás a megadott — példánkban a 3,3, illetve a 2,5 — pozíciókban kezdődik. A legújabb program forrásnyelvi listáját a 8. ábrán, a futtatás eredményét a 9. ábrán láthatjuk.

Nagy Imre

4. ábra

```

program gyak1_1;
var a,b:integer;
begin
  clrscr;
  write('Első gyakorló programunk');
  write(' a=');read(a);
  write(' b=');read(b);
  write(' a+b=');write(a+b);
  write(' a-b=');write(a-b);
  write(' a*b=');write(a*b);
  write(' a/b=');write(a/b);
end.

```

Első gyakorló programunka=1b=3a+b=4a-b=-2a*b=3a/b= 3.333333333333E-01
>

6. ábra

5. ábra

```

program gyak1_2;
var a,b:integer;
begin
  clrscr;
  writeln('Első gyakorló programunk');
  write(' a=');readln(a);
  write(' b=');readln(b);
  write(' a+b=');writeln(a+b);
  write(' a-b=');writeln(a-b);
  write(' a*b=');writeln(a*b);
  write(' a/b=');writeln(a/b);
end.

```

7. ábra

Első gyakorló programunk
a=1
b=3
a+b=4
a-b=-2
a*b=3
a/b= 3.333333333333E-01
>

8. ábra

```

program gyak1_3;
var a,b:real;
begin
  clrscr;gotoxy(3,3);
  writeln('Első gyakorló programunk');
  gotoxy(2,5);
  writeln('Az adatok:');
  write(' a=');readln(a);
  write(' b=');readln(b);
  writeln(' A műveletek és eredmények:');
  write(' a+b=');writeln(a+b);
  write(' a-b=');writeln(a-b);
  write(' a*b=');writeln(a*b);
  write(' a/b=');writeln(a/b);
end.

```

9. ábra

Első gyakorló programunk

Az adatok:
a=1
b=3
A műveletek és eredmények:
a+b= 4.0000000000E+00
a-b= -2.0000000000E+00
a*b= 3.0000000000E+00
a/b= 3.3333333333E-01
>



VERSENY A JAVÁBÓL!

Hagyomány folytatásaként 1988 októberében Bulgáriában ismét megrendezték a nemzetközi programozási versenyt. Akiiket e téma érdekol, illetve tudásukat szívesen próbára tennék, azoknak közreadjuk a versenyfeladatot.

Adott egy $m \times n$ méretű, merőleges úthálózatú település. Az úthálózat utcákból és kereszteződésekből áll. A település szélén lévő utcák kezdetét is kereszteződésként kell kezelni. A kereszteződések a település térképének bal felső sarkától indulva sorolva tonosan számozzuk. Utca a két kereszteződés közötti útszakasz, melyet tehát a két kereszteződés közötti zóds számával adunk meg. Az utcák kezdetén és végén közlekedési táblák lehetnek, de egy helyen legfeljebb csak egy.

A közlekedési táblák típusai:

1. Behajtani tilos
2. Öt tonánál nehezebb gépjárművel behajtani tilos
3. Kötvelező haladási irány: jobbra
4. Jobbra kanyarodni tilos

Az utca elején az 1. és a 2. jelű, a végén a 3. és 4. jelű táblák lehetnek.

Készítsünk programot, amely beolvassa a táblák elhelyezését egy olyan úthálózatra, ahol soronként maximum öt, oszloponként maximum négy kereszteződés van. (A szemléletesség kedvéért: a település térképén vízszintesen négy, függőlegesen három háztömb található.) A beolvasásnál meg kell adni:

- a kereszteződés számát az utca elején,
- a kereszteződés számát az utca végén,
- a tábla számjelét az utca elején,
- a tábla súlyát,
- a gépjármű indulási és érkezési helyét a kereszteződés számával.

A program határozza meg azt a kereszteződéssorozatot, amelyik leírja a gépjármű lehetséges útját az indulási ponttól az érkezési pontig (ha egyáltalán ilyen létezik).

A megoldásra négy órát adtak. Már nem tartozik szorosan a feladathoz, így csak a teljesség kedvéért említyük meg, hogy a javasolt programnyelv a Pascal vagy a C volt, továbbá az értékelésnél szempont a változókkal (változónevekkel) való takarékosság is.

Igen sok panasz hangzott már el a C16, C Plus/4 gépek kazettás egységeinek inkompatibilitásáról. Magyarán mondvá: a magnók fejének beállítása már gyárilag sem azonos, és ez a helyzet a használatban csak tovább romlik. Az ebből fakadó gondokon két, egészen más, szinte ellentétes megoldást nyújtó cikkünkkel kívánunk segíteni. Az egyik megoldás a fej beállítását segíti a mindenkor betölteni kívánt programhoz. A másik két gépet és két kazettás egységet használ a program másolásához, feltételezve, hogy a másolandó program a program átadójának magnójával betölthető.

A rovatvezető

Hangos fejbeállítás

A Magazin 1988/9. számában Bácsi Péter Programkazetták, óh! címmel írta le kesergését a magnós betöltéssel kapcsolatban. Ez valóban amolyan örökzöld probléma, annál is inkább, mivel az oktatásban nagyon sok iskolában a magnó a csúcseszköz az adatbevitelre. Ha ez nem jól működik, az egész rendszert nem tudják használni. Az alábbi eljárás bármilyen magnóhoz jó, a leírás azonban C Plus/4-es és C16-os gépekhez való magnókra szorítkozik.

A fejbeállítás elve roppant egyszerű. Az adatmagnótól műszaki szempontból ugyanazt várjuk el, mint más, zenét, beszédet visszaadó készüléktől: minél szélesebb frekvenciasávot vigyen át, azaz jól

vigye át a magas hangokat. Ebben a megközelítésben tehát a jó minőségű adatbejátszást erősítőn keresztül hallgatva nagyon szép, éles hangzású nyivákolást halunk. Ha ilyenkor elcsavarjuk a magnófejet, a „hang” tompul, gyengül, hasonlóan a zenés kazettákhoz. Különösen igaz ez a turbósított programokra, hiszen a gyorsabb betöltés magasabb frekvenciájú impulzussorozatot idéz elő.

A megoldás az iméntiekből adódik: készítsünk „hangfrekvenciás” csatlakozót a magnóhoz. Ez lehet egy 25–70 cm-es árnyékolt vezetékhez kötött tucheldugó vagy a magnó házába csavarozott Jackdugó-aljzat.

Ehhez a művelethez először természete-

tesen szét kell szedni a magnót; az alján lévő négy csavar oldásával az alsó lemezt leemeljük. Helyezzük magunk elé a magnót felfordítva, levett hátlappal. A billentyűk felénk nézzenek. A RECORD billentyű vonalában van egy soklábú tolókapcsoló. Ennek a felénk eső második lábáról egy piros vezeték is. Az árnyékolást a mellette levő összefüggő fóliafelületre kössük. Egyébként nyugodtan próbálkozhatunk a kondenzátor szabad végével, csak zárlatot ne okozunk!



A bekötés után próbáljuk ki a csatlakozót. Jól kell hallanunk az erősítőt — ami lehet egy táskártyás magnóbemenete is — a kazetta jellegzetes hangját, de műsoros kazettát is lejátszhatunk így. Az alsó fedél visszacszerelése után a csavarokat húzzuk meg jól, mert ezek is befolyásolják a fejállást! Az új árnyékoló vezetékét a régi mellett a gumiszemen keresztül búj-tathatjuk ki.

Ezek után már szinte gyerekeként a fejállítás bármilyen programhoz. A fejállító csavart forgatva figyeljük a hang magas-

ságát, élességét — vájt fülék esetleg a szalagzajt is —, mert ezeket kell maximumra állítani. Ha jó a program, bárhonnan is származik, az ekként beállított fej-eljéle kétségtelenül sikerül a betöltés.

Ezt a módszert különösen jól alkalmazhatják klubok, oktatótermek, programok csereberéjével foglalkozó közösségek. Ilyen helyeken egy magnót érdemes ilyen „fejbeállító” rendszerűvé átalakítani, a bejlesztett programot a gépből saját magnóra menteni. A túl gyakori fejállítgatás nem használ a rögzítőrendszernek, ezért

javasolom csak egy magnó felhasználását.

Ez az eljárás egyébként nagyobb biztonságot ad a tape-to-tape rendszerű programátvitelkehez is.

S még valami, ráadásként. Gyakori, hogy egy C-60-as kazettán teljesen elkeveredünk a programok között, az egész rendszer lemerevedik, nem találja a program elejét. Némi gyakorlattal ezt „fülré” megtalálhatjuk, s egy memóriatorlás után indíthatjuk a bejlesztést.

Szálkai Antal

Ne bántsd a magnót!

A C16, C Plus/4 gépek használóinak nagy bánatára az egyik datasette-en felvett program egy másikkal igen gyakran nem tölthető be. Rendszerünk kialakításánál és a működései program megírásánál abból a feltételezésből indultunk ki, hogy a saját magnóval kimentett programok ugyanazzal be is tölthetők.

Hogy mindenki a saját magnóját használhassa, egy program átvételéhez két gépet és két kazettát egységet alkalmazunk. A két gépet egy szabványos CBM csatlakozókábelrel a soros buszcsatlakozókon összekapcsoljuk. Az átmásolandó programot — programokat — tartalmazó kazettát az egyik (ezenül forrás-) géphez kapcsolt magnóba helyezük. A forrásgép-

be betöltjük a gépi kódban megírt másoló-programot, amely a forrásgép kazettás egységéből a másolandó programot bitenként olvassa, majd „teszi ki” a soros buszra. A soros buszra kerülő adat a lévő — vevő — géphez kapcsolt magnóba másolt szalagra íródik. Ehhez sem külön hardverre, sem szoftverre nincs szükség, ez a gép áramkörü felépítéséből következik.

A megoldás kiküszöböli a géptulajdonosok körében sűrűn alkalmazott magnófejállítgatást, vagyis újabb hibák forrását. Megítélésünk szerint a lehető legjobb ez a rendszermegoldás azoknak, akiknek csak kazettás egységük van a programok másolásához. Természetesen hiányossága az, hogy egy harmadik magnóval felvett prog-

```

. 3003 A9 00 LDA #00
. 3005 20 70 DB JSR #DB70
. 3008 60 RTS
. 3009 58 CLI
. 300A A9 93 LDA #93
. 300C 20 02 FF JSR #FFD2
. 300F A9 FD LDA #FFD
. 3011 25 01 AND #01
. 3013 85 01 STA #01
. 3015 4C 0A 80 JMP #800A
. 3018 78 SET
. 3019 A9 EF LDA #EF
. 301B 20 06 FF AND #FF06
. 301D 80 06 FF STA #FF06
. 301F A9 0F LDA #0F
. 3021 85 00 STA #00
. 3023 A9 F7 LDA #F7
. 3025 25 01 AND #01
. 3027 85 01 STA #01
. 3029 20 E0 30 JSR #30E0
. 302B AD 00 FF LDA #FF00
. 302D 80 19 FF STA #FF19
. 302F A0 00 LDY #00
. 3031 8C 00 FF STY #FF00
. 3033 20 EE 30 JSR #30EE
. 3035 8C 30 FF STY #FD30
. 3037 AD 08 FF LDA #FF08
. 3039 C9 FB CMP #FB
. 303B 00 E5 BNE #30AB
. 303D A9 10 LDA #10
. 303F 00 06 FF ORA #FF06
. 3041 80 06 FF STA #FF06
. 3043 0A 08 LDR #08
. 3045 85 01 ORA #01
. 3047 85 01 STA #01
. 3049 A9 EE LDA #EE
. 304B 80 19 FF STA #FF19
. 304D A9 80 LDA #80
. 304F 8C FC 07 STA #07FC
. 3051 58 CLI
. 3053 60 RTS
. 3055 A9 FD LDA #FFD
. 3057 25 01 AND #01
. 3059 85 01 STA #01
. 305B 60 RTS
. 305D 85 01 STA #01
. 305F 85 01 STA #01
. 3061 85 01 STA #01
. 3063 85 01 STA #01
. 3065 85 01 STA #01
. 3067 85 01 STA #01
. 3069 85 01 STA #01
. 306B 85 01 STA #01
. 306D 85 01 STA #01
. 306F 85 01 STA #01
. 3071 85 01 STA #01
. 3073 85 01 STA #01
. 3075 85 01 STA #01
. 3077 85 01 STA #01
. 3079 85 01 STA #01
. 307B 85 01 STA #01
. 307D 85 01 STA #01
. 307F 85 01 STA #01
. 3081 85 01 STA #01
. 3083 85 01 STA #01
. 3085 85 01 STA #01
. 3087 85 01 STA #01
. 3089 85 01 STA #01
. 308B 85 01 STA #01
. 308D 85 01 STA #01
. 308F 85 01 STA #01
. 3091 85 01 STA #01
. 3093 85 01 STA #01
. 3095 85 01 STA #01
. 3097 85 01 STA #01
. 3099 85 01 STA #01
. 309B 85 01 STA #01
. 309D 85 01 STA #01
. 309F 85 01 STA #01
. 30A1 85 01 STA #01
. 30A3 85 01 STA #01
. 30A5 85 01 STA #01
. 30A7 85 01 STA #01
. 30A9 85 01 STA #01
. 30AB 85 01 STA #01
. 30AD 85 01 STA #01
. 30AF 85 01 STA #01
. 30B1 85 01 STA #01
. 30B3 85 01 STA #01
. 30B5 85 01 STA #01
. 30B7 85 01 STA #01
. 30B9 85 01 STA #01
. 30BB 85 01 STA #01
. 30BD 85 01 STA #01
. 30BF 85 01 STA #01
. 30C1 85 01 STA #01
. 30C3 85 01 STA #01
. 30C5 85 01 STA #01
. 30C7 85 01 STA #01
. 30C9 85 01 STA #01
. 30CB 85 01 STA #01
. 30CD 85 01 STA #01
. 30CF 85 01 STA #01
. 30D1 85 01 STA #01
. 30D3 85 01 STA #01
. 30D5 85 01 STA #01
. 30D7 85 01 STA #01
. 30D9 85 01 STA #01
. 30DB 85 01 STA #01
. 30DD 85 01 STA #01
. 30DF 85 01 STA #01
. 30E1 85 01 STA #01
. 30E3 85 01 STA #01
. 30E5 85 01 STA #01
. 30E7 85 01 STA #01
. 30E9 85 01 STA #01
. 30EB 85 01 STA #01
. 30ED 85 01 STA #01
. 30EF 85 01 STA #01
. 30F1 85 01 STA #01
. 30F3 85 01 STA #01
. 30F5 85 01 STA #01
. 30F7 85 01 STA #01
. 30F9 85 01 STA #01
. 30FB 85 01 STA #01
. 30FD 85 01 STA #01
. 30FF 85 01 STA #01
. 3101 85 01 STA #01
. 3103 85 01 STA #01
. 3105 85 01 STA #01
. 3107 85 01 STA #01
. 3109 85 01 STA #01
. 310B 85 01 STA #01
. 310D 85 01 STA #01
. 310F 85 01 STA #01
. 3111 85 01 STA #01
. 3113 85 01 STA #01
. 3115 85 01 STA #01
. 3117 85 01 STA #01
. 3119 85 01 STA #01
. 311B 85 01 STA #01
. 311D 85 01 STA #01
. 311F 85 01 STA #01
. 3121 85 01 STA #01
. 3123 85 01 STA #01
. 3125 85 01 STA #01
. 3127 85 01 STA #01
. 3129 85 01 STA #01
. 312B 85 01 STA #01
. 312D 85 01 STA #01
. 312F 85 01 STA #01
. 3131 85 01 STA #01
. 3133 85 01 STA #01
. 3135 85 01 STA #01
. 3137 85 01 STA #01
. 3139 85 01 STA #01
. 313B 85 01 STA #01
. 313D 85 01 STA #01
. 313F 85 01 STA #01
. 3141 85 01 STA #01
. 3143 85 01 STA #01
. 3145 85 01 STA #01
. 3147 85 01 STA #01
. 3149 85 01 STA #01
. 314B 85 01 STA #01
. 314D 85 01 STA #01
. 314F 85 01 STA #01
. 3151 85 01 STA #01
. 3153 85 01 STA #01
. 3155 85 01 STA #01
. 3157 85 01 STA #01
. 3159 85 01 STA #01
. 315B 85 01 STA #01
. 315D 85 01 STA #01
. 315F 85 01 STA #01
. 3161 85 01 STA #01
. 3163 85 01 STA #01
. 3165 85 01 STA #01
. 3167 85 01 STA #01
. 3169 85 01 STA #01
. 316B 85 01 STA #01
. 316D 85 01 STA #01
. 316F 85 01 STA #01
. 3171 85 01 STA #01
. 3173 85 01 STA #01
. 3175 85 01 STA #01
. 3177 85 01 STA #01
. 3179 85 01 STA #01
. 317B 85 01 STA #01
. 317D 85 01 STA #01
. 317F 85 01 STA #01
. 3181 85 01 STA #01
. 3183 85 01 STA #01
. 3185 85 01 STA #01
. 3187 85 01 STA #01
. 3189 85 01 STA #01
. 318B 85 01 STA #01
. 318D 85 01 STA #01
. 318F 85 01 STA #01
. 3191 85 01 STA #01
. 3193 85 01 STA #01
. 3195 85 01 STA #01
. 3197 85 01 STA #01
. 3199 85 01 STA #01
. 319B 85 01 STA #01
. 319D 85 01 STA #01
. 319F 85 01 STA #01
. 31A1 85 01 STA #01
. 31A3 85 01 STA #01
. 31A5 85 01 STA #01
. 31A7 85 01 STA #01
. 31A9 85 01 STA #01
. 31AB 85 01 STA #01
. 31AD 85 01 STA #01
. 31AF 85 01 STA #01
. 31B1 85 01 STA #01
. 31B3 85 01 STA #01
. 31B5 85 01 STA #01
. 31B7 85 01 STA #01
. 31B9 85 01 STA #01
. 31BB 85 01 STA #01
. 31BD 85 01 STA #01
. 31BF 85 01 STA #01
. 31C1 85 01 STA #01
. 31C3 85 01 STA #01
. 31C5 85 01 STA #01
. 31C7 85 01 STA #01
. 31C9 85 01 STA #01
. 31CB 85 01 STA #01
. 31CD 85 01 STA #01
. 31CF 85 01 STA #01
. 31D1 85 01 STA #01
. 31D3 85 01 STA #01
. 31D5 85 01 STA #01
. 31D7 85 01 STA #01
. 31D9 85 01 STA #01
. 31DB 85 01 STA #01
. 31DD 85 01 STA #01
. 31DF 85 01 STA #01
. 31E1 85 01 STA #01
. 31E3 85 01 STA #01
. 31E5 85 01 STA #01
. 31E7 85 01 STA #01
. 31E9 85 01 STA #01
. 31EB 85 01 STA #01
. 31ED 85 01 STA #01
. 31EF 85 01 STA #01
. 31F1 85 01 STA #01
. 31F3 85 01 STA #01
. 31F5 85 01 STA #01
. 31F7 85 01 STA #01
. 31F9 85 01 STA #01
. 31FB 85 01 STA #01
. 31FD 85 01 STA #01
. 31FF 85 01 STA #01
. 3201 85 01 STA #01
. 3203 85 01 STA #01
. 3205 85 01 STA #01
. 3207 85 01 STA #01
. 3209 85 01 STA #01
. 320B 85 01 STA #01
. 320D 85 01 STA #01
. 320F 85 01 STA #01
. 3211 85 01 STA #01
. 3213 85 01 STA #01
. 3215 85 01 STA #01
. 3217 85 01 STA #01
. 3219 85 01 STA #01
. 321B 85 01 STA #01
. 321D 85 01 STA #01
. 321F 85 01 STA #01
. 3221 85 01 STA #01
. 3223 85 01 STA #01
. 3225 85 01 STA #01
. 3227 85 01 STA #01
. 3229 85 01 STA #01
. 322B 85 01 STA #01
. 322D 85 01 STA #01
. 322F 85 01 STA #01
. 3231 85 01 STA #01
. 3233 85 01 STA #01
. 3235 85 01 STA #01
. 3237 85 01 STA #01
. 3239 85 01 STA #01
. 323B 85 01 STA #01
. 323D 85 01 STA #01
. 323F 85 01 STA #01
. 3241 85 01 STA #01
. 3243 85 01 STA #01
. 3245 85 01 STA #01
. 3247 85 01 STA #01
. 3249 85 01 STA #01
. 324B 85 01 STA #01
. 324D 85 01 STA #01
. 324F 85 01 STA #01
. 3251 85 01 STA #01
. 3253 85 01 STA #01
. 3255 85 01 STA #01
. 3257 85 01 STA #01
. 3259 85 01 STA #01
. 325B 85 01 STA #01
. 325D 85 01 STA #01
. 325F 85 01 STA #01
. 3261 85 01 STA #01
. 3263 85 01 STA #01
. 3265 85 01 STA #01
. 3267 85 01 STA #01
. 3269 85 01 STA #01
. 326B 85 01 STA #01
. 326D 85 01 STA #01
. 326F 85 01 STA #01
. 3271 85 01 STA #01
. 3273 85 01 STA #01
. 3275 85 01 STA #01
. 3277 85 01 STA #01
. 3279 85 01 STA #01
. 327B 85 01 STA #01
. 327D 85 01 STA #01
. 327F 85 01 STA #01
. 3281 85 01 STA #01
. 3283 85 01 STA #01
. 3285 85 01 STA #01
. 3287 85 01 STA #01
. 3289 85 01 STA #01
. 328B 85 01 STA #01
. 328D 85 01 STA #01
. 328F 85 01 STA #01
. 3291 85 01 STA #01
. 3293 85 01 STA #01
. 3295 85 01 STA #01
. 3297 85 01 STA #01
. 3299 85 01 STA #01
. 329B 85 01 STA #01
. 329D 85 01 STA #01
. 329F 85 01 STA #01
. 3301 85 01 STA #01
. 3303 85 01 STA #01
. 3305 85 01 STA #01
. 3307 85 01 STA #01
. 3309 85 01 STA #01
. 330B 85 01 STA #01
. 330D 85 01 STA #01
. 330F 85 01 STA #01
. 3311 85 01 STA #01
. 3313 85 01 STA #01
. 3315 85 01 STA #01
. 3317 85 01 STA #01
. 3319 85 01 STA #01
. 331B 85 01 STA #01
. 331D 85 01 STA #01
. 331F 85 01 STA #01
. 3321 85 01 STA #01
. 3323 85 01 STA #01
. 3325 85 01 STA #01
. 3327 85 01 STA #01
. 3329 85 01 STA #01
. 332B 85 01 STA #01
. 332D 85 01 STA #01
. 332F 85 01 STA #01
. 3331 85 01 STA #01
. 3333 85 01 STA #01
. 3335 85 01 STA #01
. 3337 85 01 STA #01
. 3339 85 01 STA #01
. 333B 85 01 STA #01
. 333D 85 01 STA #01
. 333F 85 01 STA #01
. 3341 85 01 STA #01
. 3343 85 01 STA #01
. 3345 85 01 STA #01
. 3347 85 01 STA #01
. 3349 85 01 STA #01
. 334B 85 01 STA #01
. 334D 85 01 STA #01
. 334F 85 01 STA #01
. 3351 85 01 STA #01
. 3353 85 01 STA #01
. 3355 85 01 STA #01
. 3357 85 01 STA #01
. 3359 85 01 STA #01
. 335B 85 01 STA #01
. 335D 85 01 STA #01
. 335F 85 01 STA #01
. 3361 85 01 STA #01
. 3363 85 01 STA #01
. 3365 85 01 STA #01
. 3367 85 01 STA #01
. 3369 85 01 STA #01
. 336B 85 01 STA #01
. 336D 85 01 STA #01
. 336F 85 01 STA #01
. 3371 85 01 STA #01
. 3373 85 01 STA #01
. 3375 85 01 STA #01
. 3377 85 01 STA #01
. 3379 85 01 STA #01
. 337B 85 01 STA #01
. 337D 85 01 STA #01
. 337F 85 01 STA #01
. 3381 85 01 STA #01
. 3383 85 01 STA #01
. 3385 85 01 STA #01
. 3387 85 01 STA #01
. 3389 85 01 STA #01
. 338B 85 01 STA #01
. 338D 85 01 STA #01
. 338F 85 01 STA #01
. 3391 85 01 STA #01
. 3393 85 01 STA #01
. 3395 85 01 STA #01
. 3397 85 01 STA #01
. 3399 85 01 STA #01
. 339B 85 01 STA #01
. 339D 85 01 STA #01
. 339F 85 01 STA #01
. 3401 85 01 STA #01
. 3403 85 01 STA #01
. 3405 85 01 STA #01
. 3407 85 01 STA #01
. 3409 85 01 STA #01
. 340B 85 01 STA #01
. 340D 85 01 STA #01
. 340F 85 01 STA #01
. 3411 85 01 STA #01
. 3413 85 01 STA #01
. 3415 85 01 STA #01
. 3417 85 01 STA #01
. 3419 85 01 STA #01
. 341B 85 01 STA #01
. 341D 85 01 STA #01
. 341F 85 01 STA #01
. 3421 85 01 STA #01
. 3423 85 01 STA #01
. 3425 85 01 STA #01
. 3427 85 01 STA #01
. 3429 85 01 STA #01
. 342B 85 01 STA #01
. 342D 85 01 STA #01
. 342F 85 01 STA #01
. 3431 85 01 STA #01
. 3433 85 01 STA #01
. 3435 85 01 STA #01
. 3437 85 01 STA #01
. 3439 85 01 STA #01
. 343B 85 01 STA #01
. 343D 85 01 STA #01
. 343F 85 01 STA #01
. 3441 85 01 STA #01
. 3443 85 01 STA #01
. 3445 85 01 STA #01
. 3447 85 01 STA #01
. 3449 85 01 STA #01
. 344B 85 01 STA #01
. 344D 85 01 STA #01
. 344F 85 01 STA #01
. 3451 85 01 STA #01
. 3453 85 01 STA #01
. 3455 85 01 STA #01
. 3457 85 01 STA #01
. 3459 85 01 STA #01
. 345B 85 01 STA #01
. 345D 85 01 STA #01
. 345F 85 01 STA #01
. 3461 85 01 STA #01
. 3463 85 01 STA #01
. 3465 85 01 STA #01
. 3467 85 01 STA #01
. 3469 85 01 STA #01
. 346B 85 01 STA #01
. 346D 85 01 STA #01
. 346F 85 01 STA #01
. 3471 85 01 STA #01
. 3473 85 01 STA #01
. 3475 85 01 STA #01
. 3477 85 01 STA #01
. 3479 85 01 STA #01
. 347B 85 01 STA #01
. 347D 85 01 STA #01
. 347F 85 01 STA #01
. 3481 85 01 STA #01
. 3483 85 01 STA #01
. 3485 85 01 STA #01
. 3487 85 01 STA #01
. 3489 85 01 STA #01
. 348B 85 01 STA #01
. 348D 85 01 STA #01
. 348F 85 01 STA #01
. 3491 85 01 STA #01
. 3493 85 01 STA #01
. 3495 85 01 STA #01
. 3497 85 01 STA #01
. 3499 85 01 STA #01
. 349B 85 01 STA #01
. 349D 85 01 STA #01
. 349F 85 01 STA #01
. 3501 85 01 STA #01
. 3503 85 01 STA #01
. 3505 85 01 STA #01
. 3507 85 01 STA #01
. 3509 85 01 STA #01
. 350B 85 01 STA #01
. 350D 85 01 STA #01
. 350F 85 01 STA #01
. 3511 85 01 STA #01
. 3513 85 01 STA #01
. 3515 85 01 STA #01
. 3517 85 01 STA #01
. 3519 85 01 STA #01
. 351B 85 01 STA #01
. 351D 85 01 STA #01
. 351F 85 01 STA #01
. 3521 85 01 STA #01
. 3523 85 01 STA #01
. 3525 85 01 STA #01
. 3527 85 01 STA #01
. 3529 85 01 STA #01
. 352B 85 01 STA #01
. 352D 85 01 STA #01
. 352F 85 01 STA #01
. 3531 85 01 STA #01
. 3533 85 01 STA #01
. 3535 85 01 STA #01
. 3537 85 01 STA #01
. 3539 85 01 STA #01
. 353B 85 01 STA #01
. 353D 85 01 STA #01
. 353F 85 01 STA #01
. 3541 85 01 STA #01
. 3543 85 01 STA #01
. 3545 85 01 STA #01
. 3547 85 01 STA #01
. 3549 85 01 STA #01
. 354B 85 01 STA #01
. 354D 85 01 STA #01
. 354F 85 01 STA #01
. 3551 85 01 STA #01
. 3553 85 01 STA #01
. 3555 85 01 STA #01
. 3557 85 01 STA #01
. 3559 85 01 STA #01
. 355B 85 01 STA #01
. 355D 85 01 STA #01
. 355F 85 01 STA #01
. 3561 85 01 STA #01
. 3563 85 01 STA #01
. 3565 85 01 STA #01
. 3567 85 01 STA #01
. 3569 85 01 STA #01
. 356B 85 01 STA #01
. 356D 85 01 STA #01
. 356F 85 01 STA #01
. 3571 85 01 STA #01
. 3573 85 01 STA #01
. 3575 85 01 STA #01
. 3577 85 01 STA #01
. 3579 85 01 STA #01
. 357B 85 01 STA #01
. 357D 85 01 STA #01
. 357F 85 01 STA #01
. 3581 85 01 STA #01
. 3583 85 01 STA #01
. 3585 85 01 STA #01
. 3587 85 01 STA #01
. 3589 85 01 STA #01
. 358B 85 01 STA #01
. 358D 85 01 STA #01
. 358F 85 01 STA #01
. 3591 85 01 STA #01
. 3593 85 01 STA #01
. 3595 85 01 STA #01
. 3597 85 01 STA #01
. 3599 85 01 STA #01
. 359B 85 01 STA #01
. 359D 85 01 STA #01
. 359F 85 01 STA #01
. 3601 85 01 STA #01
. 3603 85 01 STA #01
. 3605 85 01 STA #01
. 3607 85 01 STA #01
. 3609 85 01 STA #01
. 360B 85 01 STA #01
. 360D 85 01 STA #01
. 360F 85 01 STA #01
. 3611 85 01 STA #01
. 3613 85 01 STA #01
. 3615 85 01 STA #01
. 3617 85 01 STA #01
. 3619 85 01 STA #01
. 361B 85 01 STA #01
. 361D 85 01 STA #01
. 361F 85 01 STA #01
. 3621 85 01 STA #01
. 3623 85 01 STA #01
. 3625 85 01 STA #01
. 3627 85 01 STA #01
. 3629 85 01 STA #01
. 362B 85 01 STA #01
. 362D 85 01 STA #01
. 362F 85 01 STA #01
. 3631 85 01 STA #01
. 3633 85 01 STA #01
. 3635 85 01 STA #01
. 3637 85 01 STA #01
. 3639 85 01 STA #01
. 363B 85 01 STA #01
. 363D 85 01 STA #01
. 363F 85 01 STA #01
. 3641 85 01 STA #01
. 3643 85 01 STA #01
. 3645 85 01 STA #01
. 3647 85 01 STA #01
. 3649 85 01 STA #01
. 364B 85 01 STA #01
. 364D 85 01 STA #01
. 364F 85 01 STA #01
. 3651 85 01 STA #01
. 3653 85 01 STA #01
. 3655 85 01 STA #01
. 3657 85 01 STA #01
. 3659 85 01 STA #01
. 365B 85 01 STA #01
. 365D 85 01 STA #01
. 365F 85 01 STA #01
. 3661 85 01 STA #01
. 3663 85 01 STA #01
. 3665 85 01 STA #01
. 3667 85 01 STA #01
. 3669 85 01 STA #01
. 366B 85 01 STA #01
. 366D 85 01 STA #01
. 366F 85 01 STA #01
. 3671 85 01 STA #01
. 3673 85 01 STA #01
. 3675 85 01 STA #01
. 3677 85 01 STA #01
. 3679 85 01 STA #01
. 367B 85 01 STA #01
. 367D 85 01 STA #01
. 367F 85 01 STA #01
. 3681 85 01 STA #01
. 3683 85 01 STA #01
. 3685 85 01 STA #01
. 3687 85 01 STA #01
. 3689 85 01 STA #01
. 368B 85 01 STA #01
. 368D 85 01 STA
```



ram átvételének lehetőségét nem garantálja. Célunk nem is a — sajnos, nagymértékben elterjedt — tisztességtelen programmásolás elősegítése volt, hanem elsősorban az, hogy saját készítésű, saját hardverrel előállított programjainkat könnyen és biztonságosan átadhassuk.

A programot két formában adjuk meg. A hexdump (1. lista) a legegyszerűbb lehetőség a tárba beírásához, az assembly nyelvű (2. és 3. lista) pedig jó eszköz azoknak, akik a program működésébe is bele kívánának látni, esetleg azt saját igényüknek és ötleteiknek megfelelően át akarják alakítani.

Az assembly nyelvű listából helytakarékossági szempontok miatt kihagytuk a képernyőn megjelenő, a program kezeléséhez útmutató adó szöveget. Ez egyébként még

a program beírása előtt is elolvasható a hexdump lista mellett, a tartalmat karakteres formában tartalmazó ábráról, a 302E és a 3082 címek között.

A program első része — ez a 2. listán látható — először a már említett szöveget írja ki a képernyőre az FFD2 címen kezdődő KERNAL rutinnal, majd beállítja a kazettás egységet. Ezt a billentyűzetfigyelés követi, amely STOP-nál a BASIC-be visszatérést, C-nél a 3098-as címen kezdődő másoló alprogramra ugrást eredményezi.

A másoló programrész (3. lista) először kikapcsolja a képernyőt, beállítja az adatirányt, bekapcsolja a kazettás egység motorját, majd a 30E0 címen kezdődő — a forráskazettáról egy bitet beolvasó — szubrutinra ugrik.

A bit olvasása után annak a soros buszra kivitele megy végbe a 30EE címen kezdődő szubrutinnal (a 30AE—30B3 rész a képernyő keretét „csikozza”).

A program következő része (30BC—30C5) a CTRL billentyűt figyeli. Ha ez nincs lenyomva, a másolás folytatódik. A CTRL lenyomása a másolást megszakítja: a 30C6-tól kezdődő részellet megszünteti a képernyő elsötétítését, kikapcsolja a kazettamotort, a keret kékre állítja, és visszatér a híváshoz a 3017-es címre. Ezután a képernyőre írás stb. következik a már ismertetett módon.

A program MONITOR módban írható a tárba. Beírás után kazettára menthető (3000-től 30BF-ig). Indítása: G3000.

Jász Tibor—Boros Tibor—Molnár István

HOVA TOVÁBB?!

KÖZÉPFOKÚ OKTATÁSI INTÉZMÉNYEK

KÖZGAZDASÁGI SZAKKÖZÉPISKOLÁK

Programozó

Közgazdasági Szakközépiskola
Budapest, Kerepesi út 124. 1145

Folyamatszervező

Fáy András Szakközépiskola
Miskolc, Jászi Oszkár u. 1. 3529

Hunyadi Mátyás Szakközépiskola
Székesfehérvár,
Népköztársaság u. 32. 8000

Lengyel Gyula Szakközépiskola
Győr, Dimitrov sétány 32. 9022

Noszlopy Gáspár Szakközépiskola
Kaposvár, Lenin u. 2. 7401

Kada Elek Szakközépiskola
Kecskemét, Katona József tér 1. 6000

Vásárhelyi Pál Szakközépiskola
Szolnok,
Tanácsköztársaság u. 7. 5000

Programozó és folyamatszervező

Sebes György Szakközépiskola
Békéscsaba,
Tanácsköztársaság u. 1. 5600

Alpári Gyula Szakközépiskola
Eger, Klapka u. 7. 3300

Csányi László Szakközépiskola
Zalaegerszeg, Jókai u. 6. 8901

Hámán Kató Szakközépiskola
Budapest, Jurányi u. 1. 1027

MŰSZAKI SZAKKÖZÉPISKOLÁK

Számítástechnikai műszerész

623. Sz. Vágó Béla Szakközépiskola
Kecskemét, Bethlenváros 15. 6000

Latinca Sándor Szakközépiskola
Budapest, Kossuth tér 12. 1191

Ságvári Endre Szakközépiskola
Székesfehérvár,
Vörös Hadsereg u. 45. 8000

Számítástechnikai műszerész

Információ- és számítástechnikai
technikus

Gépgyártástechnológia-
számítástechnikai technikus

Landler Jenő Szakközépiskola
Budapest, Dózsa György u. 26. 1041

Alkalmazott számítástechnika szak

Energetikai Szakképzési Intézet
Paks, Dózsa György u. 95. 7031

FELSŐOKTATÁSI INTÉZMÉNYEK

Eötvös Loránd Tudományegyetem
Természettudományi Kar

Matematika—számítástechnika
tanár, programozó matematikus, programtervező matematikus

József Attila Tudományegyetem
Természettudományi Kar

Szeged, Aradi vértanúk tere 1. 6720

Programozó matematikus,
programtervező matematikus,
közgazdasági programtervező
matematikus

Kossuth Lajos Tudományegyetem
Természettudományi Kar
Debrecen, Egyetem tér 1. 4010

Matematika-, ábrázológeometria-
és számítástechnikanómer,
programozó matematikus

Budapesti Műszaki Egyetem,
Villamosmérnöki Kar

Budapest, Egy József u. 18.
Műszer- és irányítástechnika szak,
informatika szak

Nehézipari Műszaki Egyetem
Kohó- és Fémpipari Főiskolai Kar

Dunaújváros,
Táncsics Mihály u. 1. 2400

Szervezési szak, számítástechnikai
szakirány, középiskolai műszaki
tanári szak, informatikai szakirány

Bánki Donát Gépipari
Műszaki Főiskola

Budapest, Népszínház u. 8. 1081

Szervező és informatika szak

Gépipari és Automatizálási
Műszaki Főiskola

Kecskemét, Izsáki út 1. 6001

Gépipari-automatizálási szak,
számítástechnikai ágazat

Kandó Kálmán Villamosipari
Műszaki Főiskola

Budapest, Tavaszmező u. 15—17. 1084

Műszeripari és automatizálási szak,
informatikai szak,
számítástechnikai eszközök szak

Pollack Mihály Műszaki Főiskola
Pécs, Boszorkány u. 2. 7624

Műszaki informatikai szak

Széchenyi István Közlekedési
és Távközlési Műszaki Főiskola

Győr, Ságvári Endre u. 3. 9026

Vasútiüzemi szak,
számítástechnikai ágazat

Azonnali szállítással kínáljuk az alábbi számítástechnikai eszközöket:

1. IBM PC terminál

- 640 kbájt RAM
- 8 MHz CPU
- 360 kbájt floppy
- 83 gombos tasztatúra
- mono monitor + kártya

Ár: 94 800,— Ft + áfa

2. IBM XT kompatibilis számítógép

- 8 MHz turbó
- 640 kbájt RAM
- 360 kbájt floppy
- 27 Mbájt winchester (ST 225 Seagate)
- mono monitor + kártya
- 83 gombos tasztatúra

Ár: 138 000,— Ft + áfa

Ue.: színesben 165 000,— Ft + áfa

3. IBM AT kompatibilis számítógép

- 80 286 CPU 8-10-12 MHz órajel
- 1 Mbájt RAM
- 1,2 Mbájt floppy
- 27 Mbájt winchester
- mono monitor + kártya
- 83 gombos tasztatúra

Ár: 205 000,— Ft + áfa

Ue.: színesben 232 000,— Ft + áfa

Ue.: 40 Mbájtos winchester diszkkal:
260 000,— Ft + áfa

4. 32 bites AT kompatibilis számítógép

- 80 386 CPU 20 MHz órajel
- 2 Mbájt RAM
- 40 Mbájt winchester
- 1,2 Mbájt floppy
- mono monitor + kártya

Ára: 550 000,— Ft + áfa (1 év garanciával) digitális állódobozban.

Ue.: színesben 570 000,— Ft + áfa

EGA monitorral ue.: 600 000,— Ft + áfa

Egyéb tartozékok, perifériák:

EPSON FX 1000 printer	72 000,— Ft + áfa
40 Mbájtos	
Archív Streamer (belső)	96 000,— Ft + áfa
SUMMASKETCH	
digitalizáló 300 × 300	144 000,— Ft + áfa
EGA monitor	76 000,— Ft + áfa

Hálózati elemek

– ARCNET kártya	24 000,— Ft + áfa
– Aktív HUB	48 000,— Ft + áfa
– Ethernet kártya	48 000,— Ft + áfa

Garancia a gépek árának 10%-a.

Szervizünk számítógépek javításával, átalánydíjas karbantartási szerződéssel, videokészülékek áthangolásával, javításával áll ügyfeleink rendelkezésére.

DÉVA KISSZÖVETKEZET

Üzlet: Bp. VIII., Pogány J. u. 9.

Tel.: 139-621, 135-601

Szervizműhely: 133-017

FELADATOK — MEGOLDÁSOK

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

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

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

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

8. feladat:

Tizenötös játék 2.

Írjon programot, amellyel kirakható a tizenötös játék!

Megoldás

A mostani feladat az előző számban közölt, Beta BASIC-ben ZX-Spectrumra megírt program továbbfejlesztésével oldható meg legkönnyebben. Azok kedvéért, akik az előző számot nem olvasták, tekintünk át röviden megint.

A tábla belső ábrázolását t(5,4,4) tömb segítségével értük el. A tömb elemei a számok egytől tizenötig a táblán lévő számoknak megfelelően, és az üres mezőt képviselő 0. Mivel az üres mező helye különös jelentőségű, koordinátáit tx és ty tartalmazza.

A tizenötös játék tábláját a kép nevű szubrutin (1000—1080) rajzolja ki. A táblázat megfelelő értékekkel való feltöltését és a mezők kezdeti kirajzolását az alap szubrutin (1400—1510) látja el.

Az egyes mezők kirajzolását minden esetben a mező szubrutin (1200—1270) segítségével kell elvégezni.

A tábla kirajzolása után a főprogram (6000—6140) egy végtelen ciklusban figyeli a billentyűzetten beadott parancsokat. A kurzormozgató nyílakkal vagy a botkormánnyal lehet a mezőket a táblán tologatni (6050—6080). A tényleges mozgatót a le (2000), fel (2100), a jobbra (2200) és balra (2300) rutinok végzik el, mint ahogy a t(5) megfelelő változtatását is. A tábla összekeverése a 0 vagy a tüzgomb lenyomásával kezdeményezhető. Ekkor a kever szubrutin (3000—3180) indul el, és újabb billentyű lenyomásáig véletlenszerűen tologatja a mezőket.

A múltkori program még nem tartalmazta a <space> parancsot. Ennek a játék kirakását kell kezdeményeznie.

Vizsgáljuk meg először, hogy az ember hogyan kezd hozzá a tizenötös játék kirakásához! A legtöbb ember először megkeresi az egyes számú mezőt, és azt próbálja a helyére vinni. Utána megkeresi a kettes mezőt, és azt is a helyére továbbítja. Eközben a többi mező helyzetével keveset törődik. Az is gyakran előfordul, hogy míg az egyest a bal felső sarokba továbbítja, a tábla más rendezett részeit teljesen elrontja.

Ennek a logikának a végig gondolása nyilvánvalóvá teszi, hogy a számítógépnek is csak a következő szám helyzetével kell törődnie. Ha az üres mező

helyét rögzítettnek vesszük, akkor a következő szám helye rakásának menetét egyértelműen meghatározza a szám helyzete. Így egy szám helye rakásához a gépnek maximum tizenötöt helyerakási algoritmusra van szüksége. Ez összesen legfeljebb 125, ami nem olyan sok; a számítógép a DATA sorokban leírt mozgatósi sorrenddel segítségével könnyen „megtanulhatja” őket.

Feltétel volt, hogy az üres mező helyének rögzítettnek kell lennie. Ez legegyszerűbben úgy valósítható meg, hogy a tábla kirakásának elkezdése előtt az üres mezőt egy ismert helyre — például a tábla jobb alsó sarkába — mozgatjuk, majd a továbbiakban minden szám helye rakása után ugyanoda visszük, függetlenül attól, hogy az adott számot hogyan vittük a helyére. Ily módon a következő szám helye rakása előtt az üres mező jól meghatározott helyre kerül.

Az eddig elmondottak alapján a program már megírható. A továbbiakban már csak a finomítással kell törődnünk.

Az így megírt program ugyan megoldás, de sok oda-vissza mozgatót tartalmaz. Sok esetben lehetőség van két egymást követő szám helye rakása között akár több mozdtítás összevonására is. Ehhez szükséges, hogy a DATA sorból kiolvasott mozgatósi sorrendet ne azonnal, hanem csak

a következő DATA sor adatainak megvizsgálása után rajzoljuk ki a képernyőre. Ekkor viszont a következő szám helyének meghatározása sem lesz olyan egyszerű, hiszen ezek a „késleltetett” mozgatók lehet, hogy megváltoztatnák a következő szám helyzetét is. Ebből látszik, hogy a hatékonyabb program megírásának ára van.

A program másik finomításának jelentősége csak akkor válik igazán érthetővé, ha az olvasó megpróbálja hibátlanul begépelni a DATA sorokat. Nyilván felmerül a kérdés: nem lehet ezek létrehozására egy hatékonyabb, megbízhatóbb eljárás generálni?

Természetesen lehet, s egy példa a mellékelt programban is látható. Ez a <1>-re meghatározza, hogy melyik DATA sort kell létrehozni, és ebbe a sorba a <t> után következő mozgásokkal megfelelően kódolva beleteszi. A „tanulás” végét <v>-vel kell jelezni. Bár ezzel a módszerrel a hibákat nem lehet teljes mértékben kiküszöbölni, az, hogy a bevitt mozgások a képernyőn követhetők, a megbízhatóságot lényegesen javítja.

Vizsgáljuk meg most részletesen a kirak rutint (5000). Az 5040—5160-as sorok között található a számok kirakását végző ciklus. Az n=0 esetén az üres mezőt rakjuk helyre. A ciklus csak 13-ig tart, mert a tizenhárom szám helye rakása után a tábla már biztosan ki van rakva.

Két szám helye rakása között az e\$ teszi lehetővé az oda-vissza mozgatók kiküszöbölését. Az 5050-es sorban a keres rutint hívjuk meg, amely az m\$-ban az n. mező helye rakásához szükséges mozgatót tartalmazza. Amíg a mozgató az e\$-ban tárolt pont ellentétes, semmi nem történik (5090). Ha nem így van, akkor az e\$ fennmaradó elemeinek megfelelő mozgató történik (5100—5120). Az m\$-ban megmaradt elemek átkerülnek e\$-ba, és a következő ciklus-

TIZENÖTÖS JÁTÉK			
2	12	9	6
3	15	1	11
13		5	14
10	8	7	4

5-BALRA
6-FEL
7-LE
8-JOBBRA
0-KEVERD
ÖSSZE
SPACE-
RAKD KI

ban ezek összehasonlíthatóságát vizsgáljuk a következő számjegy mozgásával.

Mivel így a mozgítás mindig egy ciklussal lemarad, szükségese az 5180—5200-as sorok, amelyek a legutolsó lépést is elvégzik.

A keres szubrutin feladata a megfelelő DATA sor kiválasz-

tása. Ehhez először az n\$-ben létrehozható a tényleges állás —, ha ha nem törekednénk egymást követő lépések összehasonlására, t\$ lenne —, majd megkeresi az n-et a táblázatban, végül pedig (4170) kiválasztja az ehhez a számhoz és koordinátákhoz tartozó sort. A tényleges állás létrehozásá-

hoz szükség van olyan mozgató rutinokra, amelyek nem rajzolnak ki a képernyőre. Ilyenek a kle (4200), kfel (4300), kjobbra (4400) és a kbalra (4500) rutinok.

A mostani elég nehéz feladat után biztosan pihentető lesz a következő rajzolóprogram.

9. feladat:

Rekurzív rajz

Írjon programot, amely beolvasson egy számot és kiszámolja az ennek megfelelő fraktálján rekurzív rajzát.

Pintér Gábor

```

10 REM Tizenötös játék
1100 1988.06.29.
1110 Pinter Gabor

20 RUN 5000
30 CLEAR
40 PRINT "NYELV ERSE"
50 LET (T=PEEK 23730+256+PEEK 2373)
60 SHAPE "IS V1.0" L:W=40
70 PAPER PEEK 23631+256+PEEK 23632+256
80 PRINT "Cseta 3.1"CODE (T+3,85536)
90 SETUP
40 BORDER 7
50 PAPER 7
60 DIM C$(10)
70 CLEAR C$
80 LOC "Cseta 3.1"CODE
90 CLS
100 CLEAR:USR 50419
50 LIST FORMAT 1
60 CSIZE 7.0

1000 DEF PROC lep
1010 REM A lep nem valtozo
1020 részlet fajzoja ki.

1020 LOCAL i
1030 PRINT CSIZE 0.1;AT 3.20;"TIZE"
1040 MOTORS:TRAB 24;"TAVO" TRAB 23;"S-BALAS" TRAB 20;"0-FEL" TRAB 19;"0-LE" TRAB 18;"0-JOBBRA" TRAB 17;"0-NEVENE" TRAB 16;"0-SSIZE" TRAB 15;"0-SPRICE" TRAB 14;"0-PAK" TRAB 13;"0"
1050 CSIZE 0
1060 PRINT AT 0,0
1070 PRINT "1 TO 2"
1080 NEXT i
1090 END PROC

1200 DEF PROC bezo k,v
1210 REM k:ix,v:bezoit
1220 rajz:0:esq.

1230 LOCAL i
1240 LET i=3*TRA CODE (T+1);V
1250 IF CODE (ix,v) THEN
1260 FLPT PAPER 6,40x=32,190-40v
1270 CSIZE 0.15
1280 PAPER 6,40x=16-4xLEN s$
1290 PRINT AT 0,0
1300 FLPT PAPER 5,40x=32,190-40v s$
1310 CSIZE 0
1320 PAPER 7
1330 END PROC

1400 DEF PROC a99
1410 REM a:hezsoal:lapot
1420 beallitasa.

1430 LOCAL i
1440 DIM t$(4,4)
1450 REM t:labla.
1460 LET t$(0,0)=0
1470 REM t:uzes:bezo helve.
1480 FOR j=1 TO 4
1490 IF t$(j,0) THEN
1500 ELSE
1510 LET t$(j,1)=CHRS (44+j)+4
1520 NEXT j
1530 END PROC

2000 DEF PROC le
2010 REM Mozditas lefele.

2020 BEEP .02,30
2030 LET t$(ix,v)=t$(ix,(v-1))
2040 LET t$(ix,(v-1))=CHRS 0
2050 LET ix,(v-1)
2060 END PROC

3100 DEF PROC fel
3110 REM Mozditas felfele.

3120 BEEP .02,30
3130 LET t$(ix,v)=t$(ix,(v+1))
3140 LET t$(ix,(v+1))=CHRS 0
3150 LET ix,(v+1)
3160 END PROC

4100 DEF PROC jobbra
4110 REM Mozditas jobbra.

4120 BEEP .02,17
4130 IF t$(ix,v) THEN
4140 LET t$(ix,v)=t$(ix,(v+1))
4150 LET t$(ix,(v+1))=CHRS 0
4160 LET ix,(v+1)
4170 END PROC

4200 DEF PROC balra
4210 REM Mozditas balra.

4220 BEEP .02,20
4230 IF t$(ix,v) THEN
4240 LET t$(ix,v)=t$(ix,(v-1))
4250 LET t$(ix,(v-1))=CHRS 0
4260 LET ix,(v-1)
4270 END PROC

3000 DEF PROC lever
3010 REM A játék megkezdése.

3020 LOCAL i,r,h,s,i$
3030 LET r=1
3040 LET h=1204
3050 LET h$(1204)=r
3060 IF t$(h$(1204)) THEN
3070 LET r=r+1
3080 LET h=h+1
3090 IF t$(h$(h)) THEN
3100 LET h=h+1
3110 LET r=r+1
3120 FOR i=1 TO 4
3130 IF t$(i,h) THEN
3140 LET i$=VAL LEFT$(LEN i$)+1
3150 ON i$
3160 LOOP UNTIL i$=1 OR i$=3
3170 DO
3180 LOOP UNTIL i$=1 OR i$=3
3190 END PROC

4800 DEF PROC keres n, REF es
4810 REM A táblázatban megkeresi n-et; ix,v:helyen van.

4820 LOCAL ix,v,i,j,n$
4830 DIM n$(10)
4840 FOR i=1 TO 4
4850 LET n$(i)=t$(i)
4860 NEXT i
4870 LET ix=v
4880 FOR j=1 TO LEN es
4890 IF t$(ix,es(j))
4900 kbalra
4910 kfele
4920 kjobbra
4930 NEXT j
4940 REM n-ben a tenyleges
4950 hely van.

4960 LOCAL i
4970 LET i=0
4980 LET v=1
4990 IF n$(ix,v) THEN
5000 LET ix,(v+1)
5010 IF n$(ix,v) THEN
5020 LET ix,(v-1)
5030 IF n$(ix,v) THEN
5040 LET ix,(v+1)
5050 IF n$(ix,v) THEN
5060 LET ix,(v-1)
5070 IF n$(ix,v) THEN
5080 LET ix,(v+1)
5090 IF n$(ix,v) THEN
5100 LET ix,(v-1)
5110 IF n$(ix,v) THEN
5120 LET ix,(v+1)
5130 IF n$(ix,v) THEN
5140 LET ix,(v-1)
5150 IF n$(ix,v) THEN
5160 LET ix,(v+1)
5170 IF n$(ix,v) THEN
5180 LET ix,(v-1)
5190 IF n$(ix,v) THEN
5200 LET ix,(v+1)
5210 IF n$(ix,v) THEN
5220 LET ix,(v-1)
5230 IF n$(ix,v) THEN
5240 LET ix,(v+1)
5250 IF n$(ix,v) THEN
5260 LET ix,(v-1)
5270 IF n$(ix,v) THEN
5280 LET ix,(v+1)
5290 IF n$(ix,v) THEN
5300 LET ix,(v-1)
5310 IF n$(ix,v) THEN
5320 LET ix,(v+1)
5330 IF n$(ix,v) THEN
5340 LET ix,(v-1)
5350 IF n$(ix,v) THEN
5360 LET ix,(v+1)
5370 IF n$(ix,v) THEN
5380 LET ix,(v-1)
5390 IF n$(ix,v) THEN
5400 LET ix,(v+1)
5410 IF n$(ix,v) THEN
5420 LET ix,(v-1)
5430 IF n$(ix,v) THEN
5440 LET ix,(v+1)
5450 IF n$(ix,v) THEN
5460 LET ix,(v-1)
5470 IF n$(ix,v) THEN
5480 LET ix,(v+1)
5490 IF n$(ix,v) THEN
5500 LET ix,(v-1)
5510 IF n$(ix,v) THEN
5520 LET ix,(v+1)
5530 IF n$(ix,v) THEN
5540 LET ix,(v-1)
5550 IF n$(ix,v) THEN
5560 LET ix,(v+1)
5570 IF n$(ix,v) THEN
5580 LET ix,(v-1)
5590 IF n$(ix,v) THEN
5600 LET ix,(v+1)
5610 IF n$(ix,v) THEN
5620 LET ix,(v-1)
5630 IF n$(ix,v) THEN
5640 LET ix,(v+1)
5650 IF n$(ix,v) THEN
5660 LET ix,(v-1)
5670 IF n$(ix,v) THEN
5680 LET ix,(v+1)
5690 IF n$(ix,v) THEN
5700 LET ix,(v-1)
5710 IF n$(ix,v) THEN
5720 LET ix,(v+1)
5730 IF n$(ix,v) THEN
5740 LET ix,(v-1)
5750 IF n$(ix,v) THEN
5760 LET ix,(v+1)
5770 IF n$(ix,v) THEN
5780 LET ix,(v-1)
5790 IF n$(ix,v) THEN
5800 LET ix,(v+1)
5810 IF n$(ix,v) THEN
5820 LET ix,(v-1)
5830 IF n$(ix,v) THEN
5840 LET ix,(v+1)
5850 IF n$(ix,v) THEN
5860 LET ix,(v-1)
5870 IF n$(ix,v) THEN
5880 LET ix,(v+1)
5890 IF n$(ix,v) THEN
5900 LET ix,(v-1)
5910 IF n$(ix,v) THEN
5920 LET ix,(v+1)
5930 IF n$(ix,v) THEN
5940 LET ix,(v-1)
5950 IF n$(ix,v) THEN
5960 LET ix,(v+1)
5970 IF n$(ix,v) THEN
5980 LET ix,(v-1)
5990 IF n$(ix,v) THEN
6000 LET ix,(v+1)
6010 IF n$(ix,v) THEN
6020 LET ix,(v-1)
6030 IF n$(ix,v) THEN
6040 LET ix,(v+1)
6050 IF n$(ix,v) THEN
6060 LET ix,(v-1)
6070 IF n$(ix,v) THEN
6080 LET ix,(v+1)
6090 IF n$(ix,v) THEN
6100 LET ix,(v-1)
6110 IF n$(ix,v) THEN
6120 LET ix,(v+1)
6130 IF n$(ix,v) THEN
6140 LET ix,(v-1)
6150 IF n$(ix,v) THEN
6160 LET ix,(v+1)
6170 IF n$(ix,v) THEN
6180 LET ix,(v-1)
6190 IF n$(ix,v) THEN
6200 LET ix,(v+1)
6210 IF n$(ix,v) THEN
6220 LET ix,(v-1)
6230 IF n$(ix,v) THEN
6240 LET ix,(v+1)
6250 IF n$(ix,v) THEN
6260 LET ix,(v-1)
6270 IF n$(ix,v) THEN
6280 LET ix,(v+1)
6290 IF n$(ix,v) THEN
6300 LET ix,(v-1)
6310 IF n$(ix,v) THEN
6320 LET ix,(v+1)
6330 IF n$(ix,v) THEN
6340 LET ix,(v-1)
6350 IF n$(ix,v) THEN
6360 LET ix,(v+1)
6370 IF n$(ix,v) THEN
6380 LET ix,(v-1)
6390 IF n$(ix,v) THEN
6400 LET ix,(v+1)
6410 IF n$(ix,v) THEN
6420 LET ix,(v-1)
6430 IF n$(ix,v) THEN
6440 LET ix,(v+1)
6450 IF n$(ix,v) THEN
6460 LET ix,(v-1)
6470 IF n$(ix,v) THEN
6480 LET ix,(v+1)
6490 IF n$(ix,v) THEN
6500 LET ix,(v-1)
6510 IF n$(ix,v) THEN
6520 LET ix,(v+1)
6530 IF n$(ix,v) THEN
6540 LET ix,(v-1)
6550 IF n$(ix,v) THEN
6560 LET ix,(v+1)
6570 IF n$(ix,v) THEN
6580 LET ix,(v-1)
6590 IF n$(ix,v) THEN
6600 LET ix,(v+1)
6610 IF n$(ix,v) THEN
6620 LET ix,(v-1)
6630 IF n$(ix,v) THEN
6640 LET ix,(v+1)
6650 IF n$(ix,v) THEN
6660 LET ix,(v-1)
6670 IF n$(ix,v) THEN
6680 LET ix,(v+1)
6690 IF n$(ix,v) THEN
6700 LET ix,(v-1)
6710 IF n$(ix,v) THEN
6720 LET ix,(v+1)
6730 IF n$(ix,v) THEN
6740 LET ix,(v-1)
6750 IF n$(ix,v) THEN
6760 LET ix,(v+1)
6770 IF n$(ix,v) THEN
6780 LET ix,(v-1)
6790 IF n$(ix,v) THEN
6800 LET ix,(v+1)
6810 IF n$(ix,v) THEN
6820 LET ix,(v-1)
6830 IF n$(ix,v) THEN
6840 LET ix,(v+1)
6850 IF n$(ix,v) THEN
6860 LET ix,(v-1)
6870 IF n$(ix,v) THEN
6880 LET ix,(v+1)
6890 IF n$(ix,v) THEN
6900 LET ix,(v-1)
6910 IF n$(ix,v) THEN
6920 LET ix,(v+1)
6930 IF n$(ix,v) THEN
6940 LET ix,(v-1)
6950 IF n$(ix,v) THEN
6960 LET ix,(v+1)
6970 IF n$(ix,v) THEN
6980 LET ix,(v-1)
6990 IF n$(ix,v) THEN
7000 LET ix,(v+1)
7010 IF n$(ix,v) THEN
7020 LET ix,(v-1)
7030 IF n$(ix,v) THEN
7040 LET ix,(v+1)
7050 IF n$(ix,v) THEN
7060 LET ix,(v-1)
7070 IF n$(ix,v) THEN
7080 LET ix,(v+1)
7090 IF n$(ix,v) THEN
7100 LET ix,(v-1)
7110 IF n$(ix,v) THEN
7120 LET ix,(v+1)
7130 IF n$(ix,v) THEN
7140 LET ix,(v-1)
7150 IF n$(ix,v) THEN
7160 LET ix,(v+1)
7170 IF n$(ix,v) THEN
7180 LET ix,(v-1)
7190 IF n$(ix,v) THEN
7200 LET ix,(v+1)
7210 IF n$(ix,v) THEN
7220 LET ix,(v-1)
7230 IF n$(ix,v) THEN
7240 LET ix,(v+1)
7250 IF n$(ix,v) THEN
7260 LET ix,(v-1)
7270 IF n$(ix,v) THEN
7280 LET ix,(v+1)
7290 IF n$(ix,v) THEN
7300 LET ix,(v-1)
7310 IF n$(ix,v) THEN
7320 LET ix,(v+1)
7330 IF n$(ix,v) THEN
7340 LET ix,(v-1)
7350 IF n$(ix,v) THEN
7360 LET ix,(v+1)
7370 IF n$(ix,v) THEN
7380 LET ix,(v-1)
7390 IF n$(ix,v) THEN
7400 LET ix,(v+1)
7410 IF n$(ix,v) THEN
7420 LET ix,(v-1)
7430 IF n$(ix,v) THEN
7440 LET ix,(v+1)
7450 IF n$(ix,v) THEN
7460 LET ix,(v-1)
7470 IF n$(ix,v) THEN
7480 LET ix,(v+1)
7490 IF n$(ix,v) THEN
7500 LET ix,(v-1)
7510 IF n$(ix,v) THEN
7520 LET ix,(v+1)
7530 IF n$(ix,v) THEN
7540 LET ix,(v-1)
7550 IF n$(ix,v) THEN
7560 LET ix,(v+1)
7570 IF n$(ix,v) THEN
7580 LET ix,(v-1)
7590 IF n$(ix,v) THEN
7600 LET ix,(v+1)
7610 IF n$(ix,v) THEN
7620 LET ix,(v-1)
7630 IF n$(ix,v) THEN
7640 LET ix,(v+1)
7650 IF n$(ix,v) THEN
7660 LET ix,(v-1)
7670 IF n$(ix,v) THEN
7680 LET ix,(v+1)
7690 IF n$(ix,v) THEN
7700 LET ix,(v-1)
7710 IF n$(ix,v) THEN
7720 LET ix,(v+1)
7730 IF n$(ix,v) THEN
7740 LET ix,(v-1)
7750 IF n$(ix,v) THEN
7760 LET ix,(v+1)
7770 IF n$(ix,v) THEN
7780 LET ix,(v-1)
7790 IF n$(ix,v) THEN
7800 LET ix,(v+1)
7810 IF n$(ix,v) THEN
7820 LET ix,(v-1)
7830 IF n$(ix,v) THEN
7840 LET ix,(v+1)
7850 IF n$(ix,v) THEN
7860 LET ix,(v-1)
7870 IF n$(ix,v) THEN
7880 LET ix,(v+1)
7890 IF n$(ix,v) THEN
7900 LET ix,(v-1)
7910 IF n$(ix,v) THEN
7920 LET ix,(v+1)
7930 IF n$(ix,v) THEN
7940 LET ix,(v-1)
7950 IF n$(ix,v) THEN
7960 LET ix,(v+1)
7970 IF n$(ix,v) THEN
7980 LET ix,(v-1)
7990 IF n$(ix,v) THEN
8000 LET ix,(v+1)
8010 IF n$(ix,v) THEN
8020 LET ix,(v-1)
8030 IF n$(ix,v) THEN
8040 LET ix,(v+1)
8050 IF n$(ix,v) THEN
8060 LET ix,(v-1)
8070 IF n$(ix,v) THEN
8080 LET ix,(v+1)
8090 IF n$(ix,v) THEN
8100 LET ix,(v-1)
8110 IF n$(ix,v) THEN
8120 LET ix,(v+1)
8130 IF n$(ix,v) THEN
8140 LET ix,(v-1)
8150 IF n$(ix,v) THEN
8160 LET ix,(v+1)
8170 IF n$(ix,v) THEN
8180 LET ix,(v-1)
8190 IF n$(ix,v) THEN
8200 LET ix,(v+1)
8210 IF n$(ix,v) THEN
8220 LET ix,(v-1)
8230 IF n$(ix,v) THEN
8240 LET ix,(v+1)
8250 IF n$(ix,v) THEN
8260 LET ix,(v-1)
8270 IF n$(ix,v) THEN
8280 LET ix,(v+1)
8290 IF n$(ix,v) THEN
8300 LET ix,(v-1)
8310 IF n$(ix,v) THEN
8320 LET ix,(v+1)
8330 IF n$(ix,v) THEN
8340 LET ix,(v-1)
8350 IF n$(ix,v) THEN
8360 LET ix,(v+1)
8370 IF n$(ix,v) THEN
8380 LET ix,(v-1)
8390 IF n$(ix,v) THEN
8400 LET ix,(v+1)
8410 IF n$(ix,v) THEN
8420 LET ix,(v-1)
8430 IF n$(ix,v) THEN
8440 LET ix,(v+1)
8450 IF n$(ix,v) THEN
8460 LET ix,(v-1)
8470 IF n$(ix,v) THEN
8480 LET ix,(v+1)
8490 IF n$(ix,v) THEN
8500 LET ix,(v-1)
8510 IF n$(ix,v) THEN
8520 LET ix,(v+1)
8530 IF n$(ix,v) THEN
8540 LET ix,(v-1)
8550 IF n$(ix,v) THEN
8560 LET ix,(v+1)
8570 IF n$(ix,v) THEN
8580 LET ix,(v-1)
8590 IF n$(ix,v) THEN
8600 LET ix,(v+1)
8610 IF n$(ix,v) THEN
8620 LET ix,(v-1)
8630 IF n$(ix,v) THEN
8640 LET ix,(v+1)
8650 IF n$(ix,v) THEN
8660 LET ix,(v-1)
8670 IF n$(ix,v) THEN
8680 LET ix,(v+1)
8690 IF n$(ix,v) THEN
8700 LET ix,(v-1)
8710 IF n$(ix,v) THEN
8720 LET ix,(v+1)
8730 IF n$(ix,v) THEN
8740 LET ix,(v-1)
8750 IF n$(ix,v) THEN
8760 LET ix,(v+1)
8770 IF n$(ix,v) THEN
8780 LET ix,(v-1)
8790 IF n$(ix,v) THEN
8800 LET ix,(v+1)
8810 IF n$(ix,v) THEN
8820 LET ix,(v-1)
8830 IF n$(ix,v) THEN
8840 LET ix,(v+1)
8850 IF n$(ix,v) THEN
8860 LET ix,(v-1)
8870 IF n$(ix,v) THEN
8880 LET ix,(v+1)
8890 IF n$(ix,v) THEN
8900 LET ix,(v-1)
8910 IF n$(ix,v) THEN
8920 LET ix,(v+1)
8930 IF n$(ix,v) THEN
8940 LET ix,(v-1)
8950 IF n$(ix,v) THEN
8960 LET ix,(v+1)
8970 IF n$(ix,v) THEN
8980 LET ix,(v-1)
8990 IF n$(ix,v) THEN
9000 LET ix,(v+1)
9010 IF n$(ix,v) THEN
9020 LET ix,(v-1)
9030 IF n$(ix,v) THEN
9040 LET ix,(v+1)
9050 IF n$(ix,v) THEN
9060 LET ix,(v-1)
9070 IF n$(ix,v) THEN
9080 LET ix,(v+1)
9090 IF n$(ix,v) THEN
9100 LET ix,(v-1)
9110 IF n$(ix,v) THEN
9120 LET ix,(v+1)
9130 IF n$(ix,v) THEN
9140 LET ix,(v-1)
9150 IF n$(ix,v) THEN
9160 LET ix,(v+1)
9170 IF n$(ix,v) THEN
9180 LET ix,(v-1)
9190 IF n$(ix,v) THEN
9200 LET ix,(v+1)
9210 IF n$(ix,v) THEN
9220 LET ix,(v-1)
9230 IF n$(ix,v) THEN
9240 LET ix,(v+1)
9250 IF n$(ix,v) THEN
9260 LET ix,(v-1)
9270 IF n$(ix,v) THEN
9280 LET ix,(v+1)
9290 IF n$(ix,v) THEN
9300 LET ix,(v-1)
9310 IF n$(ix,v) THEN
9320 LET ix,(v+1)
9330 IF n$(ix,v) THEN
9340 LET ix,(v-1)
9350 IF n$(ix,v) THEN
9360 LET ix,(v+1)
9370 IF n$(ix,v) THEN
9380 LET ix,(v-1)
9390 IF n$(ix,v) THEN
9400 LET ix,(v+1)
9410 IF n$(ix,v) THEN
9420 LET ix,(v-1)
9430 IF n$(ix,v) THEN
9440 LET ix,(v+1)
9450 IF n$(ix,v) THEN
9460 LET ix,(v-1)
9470 IF n$(ix,v) THEN
9480 LET ix,(v+1)
9490 IF n$(ix,v) THEN
9500 LET ix,(v-1)
9510 IF n$(ix,v) THEN
9520 LET ix,(v+1)
9530 IF n$(ix,v) THEN
9540 LET ix,(v-1)
9550 IF n$(ix,v) THEN
9560 LET ix,(v+1)
9570 IF n$(ix,v) THEN
9580 LET ix,(v-1)
9590 IF n$(ix,v) THEN
9600 LET ix,(v+1)
9610 IF n$(ix,v) THEN
9620 LET ix,(v-1)
9630 IF n$(ix,v) THEN
9640 LET ix,(v+1)
9650 IF n$(ix,v) THEN
9660 LET ix,(v-1)
9670 IF n$(ix,v) THEN
9680 LET ix,(v+1)
9690 IF n$(ix,v) THEN
9700 LET ix,(v-1)
9710 IF n$(ix,v) THEN
9720 LET ix,(v+1)
9730 IF n$(ix,v) THEN
9740 LET ix,(v-1)
9750 IF n$(ix,v) THEN
9760 LET ix,(v+1)
9770 IF n$(ix,v) THEN
9780 LET ix,(v-1)
9790 IF n$(ix,v) THEN
9800 LET ix,(v+1)
9810 IF n$(ix,v) THEN
9820 LET ix,(v-1)
9830 IF n$(ix,v) THEN
9840 LET ix,(v+1)
9850 IF n$(ix,v) THEN
9860 LET ix,(v-1)
9870 IF n$(ix,v) THEN
9880 LET ix,(v+1)
9890 IF n$(ix,v) THEN
9900 LET ix,(v-1)
9910 IF n$(ix,v) THEN
9920 LET ix,(v+1)
9930 IF n$(ix,v) THEN
9940 LET ix,(v-1)
9950 IF n$(ix,v) THEN
9960 LET ix,(v+1)
9970 IF n$(ix,v) THEN
9980 LET ix,(v-1)
9990 IF n$(ix,v) THEN

```

```

4500 DEF PROC balra
4510 REM Mozditas balra.
4520 REM Kijelzes melott.

4530 IF t$(ix,v) THEN
4540 LET t$(ix,v)=t$(ix,(v-1))
4550 LET t$(ix,(v-1))=CHRS 0
4560 LET ix,(v-1)
4570 END PROC

5000 DEF PROC alras
5010 REM Kirajza a jategot.

5020 LOCAL n,i,j,n$,es
5030 LET n=1
5040 FOR i=1 TO 13
5050 REM i:szam
5060 LET i$=i
5070 IF t$(i,n) THEN
5080 LET i$=i+1
5090 IF t$(i,n) THEN
5100 LET i$=i+1
5110 IF t$(i,n) THEN
5120 LET i$=i+1
5130 IF t$(i,n) THEN
5140 LET i$=i+1
5150 IF t$(i,n) THEN
5160 LET i$=i+1
5170 IF t$(i,n) THEN
5180 LET i$=i+1
5190 IF t$(i,n) THEN
5200 LET i$=i+1
5210 IF t$(i,n) THEN
5220 LET i$=i+1
5230 IF t$(i,n) THEN
5240 LET i$=i+1
5250 IF t$(i,n) THEN
5260 LET i$=i+1
5270 IF t$(i,n) THEN
5280 LET i$=i+1
5290 IF t$(i,n) THEN
5300 LET i$=i+1
5310 IF t$(i,n) THEN
5320 LET i$=i+1
5330 IF t$(i,n) THEN
5340 LET i$=i+1
5350 IF t$(i,n) THEN
5360 LET i$=i+1
5370 IF t$(i,n) THEN
5380 LET i$=i+1
5390 IF t$(i,n) THEN
5400 LET i$=i+1
5410 IF t$(i,n) THEN
5420 LET i$=i+1
5430 IF t$(i,n) THEN
5440 LET i$=i+1
5450 IF t$(i,n) THEN
5460 LET i$=i+1
5470 IF t$(i,n) THEN
5480 LET i$=i+1
5490 IF t$(i,n) THEN
5500 LET i$=i+1
5510 IF t$(i,n) THEN
5520 LET i$=i+1
5530 IF t$(i,n) THEN
5540 LET i$=i+1
5550 IF t$(i,n) THEN
5560 LET i$=i+1
5570 IF t$(i,n) THEN
5580 LET i$=i+1
5590 IF t$(i,n) THEN
5600 LET i$=i+1
5610 IF t$(i,n) THEN
5620 LET i$=i+1
5630 IF t$(i,n) THEN
5640 LET i$=i+1
5650 IF t$(i,n) THEN
5660 LET i$=i+1
5670 IF t$(i,n) THEN
5680 LET i$=i+1
5690 IF t$(i,n) THEN
5700 LET i$=i+1
5710 IF t$(i,n) THEN
5720 LET i$=i+1
5730 IF t$(i,n) THEN
5740 LET i$=i+1
5750 IF t$(i,n) THEN
5760 LET i$=i+1
5770 IF t$(i,n) THEN
5780 LET i$=i+1
5790 IF t$(i,n) THEN
5800 LET i$=i+1
5810 IF t$(i,n) THEN
5820 LET i$=i+1
5830 IF t$(i,n) THEN
5840 LET i$=i+1
5850 IF t$(i,n) THEN
5860 LET i$=i+1
5870 IF t$(i,n) THEN
5880 LET i$=i+1
5890 IF t$(i,n) THEN
5900 LET i$=i+1
5910 IF t$(i,n) THEN
5920 LET i$=i+1
5930 IF t$(i,n) THEN
5940 LET i$=i+1
5950 IF t$(i,n) THEN
5960 LET i$=i+1
5970 IF t$(i,n) THEN
5980 LET i$=i+1
5990 IF t$(i,n) THEN
6000 LET i$=i+1
6010 IF t$(i,n) THEN
6020 LET i$=i+1
6030 IF t$(i,n) THEN
6040 LET i$=i+1
6050 IF t$(i,n) THEN
6060 LET i$=i+1
6070 IF t$(i,n) THEN
6080 LET i$=i+1
6090 IF t$(i,n) THEN
6100 LET i$=i+1
6110 IF t$(i,n) THEN
6120 LET i$=i+1
6130 IF t$(i,n) THEN
6140 LET i$=i+1
6150 IF t$(i,n) THEN
6160 LET i$=i+1
6170 IF t$(i,n) THEN
6180 LET i$=i+1
6190 IF t$(i,n) THEN
6200 LET i$=i+1
6210 IF t$(i,n) THEN
6220 LET i$=i+1
6230 IF t$(i,n) THEN
6240 LET i$=i+1
6250 IF t$(i,n) THEN
6260 LET i$=i+1
6270 IF t$(i,n) THEN
6280 LET i$=i+1
6290 IF t$(i,n) THEN
6300 LET i$=i+1
6310 IF t$(i,n) THEN
6320 LET i$=i+1
6330 IF t$(i,n) THEN
6340 LET i$=i+1
6350 IF t$(i,n) THEN
6360 LET i$=i+1
6370 IF t$(i,n) THEN
6380 LET i$=i+1
6390 IF t$(i,n) THEN
6400 LET i$=i+1
6410 IF t$(i,n) THEN
6420 LET i$=i+1
6430 IF t$(i,n) THEN
6440 LET i$=i+1
6450 IF t$(i,n) THEN
6460 LET i$=i+1
6470 IF t$(i,n) THEN
6480 LET i$=i+1
6490 IF t$(i,n) THEN
6500 LET i$=i+1
6510 IF t$(i,n) THEN
6520 LET i$=i+1
6530 IF t$(i,n) THEN
6540 LET i$=i+1
6550 IF t$(i,n) THEN
6560 LET i$=i+1
6570 IF t$(i,n) THEN
6580 LET i$=i+1
6590 IF t$(i,n) THEN
6600 LET i$=i+1
6610 IF t$(i,n) THEN
6620 LET i$=i+1
6630 IF t$(i,n) THEN
6640 LET i$=i+1
6650 IF t$(i,n) THEN
6660 LET i$=i+1
6670 IF t$(i,n) THEN
6680 LET i$=i+1
6690 IF t$(i,n) THEN
6700 LET i$=i+1
6710 IF t$(i,n) THEN
6720 LET i$=i+1
6730 IF t$(i,n) THEN
6740 LET i$=i+1
6750 IF t$(i,n) THEN
6760 LET i$=i+1
6770 IF t$(i,n) THEN
6780 LET i$=i+1
6790 IF t$(i,n) THEN
6800 LET i$=i+1
6810 IF t$(i,n) THEN
6820 LET i$=i+1
6830 IF t$(i,n) THEN
6840 LET i$=i+1
6850 IF t$(i,n) THEN
6860 LET i$=i+1
6870 IF t$(i,n) THEN
6880 LET i$=i+1
6890 IF t$(i,n) THEN
6900 LET i$=i+1
6910 IF t$(i,n) THEN
6920 LET i$=i+1
6930 IF t$(i,n) THEN
6940 LET i$=i+1
6950 IF t$(i,n) THEN
6960 LET i$=i+1
6970 IF t$(i,n) THEN
6980 LET i$=i+1
6990 IF t$(i,n) THEN
7000 LET i$=i+1
7010 IF t$(i,n) THEN
7020 LET i$=i+1
7030 IF t$(i,n) THEN
7040 LET i$=i+1
7050 IF t$(i,n) THEN
7060 LET i$=i+1
7070 IF t$(i,n) THEN
7080 LET i$=i+1
7090 IF t$(i,n) THEN
7100 LET i$=i+1
7110 IF t$(i,n) THEN
7120 LET i$=i+1
7130 IF t$(i,n) THEN
7140 LET i$=i+1
7150 IF t$(i,n) THEN
7160 LET i$=i+1
7170 IF t$(i,n) THEN
7180 LET i$=i+1
7190 IF t$(i,n) THEN
7200 LET i$=i+1
7210 IF t$(i,n) THEN
7220 LET i$=i+1
7230 IF t$(i,n) THEN
7240 LET i$=i+1
7250 IF t$(i,n) THEN
7260 LET i$=i+1
7270 IF t$(i,n) THEN
7280 LET i$=i+1
7290 IF t$(i,n) THEN
7300 LET i$=i+1
7310 IF t$(i,n) THEN
7320 LET i$=i+1
7330 IF t$(i,n) THEN
7340 LET i$=i+1
7350 IF t$(i,n) THEN
7360 LET i$=i+1
7370 IF t$(i,n) THEN
7380 LET i$=i+1
7390 IF t$(i,n) THEN
7400 LET i$=i+1
7410 IF t$(i,n) THEN
7420 LET i$=i+1
7430 IF t$(i,n) THEN
7440 LET i$=i+1
7450 IF t$(i,n) THEN
7460 LET i$=i+1
7470 IF t$(i,n) THEN
7480 LET i$=i+1
7490 IF t$(i,n) THEN
7500 LET i$=i+1
7510 IF t$(i,n) THEN
7520 LET i$=i+1
7530 IF t$(i,n) THEN
7540 LET i$=i+1
7550 IF t$(i,n) THEN
7560 LET i$=i+1
7570 IF t$(i,n) THEN
7580 LET i$=i+1
7590 IF t$(i,n) THEN
7600 LET i$=i+1
7610 IF t$(i,n) THEN
7620 LET i$=i+1
7630 IF t$(i,n) THEN
7640 LET i$=i+1
7650 IF t$(i,n) THEN
7660 LET i$=i+1
7670 IF t$(i,n) THEN
7680 LET i$=i+1
7690 IF t$(i,n) THEN
7700 LET i$=i+1
7710 IF t$(i,n) THEN
7720 LET i$=i+1
7730 IF t$(i,n) THEN
7740 LET i$=i+1
7750 IF t$(i,n) THEN
7760 LET i$=i+1
7770 IF t$(i,n) THEN
7780 LET i$=i+1
7790 IF t$(i,n) THEN
7800 LET i$=i+1
7810 IF t$(i,n) THEN
7820 LET i$=i+1
7830 IF t$(i,n) THEN
7840 LET i$=i+1
7850 IF t$(i,n) THEN
7860 LET i$=i+1
7870 IF t$(i,n) THEN
7880 LET i$=i+1
7890 IF t$(i,n) THEN
7900 LET i$=i+1
7910 IF t$(i,n) THEN
7920 LET i$=i+1
7930 IF t$(i,n) THEN
7940 LET i$=i+1
7950 IF t$(i,n) THEN
7960 LET i$=i+1
7970 IF t$(i,n) THEN
7980 LET i$=i+1
7990 IF t$(i,n) THEN
8000 LET i$=i+1
8010 IF t$(i,n) THEN
8020 LET i$=i+1
8030 IF t$(i,n) THEN
8040 LET i$=i+1
8050 IF t$(i,n) THEN
8060 LET i$=i+1
8070 IF t$(i,n) THEN
8080 LET i$=i+1
8090 IF t$(i,n) THEN
8100 LET i$=i+1
8110 IF t$(i,n) THEN
8120 LET i$=i+1
8130 IF t$(i,n) THEN
8140 LET i$=i+1
8150 IF t$(i,n) THEN
8160 LET i$=i+1
8170 IF t$(i,n) THEN
8180 LET i$=i+1
8190 IF t$(i,n) THEN
8200 LET i$=i+1
8210 IF t$(i,n) THEN
8220 LET i$=i+1
8230 IF t$(i,n) THEN
8240 LET i
```

Írásmat a Sprite-ok bonckés alatt című cikk (1987/12. sz.) kiegészítésül szánom, megmutatva a C64-es és C128-as gépek VIC2 chipjének újabb, sehol sem definiált lehetőségét.

Programom eredménye azonos az említett cikk programjával, azzal a többlettel, hogy ha az aktuális video—RAM (16 k) legutolsó bájtjába — azaz \$3FFF, \$7FFF, \$BFFF, \$FFFF — egy nyolc bitre felbontottnak elképzelt bájtot helyezünk, akkor azt a VIC a képernyőn kívül, az ún. sprite-térben ábrázolni fogja. Mégpedig úgy, hogy a 40 vagy 38 oszlopos képernyő sprite-térben lévő részére minden oszlopba kirajzolódik az adott bájt bitképe, de az oszloponkénti külön ábrázolásra nincs módunk. Ha azonban egy kicsit hosszabb IRQ-rutinál a képernyőt az \$F9-es rasztersortól soronként megszakítjuk és soronként más-más bájtot ábrázolunk a sprite-térben, akkor például betűket, egyszerű grafikákat ábrázolhatunk, bár mind a 40 oszlopban ugyanazokat. Betűk ábrázolásakor a hatás egészen meglepő. Olyan, mintha történetesen PRINT-tel kiirhatnánk a képernyőre, a 28—29-es sorba. Sokan úgy hiszik, hogy a sprite-térbe csak sprite-tal mehetünk ki. Most, bizonyítékkal, hogy nem sprite-tal írunk ki, kirakhatunk néhányat ugyanoda. Az itt között program csak szemlélteti a módszert, és egy csik-scrollt valószínűleg meg alul-fölül, amely azután tetés szerint variálható. Íme a program:

```

C999 SEI
    LDA #00          CIA 1 IRQ kikapcsolása
    STA DC0E
    LDA #01
    STA D01A        VIC raszter IRQ bekapcsolása
    LDA D011
    AND #7F
    STA D011        raszterregiszter átvételének törlése
    LDA #2B
    STA #314
    LDA #C0
    STA #315        IRQ átirányítás $C02B-re
    LDA #F9
    STA D012
    LDA #00
    STA 9E          a sor, ahol a VIC IRQ-t vált ki
    CLJ             a táblázat mutatójának nullázása
                    IRQ engedélyezve
RTS
C02B LDA D019
    STA D019        ez VIC IRQ-nál kötelező!
    LDA D011
    STA #2          az eredeti regisztertartalom mentése a nullás lapra
                    LDA #00
                    STA D011
                    LDY 9E
                    LDA C055.Y
                    STA 3FFF
                    LDA C05D.Y
                    STA 9E
                    LDX #30
                    C04A DEX
                    BPL C04A   várakozó ciklus
                    LDA #2
                    STA D011   REG: eredeti érték, képernyő bekapcsolása
                    JMP EA31   normál IRQ végrehajtása
C055 .BYTE 7F,BF,DF,EF,F7,FB,FD,FF
C05D .BYTE 1,2,3,4,5,6,7,0

```

Minden szám hexadecimálisan értendő. Ez tehát a C64-es verzió. A C128-as változat csak abban különbözik ettől, hogy a JMP EA31 helyett JMP FA65 áll, valamint a nullás lapon más címeket kell alkalmazni, például

#2 helyett A3,
9E helyett BD.

Ezenkívül a programot a 0000—3FFF tartományba kell vinni, hogy a VIC regiszterei elérhessük. Indítás vagy BANK 15:sys kezdőcím, vagy monitorból:

.G FXXXX XXXX = a kezdőcím.

Pápai Ákos

BÁJTSOROZATOK TÖMÖRÍTÉSE

A TÖMÖRÍTÉS TERMÉSZETRAJZA

Egyre több olyan program készül, melynek az a feladata, hogy programokat és adatállományokat tömörítsen, azaz rövidebbé tegyen. Könyvnyitandó a dolgon, ilyen programok írásához kívánunk segítséget nyújtani.

A módszer a fájlban többször előforduló azonos sorozatokat tömöríti. A fontosabb tennivalók:

a) Azonos sorozatok keresése. Ezt a gyakorlatban úgy csináljuk, hogy az első — mondjuk 16 — bájtot kinevezzük keresendő sorozatnak, majd az egyes elemektől sorra összehasonlítjuk a 17. bájtton kezdődő — azaz a keresendő sorozatot közvetlenül követő — 16 bájtos sorozat megfelelő elemeivel. Ha mindegyik egyezik, a b) pont szerint folytatjuk; ha nem, a keresendő sorozat egyes elemeit a 18. bájtton kezdődő, tehát a keresendő sorozatot 1 bájtjal követő, 16 bájtos sorozat megfelelő elemeivel hasonlítjuk. Ha mindegyik egyezik, b) pont; ha nem, újabb 1 bájtos lépés előre stb., amíg elérjük a fájl végét. Akkor a keresendő sorozatot léptetjük tovább egy bájtjal, és kezdődik az egész előlről. Aztán megint a keresendő sorozatot léptetjük, amíg ezzel is elérjük a fájl végét. Akkor vége az adott hosszúságú sorozatok tömörítésének.

b) A megtalált sorozatok könyvjelzése. A megtalált sorozatot bemásoljuk egy táblázat következő üres helyére, ennek a helynek a sorszáma lesz az adott sorozat sorszáma. Vi-



gyázni kell arra, hogy a 256. sorozat megtalálása után fejezzük be a tömörítést. (Ezt a lépést egy adott sorozat harmadik, negyedik stb. megtalálása után már nem kell elvégezni.)

c) A sorozatok kicsérélése két bájtra: 1 — jelzőbájt, 2 — a sorozat sorszáma. Természetesen az egyik — mondjuk a 0. — „sorozatnak” magának a jelzőbájtnak kell lennie. A tömörítés megkezdése előtt a jelzőbájt minden előfordulása után be kell szűrni egy \$00 bájtot (0. sorozat). Újra jelentkezik tehát az a probléma, amiről a bájtos tömörítésnél már szó volt.

d) Vissza az a) ponthoz.

Ezeket a lépéseket többször egymás után, többféle sorozathosszal ajánlott végrehajtani.

Mint látható, a keresés során iszonyúan sok összehasonlítást kell végezni, emiatt a tömörítés akár órákig is eltart. Egy lényegesen gyorsabb keresőalgoritmus alapjainak leírása található a Tudomány (Scientific American) 1986-os szoftver-különszámának 31–32. oldalán, Niklaus Wirth Adatszerkezetek és algoritmusok című cikkében.

A szétpakolóknak a jelzőbájtot és a táblázat helyét kell ismernie. Ezek alapján nagyon könnyű dolga van: olvassa a tömörített fájlt, ha jelzőbájtot talál, a megfelelő sorozatot belefejt a szövegbe. Vigyázat! A továbbolvasást az éppen befejezett sorozat elején kell kezdeni, kivéve, ha a 0. sorozatot (jelzőbájt) fejtettük ki.

Deierl András—Molnár Imre



ÖRÖKÉLET

Miért is nincsen körülöttünk öröknyár! — ábrándoznak a napimádók. S miért nem lehetünk örökké gyermekek és ifjak, miért nem élhetünk örökké! — sóhajtoznak, akik álomképekkel számúzik és megtagadják a létező világot. Nem is gondolnak rá: milyen unalmas is volna az az örökkévalóság! — A játékok kedvelőinek

persze nem illúzió az örökélet. Sőt örülnek, ha minél több van belőle. Szerencsére Tuba Imre kódolt megint: C Plus/4-re és 64 kbájtá bővített C16-ra ismét készített egy csokorra való örökéletkódot.

Az ilyen terülj, terülj asztalkámra pedig igazán senki sem mondhatja hogy unalmas!

ATLANTIS	13660,29
	17128,145
BRIDGEHEAD	8282,165
	8444,165
	bomba
CUTBERT IN THE TOMBS	11446,165
	idő
DEATH RACE	7499,173
	7148,173
	8860,169
ENIGMA	8707,48
	extended play
	8933,234
	energy
FINDERS KEEPERS	13306,165
ICICLE WORKS II	28497,173
	21765,0
	24423,0
	time
JACK ATTACK	23354,0
	8635,165
	8640,234
	10460,234
	10473,165
LEAPER	8076,234
	11562,234
MONTY ON THE RUN	11572,173
PURPLE TURTLES	7077,173
	time
ROBO KNIGHT	8087,172
	15966,234
	time
TREASURE ISLAND	12171,0
	6029,173

Képcsere — bitenként

Játék- és rajzprogramokban gyakran szükség lehet két kép gyors és látványos cseréjére. Az alábbi rutinnal ez megvalósítható. A program memóriatarakos, mivel nem használ külön munkaterületet.

BASIC-ból az EXT 0,x,y,z utasítással hívható, ahol x — a cserélendő kép(részlet) kezdőcíme,

y — a háttér kép(részlet) kezdőcíme, z — a cserélendő kép(részlet)-ek hossza.

Például az EXT 0,32767,16384,15360 utasítás hatására az egész képernyőt felcseréli a 16384-es címen el tárolt képpel.

Futtatás előtt nem árt kimenteni a programot.

Kókai László

*A HTE Mikroszámítógépes Programnyelv
és Operációs Rendszerek,
és a MATE Irányításméleti szakosztályai
közös rendezésében*

A SIMULA 67 szimulációs bázisnyelv alkalmazási lehetőségei IBM PC-n

*címmel előadást tartunk. Előadó: Gáspár András,
a magyar S67 fordító szerzője.*

*Azoknak, akik két lemezt is hoznak magukkal,
a szerző díjmentesen ráviszi demoanyagát.*

*SIMULA nyelvdefiniációs könyvek is kaphatók a helyszínen
40 forintot áron.*

Minden érdeklődőt szeretettel várunk!

Időpont: 1989. február 9., csütörtök 14 óra

*Hely: Budapest V., Kossuth tér 6—8. MTESZ-székház,
III. em. 333.*

```

1 !*****
2 !# Bitenkenti kepcserere TVC-re *
3 !# (C)1988 BUDDHA Softhouse *
4 !# Kókai László *
5 !*****
10 KE=3735:I=0
20 READ A:IF A=-1 THEN GOTO 50
30 CH=CH+A:POKEKE+1,A:I=I+1
40 GOTO 20
50 IF CH=5562 THEN 60
55 PRINT"hiba a DATA sorokban!":END
60 PRINT"Kesz!!!":SOUND
70 POKE 33,3735 AND255:POKE34,3735/256
80 PRINT"EXT 0,sloter,hatter,hossz"
1000 DATA243,62,80,211,2,34,200,14
1010 DATA237,83,202,14,237,67,204,14
1020 DATA6,8,197,42,200,14,237,91
1030 DATA202,14,237,75,204,14,26,23
1040 DATA203,22,26,23,18,35,19,11
1050 DATA120,177,32,242,193,16,227,251
1060 DATA201,0,128,0,64,0,60,-1

```


AZ ESCAPE SZEKVENCIÁKRÓL

A szekvenciák rendkívül módon egyszerűsítik a gép programozását. Az EXOS ellipszist rajzol, egyenest húz, ADSR-görbét definiál stb. Az escape szekvenciák használatokra lényegében egy sztringet készítenk, amelynek első karaktere minden esetben az ESC (értéke 27 dec = 01B hex). A második karakter határozza meg, hogy melyik szekvenciát indítjuk. Az esetleges további karakterek a paraméterek. — Ezután a kész sztringet a kívánt csatornára — hang esetén például a 067 hex csatornára — írjuk. Az első program IS-BASIC-ben demonstrálja a szekvenciák használatát: definiál egy karaktert, állítja a palettaszíneket, ADSR-görbét, majd hangot generál (1. lista).

A GRAFIKÁRÓL

Az Enterprise-t használóknak a grafikus RAM-mal van a legtöbb problémájuk, pedig a megoldás nagyon egyszerű.

A grafikát a Nick chip kezeli, amely csak a 0FCH—0FFH szegmenseket, azaz 64 k-t címez (video-RAM). A Nick ezeket a szegmenseket, címeiket így látja:

SZEGMENS	CÍM/HEX	CÍM/DEC
0FCH-252D	00000H-03FFFF	00000-16383
0FDH-253D	04000H-07FFFF	16384-32767
0FEH-254D	08000H-0BFFFF	32768-49151
0FFH-255D	0C000H-0FFFFH	49152-65535

A Nick e szegmenseket mindig (!) így látja, függetlenül attól, hogy az aktuális grafikus szegmens esetleg nincs is a Z80-ra lapozva.

Amikor egy grafikus csatornát megnyitunk, akkor az EXOS meghatározott nagyságú video-RAM-ot utal ki. Amikor a RAM kezdő/végcímére vagyunk kíváncsiak, egy rövid programot futtatunk:

```
LD A,4
GRAPHICS CSATORNASZÁM (GRAPHICS-nál ez 101 dec)
LD B,3
ADDR speciális funkció
EXOS 11
```

Ezt a rutint a második BASIC programba építettem be (2. lista).

```
10 PROGRAM ESCAPE_BASICBOL
11 LET ESC$=CHR$(27):ESC VEZERLOKOD
12 ! *** SET CHARACTER ... ***
13 LET C$=ESC$;"K"&CHR$(32)
14 LET C$=C$&CHR$(0)&CHR$(0)&CHR$(126)
15 LET C$=C$&CHR$(66)&CHR$(66)&CHR$(66)
16 LET C$=C$&CHR$(66)&CHR$(126)&CHR$(0)
17 ! *** SET PALETTE ... ***
18 LET P$=ESC$;"C"
19 LET P$=P$&CHR$(7)&CHR$(32)&CHR$(0)
20 LET P$=P$&CHR$(0)&CHR$(0)&CHR$(0)
21 LET P$=P$&CHR$(0)&CHR$(0)
22 PRINT #102:C$,P$:! B-AS ESC FUNKCIO
23 ! *** ENVELOPE NUMBER ... ***
24 LET E$=ESC$;"E"&CHR$(1)&CHR$(2)
25 LET E$=E$&CHR$(255)&WORD$(0)&CHR$(63)
```

```
26 LET E$=E$&CHR$(63)&WORD$(1)&WORD$(0)
27 LET E$=E$&CHR$(193)&CHR$(193)
28 LET E$=E$&WORD$(10):PRINT #103:E$
29 ! *** SOUND ... ***
30 READ P
31 DO UNTIL P<0
32 LET S$=ESC$;"S"&CHR$(1)
33 LET S$=S$&WORD$(512*P)&WORD$(65535)
34 LET S$=S$&CHR$(48)&CHR$(1)
35 LET S$=S$&WORD$(11)&CHR$(0)
36 PRINT #103:S$
37 READ P
38 LOOP
39 DATA 90,90,90,94,90,90,94,94
40 DATA 90,92,90,94,92,94,90,90
41 DATA -1
```

```
10 PROGRAM VIDEO_RAM
11 TEXT 40:ALLOCATE 30
12 CODE X=HEX$( "3E,65,6,3,F7,B,ED,43" )
13 CODE =WORD$(X+15)&HEX$( "ED,53" )
14 CODE =WORD$(X+17)&HEX$( "C9,0,0,0,0" )
15 GRAPHICS LORES 4:CALL USR(X,0)
16 LET BC=PEEK(X+15)+256*PEEK(X+16)
17 LET DE=PEEK(X+17)+256*PEEK(X+18)-1
18 PRINT "NICK CIMEK: ";BC;"-";DE
```

A program futtatása után a BC regiszterben lesz a kezdő-, a DE regiszterben a végcím+1, úgy, ahogy a Nick látja.

Nézünk egy konkrét példát is! Tegyük fel, hogy első címként a 44076-ot kaptuk. Ez a Nick-cím 32 k-hoz tartozik (a táblázat alapján) és a 254-es szegmensen van, a 32 k-tól 44076—32768=11308 bájttal távol. Láttassuk a Z80-nal ezt a szegmenst. Adjuk be: OUT 177,254. Ekkor a Z80 az 1. lapján a 254-es szegmenst látja. (A második Z80-as lapon a 255-ös rendszer/grafikus szegmens.)

A Z80 első lapja 16384—32767-ig tart. Most a 16384-hez adjuk a különbséget: 16384+11308=27692. Tehát a grafikus csatorna a 27692-es Z80-as címen kezdődik. (De természetesen továbbra is elérhető így: SPOKE 254,44076, érték). A végcím met hasonlóan határozható meg. Példánknl maradvá a helyes Z80 cím 42091 lesz.

Vagyis a grafikát a Z80-nal is kezelhetjük (például scroll, sprite). Az ismertetett rutin ATTRIBUTE módban BC-be a bit-térkép, DE-be az attributumok kezdetét tölti.

EGY HASZNOS RUTIN

A kép valamennyi jellemzője egy sorparaméter-táblában (LPT), a táblán belül sorparaméter blokkokban (LPB) tárolódik. A tábla kezdetére az LP—POINTER (0BFF4H—0BFF5H) mutat. (Eről az EXOS könyvben a 284. oldaltól olvashatunk.) Hardver szövegmodban (TEXT 40) a képernyősorok összekoveredhetnek, így ha valaki POKE-ka-ir, meglepetések érhetnek. A 3. listán látható kis rutint olyan nagyobb programokba érdemes beépíteni, ahol BASIC-ből a gyorsaság kedvéért POKE-ot használunk. Paramétereit tekintve hasonló a PRINT AT utasításhoz, csak gyorsabb annál. Hívása CALL POS [y koordináta, x koordináta, ASCII kód].

EXOS-BEMUTATÓ

Az utolsó program — 4. lista — azoknak szól, akik most kezdenek EXOS-t gépi kódban programozni. Egy grafikus ablakot definiál és escape szekvenciákkal egy grafikát rajzol a program. Működésére könnyen rájöhetnek az EXOS-leírás és egy disassembler (MON) segítségével. Hajnal Csaba

```
10 DEF POS(PY,PX,CH)
11 POKE EY,PY:POKE EX,PX:POKE EA+3,CH
12 CALL USR(SR,0)
13 END DEF
14 ALLOCATE 50
15 CODE SR=HEX$( "0,2A,F4,BF,E5,21" )
16 CODE EY=HEX$( "0,0,6,4,87,CB,15,CB" )
17 CODE EX=HEX$( "14,10,F4,C1,9,1,4,0" )
18 CODE EA=HEX$( "9,5E,23,56,D5,11,1" )
19 CODE EX=HEX$( "0,0,9,8,7,1,1,4,0" )
20 CODE EA=HEX$( "ED,52,36,41,C9,0" )
```

3. lista

```
10 PROGRAM EXOS_DEMO
11 ALLOCATE 200
12 CODE R=HEX$( "06,01,0E,10,16,20,F7" )
13 CODE =HEX$( "10,06,01,0E,19,16,18,F7" )
14 CODE =HEX$( "10,06,01,0E,16,16,01,F7" )
15 CODE =HEX$( "10,06,01,0E,17,16,00,F7" )
16 CODE =HEX$( "10,3E,01,11" )&WORDS(R+41)
17 CODE =HEX$( "F7,01,10,07,06,56,49,44" )
18 CODE =HEX$( "45,4F,3A,3E,01,06,01,0E" )
19 CODE =HEX$( "01,16,1B,1E,01,F7,06,3E" )
20 CODE =HEX$( "01,01,0A,00,11" )
21 CODE =WORDS(R+72)&HEX$( "F7,0B,18,0A" )
22 CODE =HEX$( "1B,45,67,20,00,00,00" )
23 CODE =HEX$( "00,00,3E,01,01,06,00,11" )
24 CODE =WORDS(R+94)&HEX$( "F7,0B,18,06" )
25 CODE =HEX$( "1B,41,00,02,06,01,3E,1F" )
26 CODE =HEX$( "01,06,01,11,00,00,ED,43" )
27 CODE =WORDS(R+153)&HEX$( "ED,53" )
28 CODE =WORDS(R+155)&HEX$( "F5,C5,05" )
29 CODE =HEX$( "3E,01,01,06,00,11" )
30 CODE =WORDS(R+151)&HEX$( "F7,0B,11" )
31 CODE =HEX$( "10,00,E1,19,E5,D1,01,10" )
32 CODE =HEX$( "00,E1,37,3F,ED,42,E5,C1" )
33 CODE =HEX$( "F1,30,20,D6,C9,1B,45,00" )
34 CALL USR(0)
```

1. lista

4. lista

2. lista



AZ UGRÁLGATÓS

A programíráskor természetesen mindig az a legelső követelmény, hogy az adott feladathoz legcélszerűbben írjuk meg a programot.

Nagyon sok olyan feladat létezik, amely rendezést, átlagolást, fájl végeig olvasást, felülírást stb. igényel. Ilyenkor célszerű ezt külön megírni, s csak mindig az adott feladatba bedolgozni. Természetesen a bedolgozáson azt értem, hogy az új feladathoz át kell paraméterezni a régi eljárásainkat, de ez még mindig kisebb munka, mint ha állandóan újra meg újra megíránk őket.

A profi programozók külön lemezeken tárolják a saját, már egyszer megírt, de máskor is jól használható rutinjaikat. Ezzel időt és fáradságot takarítanak meg. Ezt a módszert mindenkinek ajánlom.

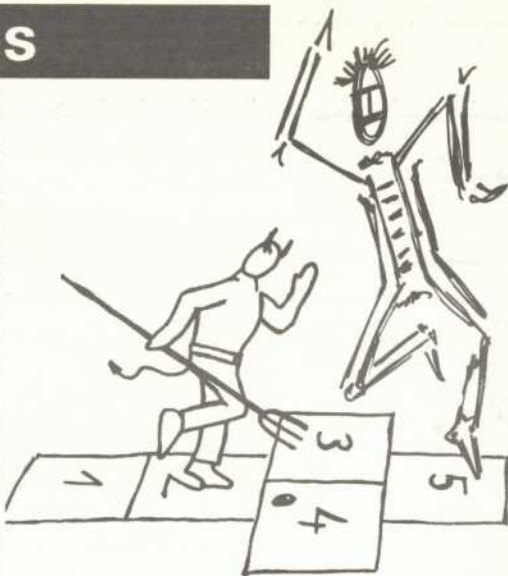
Sajnos, főleg a kezdők, akik megismerkednek egy programnyelvel, azt hiszik, hogy már profik, számukra nincs lehetetlen. De ha megnézi az ember egy feladatukat, bizony megrázó élményben lesz része, mivel a program(ocská) tele van fölösleges magyarázatokkal és rengeteg GOTO-val.

Tudvalevő, hogy a számítástechnika egyik nagyon csúf dolga a GOTO. Igaz, vannak esetek, amikor nem lehet kihagyni, de egyébként BÜN használni. Később, ha netán javítanunk kell, a sok GOTO egyszerűen úgy megkever (megugráltat) bennünket, hogy inkább újra írjuk az egész programot.

A következőkben bemutatam a rossz és a célszerű megoldást egy adott feladatra. A rossz nem úgy értendő, hogy nem működőképes, hanem hogy a program stilsztikailag elfogadhatatlan.

Az 1. lista egy IBM PC XT/AT PL/I fordító. Ez a program nagyon átláthatatlan: már ilyen kis feladatnál is gondot jelent a sok GOTO. A jó megoldást a 2. listán mutatom be.

Bartos Gyula



PROGRAMOZÁSI STÍLUSOK

```
PRG1:PROC OPTIONS(MAIN);
DCL SZALAG FILE INPUT ENV(MEDIUM(SYS001,2400)F(00)), WORK CHAR(80),
ON END FILE(SZALAG) GOTO VEG;
DO WHILE('1'0);
GOTO OLV;
C1:IF SUBSTR(WORK,9,2)='EZ' THEN DO PUT SKIP
EDIT(WORK)(A(00)); END;
END;
GOTO VEG;
OLV:GET FILE(SZALAG)EDIT(WORK)(A(00));
GOTO C1;
VEG:END PRG1;
```

1. lista

```
PRG2:PROC OPTIONS(MAIN);
DCL SZALAG FILE INPUT ENV(MEDIUM(SYS001,2400)F(00)),
WORK CHAR(80),
ON END FILE(SZALAG) GOTO VEG;
DO WHILE ('1'0);
GET FILE(SZALAG) EDIT(WORK)(A(00));
CALL VIZSG;
END;
VIZSG;
IF SUBSTR(WORK,9,2)='EZ' THEN CALL PRINT;
END VIZSG;
PRINT;
PUT SKIP EDIT(WORK)(A(00));
END PRINT;
VEG;
END PRG;
```

2. lista

TOP - lista

Felhasználói programok	IBM						386									
	IBM	AT/TA	C-128	C-64	C-4 (16)	SPECTR	ENTERP	TVC	IBM	AT/TA	C-128	C-64	C-4 (16)	SPECTR	ENTERP	TVC
1 Elga Point																
2 Keos 1.6 d.																
3 Rookmon v5.3																
4 Windows																
5 Elga Cad +																
6 Printfox																
7 Printmaster																
8 Disk manager																
9 Music master																
10 Starpainter																
Játék programok	IBM						386									
IBM	AT/TA	C-128	C-64	C-4 (16)	SPECTR	ENTERP	IBM	AT/TA	C-128	C-64	C-4 (16)	SPECTR	ENTERP	TVC		
1 Maniac M. 00.																
2 LLL																
3 Monopoly 00.																
4 Skyfox 00.																
5 TV Ball																
6 Elite																
7 Defender 00.																
8 Barbarian 00.																
9 Pirates!																
10 Batman																

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

Címünk: Mikroszámítógép Magazin Szerkesztősége
1371 Budapest, Pf. 433
Diák-szerkesztőség

Most induló hatrészes sorozatunkban az operációs rendszerekről adunk áttekintést. Az IBM PC hazai elterjedése nyomán egyre többen kerülnek kapcsolatba az MS-DOS-sal, ami bizony már lényegesen bonyolultabb, mint a C64, a Spectrum vagy az iskolaszámítógépek BASIC-re épített „operációs rendszere”.

A sorozat célja elsősorban a fogalmak, a problémák, valamint az ezek megoldásánál használt eszközök és módszerek olyan részletességgel való tárgyalása, hogy a külső szemlélők megkapják a legfontosabb információkat, az érdeklődők pedig jó kiindulási alapot nyerjenek a szakkönyvek megértéséhez.

Először az alapfogalmakat kell tisztáznunk, kezdjük tehát az operációs rendszerek működési területével.

Számítógépes rendszerek

Ezeknél a rendszereknél első közelítésben négy alapvető alrendszert lehet megkülönböztetni:

- hardver,
- operációs rendszer,
- nyelvek,
- alkalmazások.

Természetesen nem ez az egyetlen felosztás, hiszen gyakran elegendő az a durvább megközelítés is, amely az utolsó három alrendszer szoftver néven foglalja egybe. A felhasználók szempontjából a számítógépes rendszer alapja, a hierarchia legalsó rétege a hardver alrendszer, ami a gép látható felépítését reprezentálja. Fizikailag processzorokból, memóriákból, ki- és bevitteli egységekből, vezérlőegységekből és egyéb berendezésekből áll, amelyekhez olyan „elvont” alkotórészek is járulnak, mint az utasításrendszer (gépi kód), a címkezőlő alrendszer stb.

A hardverretek elsősorban az operációs rendszerek fejlesztői számára fontos, mivel az általuk létrehozott operációs rendszerrel tulajdonképpen a hardvert igyekeztek kibővíteni, amely azután jobban használható alapot ad a nyelvi rendszerek és az alkalmazások fejlesztésére.

Az operációs rendszer olyan kapcsolatot teremtet például a fordítóprogramot írónak, amely csökkentheti a programozási igényt és egyszerűsíti a gép egyes szolgáltatásainak igénybevitelét. Az operációs rendszer által létrehozott kapcsolat természetesen nem képes minden esetben a hardverretek teljes eltakarására. Egy fordítóprogram kódgenerátorának írójára nem nélkülözhető a tárgyvalny utasításrendszerének ismeretét, ami gyakran maga a gépi kód. Ugyanakkor a nyelvi alrendszereket — fordítóprogramokat, értelmező rendszereket stb. — létrehozó szakembereknek az az alapfeladatuk, hogy a programozási nyelvekkel kapcsolatot teremtsenek a gyakorlati feladatok specialistái és a számítógépes rendszer között.

Az előző réteghöz hasonlóan a nyelvek sem képesek az operációs rendszer teljes elfedésére. Sok programozási nyelv például lehetővé teszi az operációs rendszer egyes szolgáltatásainak igénybevitelét egy egyszerű hívási mechanizmuson keresztül. Több, ma is működő

rendszerben a fenti elv olyan, nyilvánvalóan hibás érvényesüléseivel is találkozhatunk, mint például a magas szintű nyelven írt programok hibáinak a hexadecimális memóriatartalom alapján való felderítése.

Az eddig elmondottak továbbvihetők az alkalmazások rétegére is. Ezzel az alrendszerrel az alkalmazásfejlesztők foglalkoznak, akik a programozási nyelveken keresztül támaszkodnak az operációs rendszer által bővített hardverre. Egy számítógépes rendszer használati értékét végső soron közvetlenül az alkalmazási szoftver mennyisége és minősége határozza meg. A többi alrendszer fejlettsége csak a jól hasznosíthatóságot teszi lehetővé.

A hierarchiához tulajdonképpen legfelső réteggént hozzávehetnénk a felhasználókat is. Ez az emberekből álló alrendszer azonban összetételében állandóan változik, tendenciájában bővül. A számítástechnika társadalmiasodásában ma már hivatalnokok, katonák, tervezőmérnökök és vezérgazdátok egyaránt felhasználók lehetnek. A felhasználók körének száma nem az összetétele, hanem a kiszolgálása a fontos. Állandóan szem előtt kell tartani, hogy a felhasználókat elsősorban a rendszertől kapható információ érdeki. Mint csoport, a lehető legkevesebbet akarja tudni a számítógépes rendszerek speciális tulajdonságairól. Egyes felhasználók azonban programoznak is, előre nem látható problémákat hoznak, melyek megoldásához elvárják a rendszer hatathos támogatását. Ezért a jó számítógépes rendszernek olyan felhasználói kapcsolatot és feladatmegoldó környezetet kell nyújtania, amely csak minimális mértékű számítógépes szakmértet igényel.

Az operációs rendszer fogalma

Az operációs rendszer nem teljesen egyértelmű fogalom. Függetlenül a hagyományoktól, a hardverretek jellemzőitől és a különböző szolgáltatásoknak a teljes rendszeren belül való elosztásától. Az operációs rendszert tehát környezetéből kiragadva — géptől függetlenül — nem lehet meghatározni, lehet adni azonban egy olyan általános felsorolást, amely az operációs rendszer részneinek tekintett alrendszereket tartalmazza. Egyes konkrét rendszereknél az alrendszerek halmaza szűkülhet vagy bővülhet.

1. A rendszer működésébe gépkészítői beavatkozást megvalósító alrendszer. Ide tartozik a rendszer átkonfigurálása, a működési statisztikák lekerdezése, programok és berendezések állapotának vizsgálata stb.

2. A rendszer és a külvilág közötti kapcsolat legalsó szintjét megvalósító bevitteli/kivitteli (B/K) alrendszer. Ide tartozik a B/K igények sorkezelése, a B/K műveletek indítása és a külső egységekről érkező megszakítások kezelése.

3. A multiprogramozás vezérlését megvalósító alrendszer. Tágabb értelemben ide sorolható minden olyan tevékenység, amely a rendszer erőforrásainak az igénylők közötti szétosztására vonatkozik. A munkavezérlő nyelvek értelmezői vagy a parancsértelmezők például az operációs rendszerek speciális alkotórészei, hiszen ezek sokszor magas szintű programozási nyelvek is. Nem tekinthető teljesen hibásnak az a nézet sem, amely ezeket az operációs rendszer bővítéseinek tekint.

Nem szokták az operációs rendszer részének tekinteni a programozási nyelvek gépi megvalósításait, bár vannak olyan gépek, amelyek az operációs rendszer teljes vezérlése

egy nyelv — az egyszerű mikrogepeknek általában a BASIC — köré épül.

Annak eldöntése, hogy egy szoftverkomponens az operációs rendszerhez tartozik-e, részben megítélés kérdése, részben a kérdező nézőpontjának függvénye, részben pedig a rendszer történelmi hagyományával kapcsolatos. Az első és második generációs gépeknek például vagy nem is volt operációs rendszer, vagy jórómán kizárólag hardverelemekből állt. Különféle kapcsolókból, címkevezérlő kúlcsovból, indítókból, megállítókból, lépésenkénti végrehajtó kiváló gombokból stb. A harmadik generációnál — 1965 körül — alakult ki a főleg szoftverrel megvalósított operációs rendszer (például az IBM DOS és az OS), amit ma már mindenki természetesen tart.

Hardver alapfogalmak

A számítógépes rendszerek legalsó rétege, a hardver, közelebből vizsgálva igen változatos képet mutat. A teljesség igénye nélkül általában a következő alkotórészeket különböztethetjük meg: processzorok, memóriák, lemezek, szalagok, nyomtatók, képernyők, billentyűzetek, adatátviteli vonalak. Ezek közül csak a fontosabbakat tárgyaljuk.

Processzorok

A felhasználók számára a processzorok csak egy elektronikus alkatrészt jelentenek — például Intel 8086 —, hasznos formájukban a „központi feldolgozóegység” (angol rövidítés: CPU) vagy röviden központi egység elnevezésű dobozoként jelennek meg, ami más alkatrészeket is tartalmaz. Bár a központi egység növekvő teljesítménye mellett is a számítógépes rendszer összköltsége egyre csökkenő hányadát képviseli, mégis, több szempontból is, a rendszer legfontosabb összetevője. Az operációs rendszerek felépítésének szempontjából a központi egység nehány tulajdonsága különösen nagy jelentőségű. Ilyen a címzési mechanizmus, az utasításrendszer, a megszakítási rendszer és a védelmi mechanizmus.

A címzési mechanizmus mindazon eszközök és módszerek összessége, amelyeknek segítségével az utasításokhoz memóriarekeszek rendelhetők hozzá. Legegyszerűbb eset az egyenes címzés, ahol a címek közvetlenül a memóriarekeszre hivatkoznak. Ilyen központi egység programozása igen nehézkes lenne. A következő „fokozat” egy vagy több indexregiszter használata, amelyeknek tartalma — általában összehadással — módosíthatja az egyenes címet. Az indexléssel is kombinálható indirekt címzés — a cím tartalma egy újabb cím, ahol az operandus van — sok lehetőséget tartogat az ügyes programozónak. Gyártókat már többszörös indirekciók megengedő központi egységeket is.

A központi memória méretének növekedésével együtt bővíteni kellett az utasítások címtartományát is. Ma már a 16 bites címmező 64 kb-át abszolút minimumot jelent. Hogy ne kelljen az utasításokhoz a teljes címtartományt megjelölni, bevezették a bázisregisztereket. Az alapcím a megfelelő méretű (például 32 bites) bázisregiszterben van, az utasítások csak az ehhez hozzáadandó növekményt (angolul: displacement vagy offset) tartalmazzák.

A háttérterek fejlődése lehetővé tette a virtuális tárkezelés kialakulását, amikor is a felhasználó által látott, jórómán korlátlan méretű tárolónak csak egy kis része létezik a rendszerben operatív memóriaként, nagy része valami-

motorja I.

lyen háttéráron — általában mágneslemezen — van. A központi egység címzési mechanizmusa főle az operációs rendszerben egy olyan virtuális térkezelő alrendszer építenek, amely a felhasználó számára a címhivatkozások során igényelt információcserét a háttérért az operatív tár között automatikusan elvégzi.

Az utasításrendszer a hardverretegben mindazon utasítások összessége, amelyeket a központi egység végre tud hajtani. Bár az utasítások szerkezete gépenként nagyon változatos lehet, a legtöbb utasítás egy utasításkódból és egy vagy több címből áll. Míg a régebbi processzoroknál az utasításkezelést néhány tucat utasításra korlátozták, a mai rendszerek általában több száz különböző utasításkódot használhat. A kódón és a cím(ek)en kívül gyakran valamilyen járulékos információ is szerepel az utasításban, például az indexelés, indirekció stb.

Az operációs rendszer egyrészt messzemenőn figyelembe veszi az eredeti utasításrendszer, másrészt általában kibővíti azt. A bővítés elvileg olyan, hogy az eredeti utasítások segítségével programozott utasítások egy kóddal hajthatók végre.

A bővítést megvalósító programokat szokták mikrokódnak, mikroprogramnak is nevezni. Az így létrehozott virtuális központi egység sok egyéb lehetőséget is tartalmazhat. Igen általában például több működési állapot megvalósítása. Ezek közül gyakoriak a kéttápláló architektúrák. Normál állapotban a bővített utasításkezelésnek csak egy részhalma használható, míg privilegizált állapotban a teljes utasításrendszer rendelkezésre áll. A privilegizált állapot az IBM 370-en például „supervisor állapot”-nak hívják. A privilegizált állapot legtöbb szór a megskatitásokkal kapcsolatos, sok rendszerben a megskatitás fellépése kiváltja a privilegizált állapotba való átmenetet is. Természetesen szükség van egy olyan utasításra is, amely normál állapotban mesterséges megskatitást és ezzel állapotváltást idéz elő; ilyen például az IBM 370 SVC utasítása. Az IBM PC-n a privilegizált állapotot az ún. MOS—DOS vagy rendszerhívások reprezentálják.

A megskatitási rendszerre azért van szükség, mert a számítógéprendszerek olyan alkotórészeket is tartalmaznak, amelyek párhuzamosan is működhetnek. Gondoljunk például a B/K műveletekre: ezek egyrészt időigényesek, másrészt általában csak egy indítást igényelnek a központi egységtől, végrehajtásukról külön processzorok — csatornák — gondoskodnak. A megskatitási rendszer teszi lehetővé a párhuzamosan üzemeltethető egységek közötti koordinációt, és a központi egységben belül előálló speciális feltételekre — például a processzorhírára, amit akár a nullával való osztás, akár egy nem létező bájtra való hivatkozás okozhat — való megfelelő reakciót.

A megskatitások kezelését a hardver- és szoftvereszközök igen változatos keveréke végzi bonyolult algoritmusok alapján. Minden megskatitási rendszerrel azonban azt az alapvető problémát kell megoldani, hogy egy véletlenszerűen bekövetkező esemény képes legyen megváltoztatni a műveletek végrehajtási sorrendjét és éppen elvégzésé alatt álló művelet eredményének megváltoztatása nélkül.

A végrehajtási sorrend megváltoztatása tulajdonképpen vezérlésátadást jelent a megskatitást feldolgozó ún. megskatitórutinra. A megskatitórutin indulása előtt azonban úgy kell rögzíteni — menteni — a központi egység állapotát, hogy a megskatitás feldolgozása után az eredeti program hibátlanul folytatható legyen.

A központi egység állapotát minden pillanatban egy rögzített hosszúságú állapotvektorral lehet jellegelni (utasításslámláló regiszterek, állapotjelzők tartalma). A megskatitórutin általában egy ugrótablán keresztül hívható, ahova a vezérlés a megfelelő megskatitás bekövetkezésekor hardver úton adódik át. Az ugrótablából a megskatitórutin átveheti a működéséhez szükséges paramétereket is.

A megskatitási rendszert főleg az teszi bonyolulttá, hogy egy megskatitás feldolgozása közben újabb megskatitás érkezik. amit szintén kezelni kell, és ez a jelenség jóformán tetszőleges mélységben, egymásba szalutávlya is felléphet. A problémát a gyakorlatban úgy egyszerűsítik, hogy bizonyos megskatitások kezelésének (például processzorhiba) idejére leltiják minden más jellegű megskatitás fogadását. Ez azt jelenti, hogy ha ilyen megskatitás bekövetkezik, azt a központi egység várakoztatja az éppen feldolgozás alatt álló processzorhiba kezelésének befejezéséig. Ugyanakkor természetes, hogy valamilyen megskatitás kezelése közben a központi egység más megskatitásokat elfogad. Egy nyomtatott működése közben például egy lemezrel való információcsere kényelmesen lefolytatható. Azt, hogy a megskatitórutinok is általában megskatitathatók legyenek, úgy oldják meg, hogy az állapotvektor egy veremként működő memóriába mentik. A veremből mindig az oda utoljára beírt állapotvektor emelhető ki — az ilyen vermet angolul LIFO=Last In First Out stack-nek hívják —, a veremutató pedig automatikusan áttál az eggyel mélyebben elhelyezett elemre.

A közös kód minden esetben az, hogy a rendszer minél kevesebbet ideig legyen elvágva a külvilágtól megskatitás fogadására alkalmatlan állapotban.

Védelmi mechanizmusra azért van szükség, mert a megskatitások kezeléséből már egyesesen következnek a multiprogramozásra való igény, már csak azért is, mert az operációs rendszer és a felhasználó programja „párhuzamosan” fut. A védelmi mechanizmusok alapvetően háromfajta védelmet nyújtanak: a B/K műveletek, a memória és a központi egység védelmét.

A B/K műveletek védelme tulajdonképpen a privilegizált utasítások segítségével oldható meg oly módon, hogy a külső egységekkel való információcsere alsó szintjét csak privilegizált utasításokkal lehet megvalósítani. Erre azért van szükség, mert különben egy hibás — például az állomány végét nem érzékelő — olvasóprogram megbéníthatná az egész rendszert.

A memória védelme azt jelenti, hogy minden program csak a számára valamilyen módon kiosztott memóriaterületen dolgozhat. Ha ezt a szabályt megsérti, egy megskatitás jön létre, és az operációs rendszer be tud avatkozni. Azt, hogy ilyenfajta védelemre szükség van, nem kell különösebben indokolni, hiszen például a felhasználói programok átirhatnak az operációs rendszer területének tartalmát, ami teljes anarchiához vezetne.

A központi egység védelme szintén a felhasználói programok egyszerűen viselkedése által okozható veszélyek elhárítását célozza. Ha például a program végtelen ciklusba kerül, az operációs rendszer nem kapja meg a vezérlést bizonyos, periodikusan ismétlődő feladatok (például a képernyő tartalmának frissítése) ellátásához, ami nyilván nem megengedhető. A szokásos megoldás az ún. óramegskatitás, amely szabályos időközönként generálódik. Az óramegskatitás lehet állandó (például 1/50 másodperc), vagy szabályozható időközű. Maga a megskatitás egy regiszterben való számlálással valósítható meg. Az órára vonatkozó műveletek természetesen csak privilegizált módban hajthatók végre.

Memóriák

Általánosan értelmezve memória minden információátvitelre használható eszköz, amely

lyel írás és olvasás útján kommunikálni lehet. Mi a számítógépes rendszerekkel kapcsolatban a gép belső vagy operatív tárolóját — melynek szavai vagy bájttjai az utasításkezelés számára elérhető — hívjuk memóriának. Ha mágneslemezekről, mágnesszalagról és egyéb hasonló memóriákról van szó, inkább a háttérért vagy külső tároló kifejezést használjuk.

A memória a hardverreteg nélkülözhetetlenül fontos része; nem véletlen, hogy egy számítógéppel kapcsolatban általában ez az első adat, amit megkérdőjelez vagy megadnak.

A fejlődés magával hozta, hogy nőtt a memóriák kapacitása és sebessége, csökkent az ára és fizikai mérete, a memóriákkal kapcsolatban egyéb szempontok — elsősorban az elérési mód — kerültek előtérbe. Általánosság váltak a csak olvasható (angolul ROM=Read Only Memory) memóriák, amelyekben a rendszer állandóan használt szoftverösszetevőt — például az operációs rendszer fontosabb rutinjait, fordítóprogramokat — tárolják folyamatosan. Egyes rendszerekben már a ROM előtt is alkalmaztak erősen korlátozott módot átirható tartalmú memóriákat — például a Burroughs B1700 gépe — a látható architektúra változtatása céljából. Ezek utódja az integrált áramkörök korában a PROM és az EPROM — a törölt, programozható memória — lett, amelyeknek a tartalma csak egy, a rendszeren kívüli folyamattal, elektronikus programozással, égetéssel írható át.

A mindennapi használatban a szalagot a mágneslemezek elterjedése szorította vissza. Előnye, hogy valamilyen kisebb, de még mindig 100 megabájttal mérhető kapacitással elfogadhatóan gyors véletlen elérés tesz lehetővé. A cserélhető fog feladásával az elérési idő tovább csökkent, a fix és cserélhető lemezek kombinációja pedig minden rendszer számára megvalósíthatóvá tette az alkalmazás közzétételhez legjobban illeszkedő memóriakonfiguráció kialakítását. Az ún. winchester-rendszert, valamint a hajlékony cserélhető lemezek bevezetésével a nagy gépek mellett a mini- és a mikrogepek is megkapták az olcsó, gyors és elég megbízható nagy kapacitású háttérárat.

Egyéb perifériák

A számítógéprendszerhez kapcsolható készülékek igen nagy változatosságúak mutatnak. Mi azonban csak a legfontosabbakkal, a nyomtatókkal, képernyőkkel, billentyűzetekkel és adatátviteli vonalakkal foglalkozunk. Ezek legfontosabb közös tulajdonsága, hogy bájttal-kénti adatátvitelt valósítanak meg.

A nyomtatók az írógéptől indulva a sornyomatokot keresztül a mátrix-, majd a lézernyomatokig sokat fejlődtek s nagy változatoságot mutatnak, minden számítógépes rendszer nélkülözhetetlen részei.

A képernyő és a billentyűzet a manuális adatszeres eszközei, a grafikus lehetőségekkel felruházott finom felbontású színes képernyő nagyban segíti az igazán barátságos — általában interaktív — alkalmazási rendszerek fejlesztését. Az operációs rendszerek fejlődésének külön lökést adott a képernyős terminál elterjedése, mivel így a nehézkes kötegel rendszereket a legtöbb alkalmazási területen a felhasználókat segítő, interaktív rendszerek váltották fel. Ez legálább egy hardvergenerációval feléről fejlődés volt.

Az adatátviteli vonalak a távközlési vonallal összekapcsolt számítógépeknek játszanak fontos szerepet. Az információátvitel itt bájttal-kénti, a vonalra az adat egy modem közbeteítésével jut. Az adatátvitelhez természetesen meglehetősen összetett szoftver tartozik, hiszen a kétdoldali kapcsolattartás bonyolult feltételeket teremt.

Programozási fogások és melléfogások

Cikksorozatomban a nyomtatásban közölt BASIC programokban sűrűn előforduló programozási hibákat mutatom be, ismertelve javításuk módját is. Legutóbb egy hibásan működő program javítás utáni változatát is túlságosan lassúnak találtam, ezért egy jobb algoritmust választva, több mint kétszázszor gyorsabb programot írtam ugyanannak a feladatnak a megoldására BASIC nyelven.

A most nagyító alá vett program hibátlanul működik, csak nagyon lassan. Közös vonása a múltkorival, hogy ez is egy feladatmegoldó pályázat megoldásaként került előm. Mivel a feladat és megoldása a Mikroszámítógép Magazin ma már — sokak számára — nehezen elérhető, első számaiban jelent meg, részletesen ismertetem mindkettőt.

A Magazin 1983. évi egyetlen számának *Agyafurmány* rovatában a 3. feladat a következő volt: „Készítsünk algoritmust vagy program segítségével számoljuk ki az alábbi feladat megoldását.

Hány olyan pozitív, egymilliónál nem nagyobb egész szám van, és melyek azok, amelyekre az alábbi feltételek mindegyike teljesül:

- négyszetszám
- köbszám
- a szám utolsó jegye nem lehet 2, 3 és 8.

Elhagyható-e az előbb felsorolt feltételek közül akár egy is úgy, hogy a feladat megoldása ugyanaz legyen?

Beküldendő az algoritmus és/vagy a magas szintű programozási nyelven (BASIC, Pascal, FORTRAN, ALGOL stb.) írt program, amely nem lehet 20 utasításnál hosszabb.”

A feladatot megoldottam, de megoldásomat — már nem emlékszem pontosan, milyen okból — nem küldtem be. A lap következő számában jelent meg az 1. listán látható megoldás, alatta a következő szöveggel:

„A képernyőn megjelenő számok között (10 darab) nincs egyetlenegy olyan szám sem, amely

2, 3, 8-ra végződik, így a feladat c) állítása elhagyható, és a megoldás ugyanaz lesz.”

A közölt program hibátlanul működik, erről a 2. listán látható eredményből bárki meggyőződhet. Ennek ellenére bosszantott, hogy ezt tekintették helyes megfejtésnek. Zavarónak találtam az IF...GOTO utasítás párral megvalósított ciklust és az egész típusú változók indokolatlan használatát is. Ezek kiküszöbölésével hoztam létre a 3. listán lévő változatot, amely nem egészen 3 százalékkal fut gyorsabban az eredetinel, tehát a hiba magának az algoritmusnak a megválasztásában van.

Könnyen beláthatjuk, hogy ha egy szám egyszerre négyszetszám is és köbszám is, akkor az a hatodik hatvány. Elegendő tehát a számok hatodik hatványát vizsgálni. Ez a művelet zsebszámológéppel is hamar elvégezhető. A 4. listán látható program az 1. listán levővel azonos megoldást ad, de annál több mint négyszázszor gyorsabb. A sorok összevonásával közvetlen módban is futtatható, de a futási idők összehasonlítása érdekében itt igyekeztem az eredetéhez hasonló szerkezetet használni.

Lássuk a futási időket. Az eredeti 16 290 TI-egységet vett igénybe, ami több mint négy és fél perc. Ezt kisebb változtatásokkal 15 833 TI-egységre lehetett csökkenteni. A 4. listán levő program 34 TI-egység alatt futott le, 479-szer gyorsabban az eredetinel.

Barna László

1. lista

```
5 REM AZ ALABBI PROGRAM
  COMMODORE 64 SZAMITOGEPEN KESZULT
10 I%=0
20 I%=I%+1
30 J=I%*I%*I%
40 FOR K=I% TO SQR(J)
50 IF J<>K*K GOTO 90
60 S%=S%+1
70 PRINT S%;I%;K;J
80 GOTO 100
90 NEXT K
100 IF I%<101 GOTO 20
110 PRINT "VEGE": END
```

3. lista

```
20 FOR I=1 TO 100
30 J=I*I*I
40 FOR K=I TO SQR(J)
50 IF J<>K*K GOTO 90
60 S=S+1
70 PRINT S;I;K;J
80 GOTO 100
90 NEXT K
100 NEXT I
110 PRINT "VEGE": END
```

2. lista

```
1 1 1 1
2 4 8 64
3 9 27 729
4 16 64 4096
5 25 125 15625
6 36 216 46656
7 49 343 117649
8 64 512 262144
9 81 729 531441
10 100 1000 1000000
```

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

4. lista

```
10 for s=1 to 10
20 i=s*s
30 j=i*i*i
40 k=i*s
50 print s;i;k;j
60 next
70 print "vege"
```

A TUDOMÁNSZERZÉSI ÉS INFORMATIKAI INTÉZET

előzetes megbeszélés szerint díjmentes programbemutatót tart (vidéken is) az általa forgalmazott oktatóprogramokból.
Horváth Zsuzsa 665-011/2663 mellék
vagy 813-197

Budapest, Pf. 454, 1372

Mintakötés nemesak hölgyeknek

Egy kis kötéstechika

A kötött holmikak gépen is kötődnek készítenek, bár ez utóbbiak inkább hasonlítanak a horgolótűhöz, mint a kézi kötésnél használatos eszközökhöz. A könnyűiparban az úgynevezett körkötőgépeknél egy henger palástjába mart alkotóirányú hornyokban működnek a tűk. Ahhoz, hogy a kötés mintás legyen, szükség van a gépen az erre szolgáló készülékekre, azokra, amelyek a kötőtűket különböző fajta mozgásokra készítik. A tűk mozgáspályájában létrehozott különbségekkel alakulnak ki a különböző formájú szemek, végül ezek összehatása alkotja a mintát. Ilyen típusú kötésmintát szemléltet az 1. ábra.

Tipikus készülék a ferde síkú fogazott tárcsa, az úgynevezett mintázókerék (2. ábra). A tűket magában foglaló henger, tűshenger mellett található. Minta szerint az egyes tűket más pályára tereli, mint ahol a többi tű halad. Míg a tűshenger körben forog, a helyben maradó mintázótárcsa legördül rajta, ahogyan két fogaskerék gördül egymáson. Eközben a mintázókerék különbözően kialakított (üresen hagyott vagy kitöltött fogközű) fogai a velük érintkezésbe kerülő tűket minta szerint terelik.

Ahogy a tűshenger körben forog, a hornyokban az egymás melletti tűk megkötik a szemeket. Az ilyen típusú kötött kelmében a szemek a csavarvonal mentén helyezkednek el (3. ábra).

A minta kialakításához a mintázókerék fogaiból csoportokat alakítanak ki. A gya-

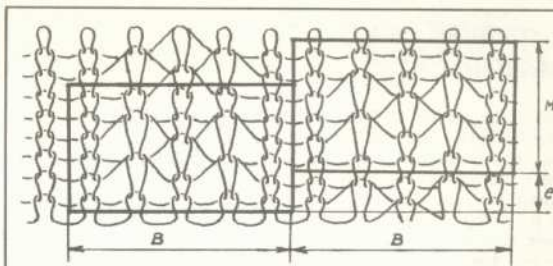
korlatban alkalmazottak három lehetőséget kínálnak:

- I. A fogcsoportok fogszáma egyenlő, a beosztásuk is egyforma, azaz az üresen hagyott és kitöltött fogházak minden csoportban azonos rendszer szerint ismétlődnek (4/a ábra).
- II. A fogcsoportok mérete (fogszáma) egyenlő ugyan, de beosztásuk különböző (4/b ábra).
- III. A fogcsoportok mérete egy kivételével egyforma (egy fogcsoport a többinél kevesebb számú fogból áll), azonban beosztásuk különböző (4/c ábra).

A fogcsoportok mérete (fogszáma) határozza meg, hogy az 1. ábrán látható mintaismétlődés – mintaelem – milyen széles (B), azaz egymás mellett hány szemből áll. Ha a mintázókerék valamennyi fogszáma K, akkor az I. és II. esetben a K/B osztásnak nincs maradéka, a keréken épp $M=K/B$ számú csoport fér el. A mintaelem téglalap alakú. A III. esetben az osztásnak maradéka van, így a mintaelem L alakú lesz (6. ábra).

Az, hogy a mintaelemek egymáshoz viszonyítva hogyan helyezkednek el – pontosan egy vonalban-e, avagy bizonyos eltolódással –, az attól függ, hogy a gép valamennyi tűjének száma (N) és a mintázókerék fogszáma (K), illetve a fogcsoport (B) mérete milyen viszonyban van egymással. Az I. eset a gyakorlatban akkor használható, ha az N tűszám és a B fogcsoportszélesség osztásánál $m < B$ maradék lesz, azaz

$$N = iB + m,$$



1. ábra

ahol $m < B$ és $K = MB$. Ilyenkor az 5. ábra szerint alakul a mintaelem. A II. esettel akkor számolhat a tervezés, amikor N-nek K-val való osztásakor olyan m maradék keletkezik, amely B-vel maradék nélkül osztható:

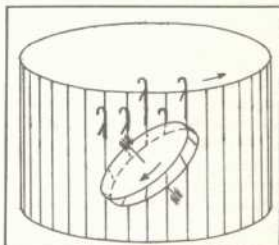
$$N = xK + yB,$$

ahol x és y egész számok és $K = MB$. Ez esetben levezethető az 1. ábrán e-vel jelölt magassági eltolódás

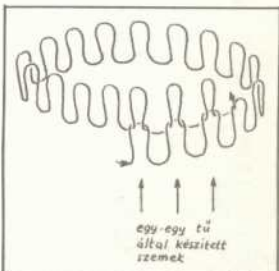
$$e = \frac{cH + 1}{y}$$

ahol $H = My/a$ és a, valamint c a legkisebb olyan egész szám, amelynél H-ra és e-re is egész szám jut.

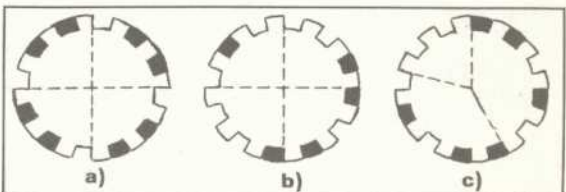
A III. esetben az alkalmazhatóság feltétele, hogy N-nek



2. ábra



3. ábra



4. ábra

K-val való osztásakor a maradék $m = B_n$ legyen:

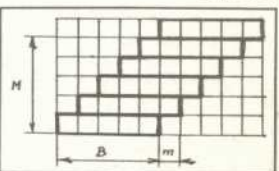
$$N = xK + B_n$$

és emellett fennálljon a

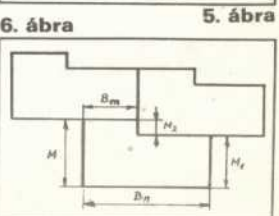
$$K = M_1 B_n + B_m$$

összefüggés is (6. ábra). M_2 ilyenkor 1-gyel egyenlő.

A kötőgépeken nem csak 1, hanem több (z számú) mintázókeréket is használnak, amik gazdagabb, nagyobb méretű minták kialakítását teszik lehetővé. A végleges mintaelemméretekre úgy hatnak, hogy a magassági méreteket (M, M_1, M_2) minden esetben z-vel kell szorozni.



5. ábra



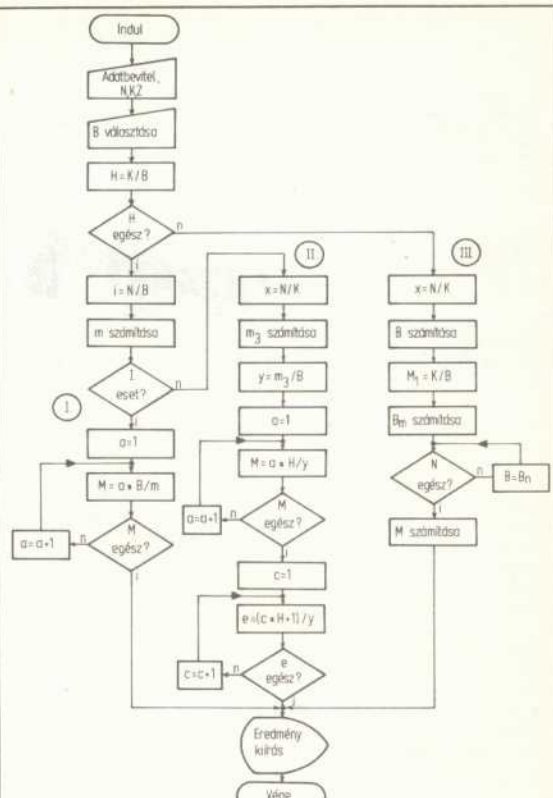
6. ábra

A számítási eljárás felépítése

Adott kötőegynél N , K és Z gépszervezeti adatnak számít, tehát ismert. A mintaelem B szélességét a tervező előileg szabadon veheti fel. Ha B a K -nak maradék nélküli osztója, a fenti I. vagy II. eset használható. Ha van maradék, akkor ez a III. esethez vezet. Amint az előzőekből kiderül, B azért nem lehet teljesen tetszőleges érték, mert ahhoz, hogy ésszerű mintaelemméretek és mintaelemlforma alakuljon ki, B -nek bizonyos feltételeknek kell megfelelnie. Ez — számítógép hiányában — csak bizonyos szempontok szerinti próbálgatásokkal állapítható meg. Számítógéppel azonban a munka lényegesen egyszerűsödik.

A számítási eljárás menete — ami egyúttal a számítógépes program felépítésének is alapja — a folyamatábrán (7. ábra) követhető.

Az I. és II. eset között az a különbség, hogy a fogcsoportok beosztása egyforma-e vagy sem. Ha igen (I. eset), akkor a mintaelem M magasságát az $M = aB/m$ osztás adja, ahol m az előzőek szerint az N tőszám és a választott B szélesség osztásánál keletkező maradék. Az a érték az a legkisebb szám, amelynél M -re egész szám adódik. Ha N -et K -val osztva maradék keletkezik, ez B egész számú többszöröse lesz, mely a II. esethez vezet. Ha ez nincs, a minta nem építhető fel ésszerűen. Tehát célszerű új B értéket választani, mégpedig lehetőleg a K és m legnagyobb közös osztóját. A mintaelem méretviszonyai akkor a legjobbak, ha ezzel dolgozunk. Nem szándékunk a szakmai részletekbe túlságosan belemerülni. A számítás nagyvonalú menete a folyamatábrán követhető. Részletei pedig a közölt, Sinclair Spectrumra írt programból kiolvashatók. A program ismertetésével célszerű volt, hogy bemutassuk: a



7. ábra

```

100 INPUT "A gép tőszáma (N) :"; N
110 INPUT "A mintaszélesség (B) :"; B
120 INPUT "A mintaelemmagasság (M) :"; M
130 CLS
140 PRINT "KÖTŐEGY FELTÉTELEK"
150 PRINT "MINTAELEM-MÉRETEK"
160 PRINT "SZÁMÍTÁS"
170 PRINT "© Lázár Károly 1987"
180 PRINT "-----"
190 INPUT "A gép tőszáma (N) :"; N
200 INPUT "A mintaszélesség (B) :"; B
210 INPUT "A mintaelemmagasság (M) :"; M
220 CLS
230 LET H = K/B
240 PRINT "H ="; H
250 IF H = INT(H) THEN GOTO 260
260 PRINT "H nem egész szám, a mintát nem lehet felépíteni."
270 GOTO 280
280 LET i = N/B
290 PRINT "i ="; i
300 PRINT "i eset?"
310 IF i = INT(i) THEN GOTO 320
320 PRINT "i nem egész szám, a mintát nem lehet felépíteni."
330 GOTO 280
340 LET a = 1
350 PRINT "a ="; a
360 LET M = a * B / m
370 PRINT "M ="; M
380 PRINT "M egész?"
390 IF M = INT(M) THEN GOTO 400
400 LET a = a + 1
410 GOTO 350
420 LET x = N/K
430 PRINT "x ="; x
440 PRINT "x3 számítása"
450 LET y = x3/B
460 PRINT "y ="; y
470 LET a = 1
480 LET M = a * H / y
490 PRINT "M ="; M
500 PRINT "M egész?"
510 IF M = INT(M) THEN GOTO 520
520 LET c = 1
530 LET e = (c * H + 1) / y
540 PRINT "e ="; e
550 PRINT "e egész?"
560 IF e = INT(e) THEN GOTO 570
570 LET c = c + 1
580 GOTO 530
590 PRINT "Eredmény kiírás"
600 PRINT "-----"
610 PRINT "Vége"

```

```

550 PRINT "A választott mintaelem magasságát az M = a*B/m osztás adja, ahol m az előzőek szerint az N tőszám és a választott B szélesség osztásánál keletkező maradék."
560 PRINT "A mintaelem méretviszonyai akkor a legjobbak, ha ezzel dolgozunk."
570 PRINT "Nem szándékunk a szakmai részletekbe túlságosan belemerülni."
580 PRINT "A számítás nagyvonalú menete a folyamatábrán követhető."
590 PRINT "Részletei pedig a közölt, Sinclair Spectrumra írt programból kiolvashatók."
600 PRINT "A program ismertetésével célszerű volt, hogy bemutassuk: a"

```

```

5000 REM ***** VÁLTOZÁS *****
5100 PRUSE 0
5200 LET W=MINKEYS
5300 IF W=1 THEN W=0
5400 IF W=2 THEN W=1
5500 RETURN
5600 REM ***** TÖRZSÁMÍTÁSI ELJÁRÁS *****
5700 LET N=10
5800 LET B=1
5900 LET M=1
6000 LET H=K/B
6100 IF H=INT(H) THEN GOTO 6200
6200 PRINT "H nem egész szám, a mintát nem lehet felépíteni."
6300 GOTO 6400
6400 LET i=N/B
6500 PRINT "i ="; i
6600 PRINT "i eset?"
6700 IF i=INT(i) THEN GOTO 6800
6800 PRINT "i nem egész szám, a mintát nem lehet felépíteni."
6900 GOTO 6400
7000 LET a=1
7100 LET M=a*B/m
7200 PRINT "M ="; M
7300 PRINT "M egész?"
7400 IF M=INT(M) THEN GOTO 7500
7500 LET a=a+1
7600 GOTO 7100
7700 LET x=N/K
7800 PRINT "x ="; x
7900 PRINT "x3 számítása"
8000 LET y=x3/B
8100 PRINT "y ="; y
8200 LET a=1
8300 LET M=a*H/y
8400 PRINT "M ="; M
8500 PRINT "M egész?"
8600 IF M=INT(M) THEN GOTO 8700
8700 LET c=1
8800 LET e=(c*H+1)/y
8900 PRINT "e ="; e
9000 PRINT "e egész?"
9100 IF e=INT(e) THEN GOTO 9200
9200 LET c=c+1
9300 GOTO 8800
9400 PRINT "Eredmény kiírás"
9500 PRINT "-----"
9600 PRINT "Vége"

```


24 vonalas be-, kimeneti bővítő

A bővítő a sokoldalú 8255-ös IC felhasználásával 24 külső eszközzel tud kapcsolatot tartani, mint például a nyomógomb, a modellvasút váltója, jelző LED stb. Ezek állapota a C64 programjából lekérdezhető és vezérelhető. A bővítő használatával a user port nyújtotta független csatlakozási lehetőségek száma megnégyszereződik. Óriási előnye, hogy a bővítőhöz csatlakoztatott eszközben — az amatőr munkáról lévén szó — elküvetett hiba csak a bővítő 8255-ös IC-jét teszi tönkre, s nem a C64 belsejében lévő, nehezen cserélhető IC-eket. (Extrém esetben, például 220 volt rákapcsolásakor természetesen a C64 is károsodik.)

A 8255-ös IC 24 darab vonala három darab úgynevezett portra van osztva. Ezeket A, B, C jellel jelöljük. A portok beállíthatók jel fogadására (INPUT) vagy jel kiadására (OUTPUT). A C port 8 vonala két csoportra osztva állítható be.

Bekapcsolás után mind a 24 vonal INPUT-ban van, azaz jel fogadására állítva. A 24 vonal állapota portonként kérdezhető le, a következő utasításokkal:

A PORT:PEEK (57088)
B PORT:PEEK (57089)
C PORT:PEEK (57090)

A beolvasott szám a következő módon mutatja a bejövő vonalak állapotát:

128	64	32	16	8	4	2	1	
								számérték
7	6	5	4	3	2	1	0	vonal száma

Ha a beolvasott számot felbontjuk a fenti számértéksorban található számokra, akkor amelyik szám szerepel benne, az ahhoz tartozó vonal logikai 1-ben van (azaz 2,4 voltnál magasabb a feszültsége), amelyik viszont nem, az ahhoz tartozó vonal logikai 0-n van (azaz 0,8 voltnál alacsonyabb a feszültsége).

A bővítővel úgy vezérelhetünk, hogy a 8255 utasításregiszterébe parancsot küldünk. Ez a POKE 57091,XX utasítással hajtható végre.

Az XX értékét aszerint határozzuk meg, hogy mely vonalakat kívánjuk OUTPUT-ba fordítani, azaz kimenetként használni. Az XX alapértéke 155. Ekkor minden vonal befelé fordul, mint a bekapcsolás után. Ebből levonva az alábbi számokat, a jelölt portvonalak kimenetbe fordulnak a POKE 57091,XX utasításra

A port 0—7 vonala: 16
B port 0—7 vonala: 2
C port 0—3 vonala: 1
C port 4—7 vonala: 8

A kimenetek alacsony szinten 1,7 mA-t nyelnek (TTL-terhelés). Ha tran-

zisztort hajtunk meg, akkor 750 ohmos soros ellenállást alkalmazva 1—4 mA-es meghajtó áramot kapunk. Ekkor a kimenetet logikai 1-be programozzuk.

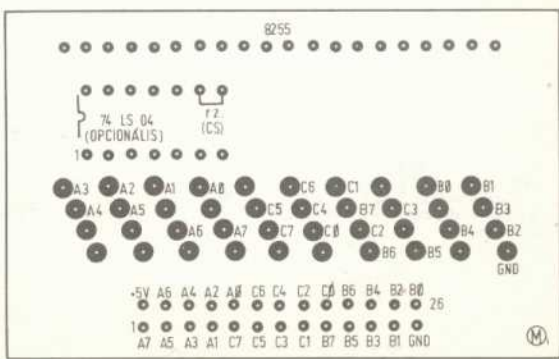
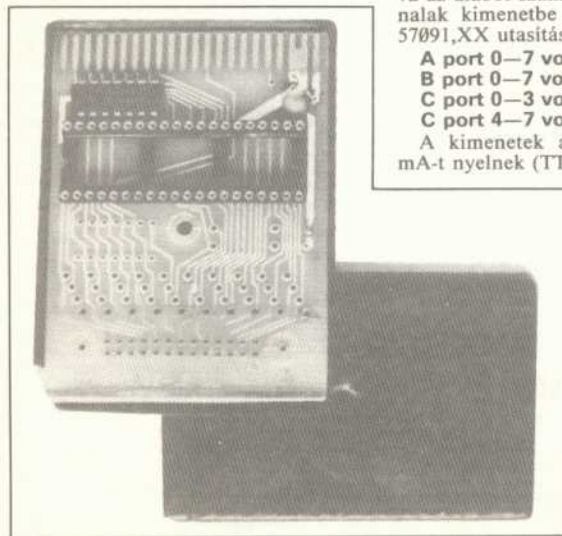
A 8255 a leirtakon kívül számos más üzemmódban is működtethető. Alkalmasságuk között gyors, párhuzamos adatátvitelre, intelligens eszközökkel való kapcsolattartásra, hand-shake üzemmódra. Ezek a profi felhasználási módok a 8255 gyári adatlapjain találhatóak.

A bővítőkátyán van hely egy opcionális 74LS04-es vagy ezzel lábkompatibilis IC beforrasztására, amely a kimenő vonalak meghajtására használható. Ezen a kártyán egyedül a PC5 vonalat hajtja meg. Az IC beforrasztásakor a 8—9. láb közötti rövidzár megszüntendő! A 74LS04 szabad kapui a többi vonalba beköthetők.

Bekötéskor a bővítőkátyába kétféle csatlakozó forrasztható: az egyik a Kontakta gyártotta 26 pólusú laposkabel-csatlakozó, a másik egy 40 lábú laposkabel-csatlakozó, ferdén eltolt pontokkal. Természetesen a legegyszerűbb a közvetlen beforrasztás. A csatlakozópontok kiosztása az ábrán látható.

A C64-es +5 voltos tápját külső eszközzel legfeljebb 100 mA-rel szabad terhelni. Ebből a bővítőkátya elhasznál 45-50 mA-t. További külső eszközt tehát maximum 50 mA-es fogyasztásig csatlakoztassunk a bővítő +5 voltos pontjára!

Mozsáry Gábor
Bp., Kékgolyó u. 19. 1123.



Nem rejtjük véka alá

Levelezési rovatunktól önállósulva közlünk két levelet. Az egyiket szerkesztőségünk kapta, a másik megírására — persze az is hozzánk érkezett — mi kértük az illetékes szervezet. A különös levélváltásban szívesen működünk közre, mert a papírra vetett gondolatok, javaslatok, vélemények — pro és kontra — az ország számítástechnikai kultúrájára világítanak rá.

Ime az első levél és a hozzá mellékelt felhívás, tiltakozás, melyet harmincan írtak alá:

Jisztelt Kovács Győző!

És a válasz:

A Mikroszámítógép Magazin 1988/9. számának 2. oldalán, továbbá a 22—23. oldalán foglaltakkal egyetértünk, és a gondolatot tovább folytatjuk.

Ha a külföldi vételár a hazai valutaárfolyamon keresztül is elérhető a hazai magánszemélyeknek, akkor a hazai vásárlót miért büntetik a külföldi vételár 2-5-8-szorosával, amit idehaza vámértéknek neveznek, és még ennek a 30 százalékaival? (Lásd a Vámárúk és áruk 2. bővített kiadását.)

A hazai HC (Home Computer) kategóriában olyan számítógépeket állítottak elő minimális darabszámban (HT-1080Z, Primo, Videoton TVC), amik jelenleg megszűntek, elavultak a szoftver és hardver hiánya miatt. Ezért nem lehet komoly hazai HC kategóriáról beszélni, így nincs mit védeni védővámokkal. Az egyetlen reális teljesítmény/ár aránya az Enterprise-nak van, de ez sem hazai gyártású.

Azazk a vállalatoknak (ha ugyan törődött is ezzel illetékes, annál inkább a vámértékkel) olyan típusok behozatalt kell megkönnyíteni, amik hosszabb távon is alkalmasak lesznek egyének, iskoláknak stb.

A hazai számítógépes kultúrát nem a 8 bites játékgépekre kell alapozni, hanem a 16/32 bites HC-PC típusokra, amelyek az árban is elérhető európai általános műszaki színvonalat képviselik! Például az IBM PC-XT és tartozékai vagy az Amiga 500 és tartozékai (Nyugaton ez a HC kategóriába tartozik!). Ezekhez a gépekhez komoly szoftver-hardver iralom van, valamint sokrétű programok léteznek, amik az egységesség (kompatibilitás) miatt is indokolták teszik e típusok országos elterjedésének támogatását! Ezeket a rendszereket ne sújtásd semmilyen büntetővámmal, mivel arra nincs elfogadható érv!

A miskolci Sinclair — Enterprise Klub tagsága úgy döntött, hogy tiltakozik a HC-PC kategóriájú számítógépeket és tartozékaikat sújtó indokolatlanul magas vámérték és vám ellen. Ezek megszüntetésére a FELHÍVÁS-ban foglaltakkal fordul az országban élő jelenlegi és leendő számítógép-tulajdonosokhoz, hogy csoportos vagy egyéni tiltakozásukkal az egyértelműséget fejezzék ki. A Magazin olvasóitól pedig várjuk egyéb észrevételeiket, javaslatokat kezdeményezésünkkel kapcsolatban.

Tisztelettel:

Sinclair — Enterprise Klub

FELHÍVÁS!

A PC számítógépek világában a 16/32 bites gépek lesznek világszerte elterjedőben — amit magánszemélyek is meg tudnak venni. A PC kategórián belül a HC (Home Computer) kategóriában is megjelennek a 16 bites gépek, amelyek minimálisan 512 k RAM-ot tartalmaznak beépítve, mint például az Amiga 500. A 8 bites gépekéé is minimum 256 k belső RAM és maximum 512 k külső RAM kerül alkalmazásra (Z80/B processzorú gépek).

1. A hazai viszonylatban „felső határként” megállapított 128 k RAM korlátozás technikai anakronizmus, és gátolja a korszerűbb rendszerek használatát.

2. A másik akadály a túlzásba vitt hazai HC-PC árak, vámárak, vámok.

Tiltakozunk az 1. és 2. pontokban megjelölt állapotok ellen, és követeljük a változtatást a magánszemélyek részére, a saját használatára behozt

— HC típusokra előírt 128 k RAM korlátozás eltörléséért,
— a HC-PC típusokat, tartozékaikat (monitor, nyomtató, dual floppy stb.) vámentesen leheszen behozni,

— a HC-PC típusokra előírt 25 000-és rendeletet szüntessék meg.

Sinclair — Enterprise Klub
Miskolc

A miskolci Sinclair — Enterprise Klub harmincfős tagsága által aláírt — és kérésüknek megfelelően a VPOP-hoz eljuttatott — levelet és a levél mellékletét képező, kiadni tervezett FELHÍVÁS-t át tanulmányoztuk, és az azokkal kapcsolatos álláspontunkról az alábbiakban tájékoztatom.

Főként 1988 éjétől — a külföldi utazás szabályainak megváltoztatásával — lehetővé vált a többszöri kiutazás, a családok kirándulás és a család összes tagja részére kiváltott valuta felhasználásával a komolyabb vásárlás, tehát nemcsak a devizaszámlával rendelkezők, hanem szélesebb rétegek is hozzájuthatnak az igényeiknek megfelelő számítógépekhez vagy egyéb — valjuk be, külföldön olcsóbb — műszaki cikkekhez. Nem kívánunk foglalkozni a hazai számítógépgyártással, illetve azzal, hogy miért nem gyártsanak az igényeknek megfelelő készüléket, ugyanis ez nem a Vám- és Pénzügyőrség hatáskörébe tartozik. Marad tehát a levélnek az a kitétele, hogy mit véd a vám, ha nincs hazai gyártmány?

Ez a kérdés korábban több áruval kapcsolatban felmerült, és a Pénzügyminisztérium az illetékes kereskedelmi szervekkel történt egyeztetés után, a felhasználás célját is figyelembe véve, egyes gépek, készülékek, szerszámok behozatala esetén lényeges vámkedvezményt biztosított. A vámkedvezmény azonban feltételhez kellett kötni, ugyanis tapasztalataink azt mutatják, hogy a külföldről behozott műszaki cikkek és közöttük főként a számítógépek és tartozékaik nem mindig annak a személynek a szükségleteit elégítik ki, aki azokat bezozta, hanem a felvásárló vállalatok, kisközvetítők stb. által a kereslet és a kínálat figyelembevételével, de sokszor indokolatlanul magas összegben megállapított átvételi árain eladásra kerülnek.

A levél és a felhívás alapján nem állapítható meg teljes biztonsággal, hogy mire kívánják felhasználni a gépeket, az „iskola” szó csak egyszer szerepel, amiből csak következtetni lehet arra, hogy oktatási célra történő felhasználásra is gondolnak.

A FELHÍVÁS szerint anakronizmusnak minősített 128 k RAM teljesítményű gépek az iskolákban — más szakemberek véleménye szerint — az alapismeretek elsajátítására alkalmasak, oktatási célra jól használhatók, míg a nagyobb gépekkel bémunkált stb. is folytatnák.

A külföldi vételárakat és a hazai kereskedelem által kialakított beföldi piaci árakat figyelemmel kísérjük, és a vám alapjául szolgáló beföldi forgalmi értéket szakértők bevonásával, több ízben kedvezően módosítottuk.

Az alapgépekhez csatlakoztatható tartozékok, kiegészítő egységek stb. minden meghatározott értéktől vagy teljesítménytől függetlenül részesülhetnek az 50%-os vámkedvezményben.

A levélben és a FELHÍVÁS-ban foglaltakkal — ami a fejlettebb gépek hosszúságát illeti — egyetértünk, az természetes, hogy jobb és nagyobb teljesítményű géppel jobban és többet lehet dolgozni, termelni, illetve felhasználási területük sokrétű lehetőséget biztosít a tulajdonos részére.

A közelmúltban már napilapok is foglalkoztak azzal a kérdéssel, hogy a vámjogszabály egyes részei módosításra kerülnek-e és ezen belül a vám alapjául szolgáló értéket milyen módon fogjuk megállapítani.

Valóban tervezés alatt áll ilyen változtatás, mely szerint a külföldi vételár és a hivatalos árfolyam figyelembevételével kerülne az érték megállapításra, úgy gondolom, ez a klub tagjainak, illetve a számítógépeket behozó személyeknek nagy segítséget fog nyújtani.

Az alapgépek, illetve a tartozékokra vonatkozó, valamint az egyéb készülékek stb. behozatalakor alkalmazásra kerülő vámkedvezmények mértéke is vizsgálatt tárgyat képezi.

Levelükben, illetve a FELHÍVÁS utolsó mondatában, sajnálatos módon, szerepel a „25 000-és” rendelet — ami valószínűleg a 17/1987. (XII. 27.) AH számú rendelet, mely szerint a 25 000 Ft-ot meghaladó értékű áruk eladása, bérbe vagy lízingbe adása korlátozásra került, tehát ennek eltörlését kérve a kereskedést és a bérbeadást stb. kívánják lehetővé tenni.

Az ilyen kereskedelmi, illetve haszonzerzésre irányuló tevékenység hivatott a fenti rendelet megváltoztatására. Ezt a rendeletet nem a Pénzügyminisztérium hozta, hanem a rendelet kiadásának időpontjában a piacfelügyelettel kapcsolatos intézkedésre illetékes Országos Anyag- és Árhivatal.

Kérem a fentiek szíves tudomásulvételét.

Vám- és Pénzügyőrség
Országos Parancsnoksága

ÁTALAKÍTÁS NÉMETRŐL ANGOLRA

Amint az köztudott, az Enterprise 128-at valóban több változatban forgalmazták. A működés szempontjából azonban lényeges különbség csak a gép bal oldalába dugható ROM-kártyában van.

Az „angol” gépek kártyájában egyetlen 16 k-s ROM IC van, amely az Intelligent Software BASIC interpreterjét tartalmazza. A „német” gépekében pedig egy ugyanilyen ROM mellett egy 16 k-s EPROM (27128) is található.

Am a ROM, illetve az EPROM nem akármilyen kommersz típus!

A Z80 mikroprocesszor utasításelhőzásokor mintegy 3, adat olvasásakor és írásakor pedig 2 órajelciklusnyi időt ad a memóriának a művelet elvégzésére. 4 MHz-es órajel mellett ez tehát 500 ns. A legtöbb olcsó EPROM ciklusideje 400-450 ns, tehát jól használhatók Z80 alapú rendszerekben.

Az Enterprise lapozásos memóriakezelése ebben az esetben hátrányos az amatőröknek. A bonyolult lapozó logika a Nick chipben közvetítőlegesen 200 ns-ot ellop a Z80 minden egyes memóriához fordulásakor. Vagyis a ROM-kártyában a lehető leggyorsabb címdekó-

doló IC-t: a TTL-AS vagy az ALS sorozatú demultiplexert és 250 ns-os hozzáférési idejű EPROM-ot kell használnunk.

A német gépek tulajdonosai az :UK paranccsal félig angolá varázsolhatják a gépüket. Azért félig, mert így csak a hibáuzenetek nyelvét és a billentyűzetet alakítják át. A valódi megoldás csak az, ha a német EPROM-ot működésen kívül helyezzük.

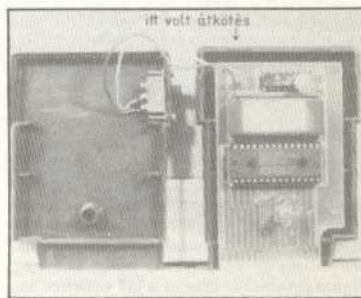
Ha valaki szedett már szét kétnyelvű ROM-kártyát — tényleg csak egy csavar tartja, és a gép garanciáját sem rontja el —, a kártya egyik sarkában (lásd a *fényképen*) egy kis drótdarabot figyelhet meg. Ez a kis drótdarab a megoldás: megszakítva, gépünk angolá válik! A drótvágásnál elegánsabb megoldás egy kicsi kapcsoló beépítése. Ez az átalakítás sajnos már ELRONTJA a ROM-kártya garanciáját. (Persze ha a számítógép meghibásodik, kölcsön ROM-mal is mehetünk a szervizbe.)

A ROM-kártyákban a ROM és az EPROM általában — az enyémben legalábbis — foglalatban van. A német kártyák EPROM-jának helyére saját EPROM-ot is tehetünk. Ebben a fontosabb

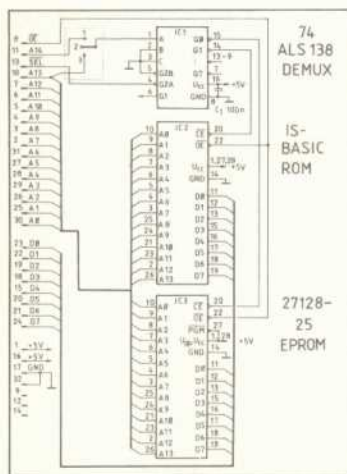
programjainkat, segédprogramjainkat tarthatjuk. Erről részletesebben majd később írok.

A német ROM-kártya kapcsolási rajza, valamint a nyomtatott áramkörti lap huzalozási rajza az (1-2 sz.) *ábrákon* láthatók. Ennek alapján otthon bárki elkészítheti a 16 kb-át vagy 32 kb-át EPROM befogadására alkalmas bővítő-kártyát.

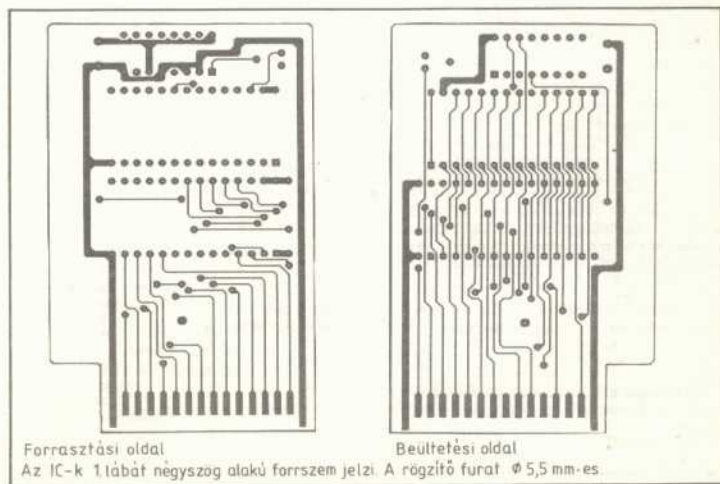
Az eredeti, „gyári” kártyák furatfémzett technológiával készültek. Ez igen bonyolult, anyag- és eszközigenyes, ezenkívül az egészségre káros technoló-



1. ábra



2. ábra



gia (a palládiumvegyületektől) sok embernek kihullottak már a fogai. Házilag a furatfémzést rövid drótdarabokkal pótolhatjuk, amelyek összekötik a panel két oldalán levő forrszemeket. Mivel a két oldalon huzalozott NYÁK-lemezeknél a legnagyobb probléma a kétoldali forrszemek „illeszkedésével” szokott lenni, ajánlatos először a még maratlan lemezt kifúrni. Ellenőrzésül: a kész panele 105 furatnak kell lennie. A kondenzátor lábaihoz 1 mm-es, máshova 0,6–0,8 mm-es lyukakat kell fúrni, különben „elfognak” a forrszemek. A kifúrt és lemaratott panelt célszerű ún. hideg önozóban bevonni, így elkerülhetjük a káros oxidációt. Aki nem tud a 32 pólusú csatlakozóra házilag aranyat galvanizálni, próbálkozzon meg néhány hónapoként az újraönozással.

E rövid technológiai kitérő után a ROM-kártya és az átalakítás működését ismertetem.

A 74 ALS 138 egy 3/8 demultiplexer, három címző és három engedélyező bemenettel. A címző bemenetek által kiválasztott kimenet alacsony szintű lesz, ha a G1 magas és a G2A és a G2B alacsony szinten van. Jelölések: L szint=„0” bit, H szint=„1” bit, X=a bemenet állapota nincs hatással a kimenetre.

Feltételezzük, hogy a 74 ALS 138 G2A engedélyező bemenetére aktív alacsony (L=low) szint érkezik. Ha most az A14 címvezetéken a dekóder „A” bemenetére magas (H=high) szint kerül, akkor az igazságtáblázat szerint a Q7 kimenet kerül az aktív L szintre, így engedélyezi a ROM működését. Ha eközben az OE (output enable=kimenet engedélyezése) jel is aktív L szintű, a ROM az adatbuszra kapuzza a kívánt bájtot.

A	B	A	B	D	C	D	C	-		
A	B	A	B	C	C	C	C	-		
A	B	A	B	d	c	d	c	d	c	-
0.	1.	2.	3.	4.	5.	6.	7.	8.		

- A: belső ROM 0–16K része
- B: belső ROM 16–32K része
- C: angol IS-BASIC ROM 0–16K
- D: német BASIC-bővítés EPROM 0–16K
- c: angol ROM 0–8K része
- d: német EPROM 0–8K része

szegmensek

Ha az A 14-en L szint van, akkor Q7 kimenet lesz alacsony szintű, így az EPROM kapja meg a CE (chip enable = működés engedélyezése) jelet.

G1	G2A	G2B	C	B	A	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
X	H*	X	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H	H
H	L	L	L	L	H	H	H	H	H	H	H	H	H
H	L	L	L	H	H	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	L	H	H	H	H	H
H	L	L	H	H	L	H	H	H	L	H	H	H	H
H	L	L	H	H	L	H	H	H	L	H	H	H	H
H	L	L	H	H	L	H	H	H	L	H	H	H	L

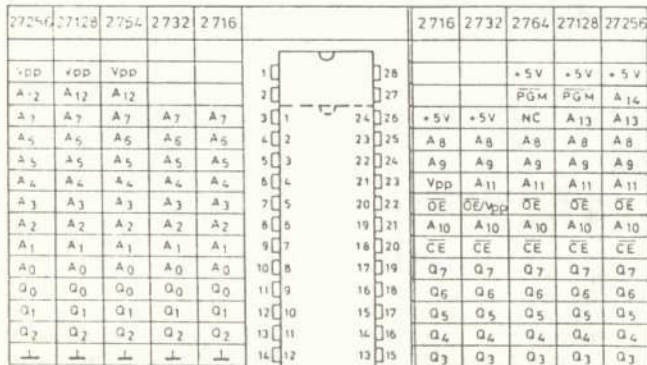
* G2A + G2B = H

A drót átvágásával — illetve a kapcsoló nyitásával — a dekóder IC A bemenete a legegyszerűbben fog lógni, amit a TTL áramkörök magas szintnek érzékelnek — nem tudják az L szintnek megfelelő 1,6 mA áramot kinyomni rajta. Így az A14 címvezeték állapotától függetlenül csak a ROM kaphat engedélyező jelet.

Ha az átkötést a másik — szűz — pontra tesszük, a dekóder IC A bemenete az A13 címvezetékhez kapcsoljuk. Ezzel azt érhetjük el, hogy a két memória engedélyezése közötti cikmülönbség 18 kbajt helyett 8 kbajt lesz. A 27xx sorozatú EPROM-ok és ROM-ok, sőt, az ugyanilyen kapacitású statikus RAM-ok lábkiosztása nagyon hasonló. A két foglalatba két 8 kbajtos EPROM-ot dugva, ezek egy 16 kbajtos memóriát alkot, illetve felső 8 kbajtos részei lesznek.

Az Enterprise EXOS operációs rendszere egyszerűen képes kezelni a ROM-kártyákat. Ahhoz azonban, hogy automatikusan induló és a számítógépet ténylegesen bővítő programokat tárolhassunk EPROM-ban (vagy CMOS SRAM-ban), bizonyos formai követelményeknek eleget kell tenni.

Mészáros László



A beépíthető 27xx sorozatú EPROM-ok lábkiosztása

↑
billentyűzet

1: +V _{CC}	32: GND
2: A ₇	31: A ₆
3: A ₈	30: A ₈
4: A ₉	29: A ₃
5: A ₉	28: A ₄
6: A ₁₁	27: A ₅
7: A ₁₂	26: A ₂
8: OE	25: A ₁
9:	24: D ₇
10: A ₁₃	23: D ₈
11: A ₁₄	22: D ₁
12:	21: D ₆
13: SEC	20: D ₅
14:	19: D ₂
15: D ₄	18: D ₃
16: +V _{CC}	17: GND

A ROM-BAY csatlakozó lábkiosztása kiegészítő számítógéphez „nézve”

Az Úttörő Áruháznak kiemelkedő szerepe van az Enterprise gépek értékesítésében. Ez az egyik a három budapesti márkabolt közül. A gépek kiskereskedelmi forgalmazása során összegyűjtött tapasztalatokról kérdeztük Herbert Ferencet, a műszaki osztály dolgozóját.

M. M. Az önök áruháza nagy lendülettel és hozzáértéssel kezdte el hazánkban egy teljesen új és ismeretlen típusú számítógép értékesítését. Kezdetben milyen nehézségei támadtak a műszaki osztálynak?

H. F. Azokban az áruházakban volt gond, ahol nem is akartak az új számítógéppel foglalkozni. Mi nagyon örültünk ennek az új, nagy tudású és olcsó gépnek. Az áruház rögtön az elején ötszázat rendelt belőle. Ez mutatja, hogy nagyon bíztunk benne.

M. M. Ennek köszönhető, hogy később az áruházat márkaboltta jelölték ki?

H. F. Azt hiszem, részben igen. Bár ez szerintem visszalépés az eredeti célokhoz képest. Az ország minden részéből jönnek hozzánk Enterprise-t vásárolni, miközben majd szétúrranak a méregtől, hogy miért nem kapnak gépet a lakhelyükhöz közelebb.

M. M. Hány gépet értékesítettek 1988 októberéig?

H. F. Körülbelül kétezer darabot. Ez elég nagy munkát igényelt tőlünk. Néha szédelegve megyek haza. Most is eladunk naponta három-négy darabot, de nehezebb feltételek között.

M. M. A megnövekedett turistaimportra gondol?

H. F. Nemcsak arra. Az emberek kevesebb pénzzel van szórakoztatva elektronika vásárlására. Az OTP-hitelalkció észrevehetően fellendítette a forgalmat. Sok vásárló kéri, hogy hasonlítsuk össze az Enterprise-t a Commodore gépekkel. Én nem vagyok a Commodore-ok ellensége, de ezt egyszerűen nem lehet megtenni. Az Enterprise teljesen más gép, és összehasonlíthatatlanul többet tud.

M. M. Nézzük meg az árakat. Egy Commodore-konfiguráció, hazai áron számolva, monitor nélkül körülbelül nyolcvanezer forintért megvásárolható. Ugyanez az Enterprise-nál mennyibe kerül?

H. F. Nézzük! Az alapgép 15 800 forint, a DOS controller 10 550 forint, egy jól használható nyomtató 24 400 forint és a VT meghajtó 29 300 forint. Ez is kijön körülbelül nyolcvanezer forintból. A vásárló azonban minőségileg sokkal többet

kap. A lemez 720 kbájtra formázható, nem beszélve a gépbe beépített RAM diszkról, ami további 4 x 16 kbajt virtuális tárat jelent. Ha valaki komolyan fejleszteni akar, annak ez kifizető lehetőség. A tárolt adatok IBM formátumúak. Ez sem lebecsülendő, mert több vevő azért vett Enterprise-t, hogy a munkahelyére lemezen vigye be az otthon összeállított adatokat.

M. M. Sokat hallottunk a Spectrum-emulátor körüli bonyodalmakról.

H. F. Az emulátor körüli bonyodalmakról.

MEGKÉRDEZTÜK AZ ENTERPRISE -RÓL

Hogyan élték át önök ezt az időszakot?

H. F. Nagyon sokan keresték. Ezek között volt, aki ennek reményében vette meg az alapgépet. Aztán emulátor helyett jöttek a kifogások. Egyéves késésre nincs mentés!

M. M. Milyen a jelenlegi helyzet?

H. F. Nemigen fogy. Nekünk az igazságnak megfelelően el kell mondanunk a vásárlónak az előnyeit, de a korlátait is. Ez nemcsak az emulátorra, hanem minden kiegészítőre is igaz. Sokan az árat is mérlegelve arra az álláspontra jutnak, hogy inkább vesznek egy Spectrumot.

M. M. A különböző „nyelvű” Enterprise gépek okoztak-e gondot?

H. F. Számottevő gondunk nem volt vele. Kétnyelvű és angol nyelvű gépeket kaptunk.

M. M. Milyen gyakori a három napon belüli reklamáció?

H. F. Nekünk ez olyan, mintha nem is lenne, gyakorlatilag nem fordul elő, mert alaposan kipróbált, „meleg” gépeket adunk el.

M. M. És milyen gyakran találunk hibás gépet?

H. F. A gépek minősége elég rapszodikus. Nem tudom pontosan, hogy hol és kik gyártották, de vannak öt és hat számjegyből álló gyártási számú gépek. Az öt számjegyűek kö-

zött gyakori hiba, hogy nem érzékelik a DOS-t és valami rendellenesség van a felső 64 kbájton. Szerintem az Enterprise gyenge pontja az RF modulátor. Lehet, hogy erre nem fordítottak a tervezésnél kellő figyelmet. Minden típusú, gyártási hibás televíziónál előfordulhat a sokat reklámlált „visszahatás”. Ennek az az oka, hogy ha az RF-jel valamiért visszaáramlik, azt a gép zavaroként értékeli és megszakítja a betöltést. Ez kiküszöbölhető az általunk forgalmazott szűrőtaggal, de ennek az 599 forintos árát soknak találok. Szerintem a programok is drágák, és egyre többre kerülnek. A máso-

lásoknak is szerintem az az oka, hogy drága és kevés szoftver van.

M. M. Sok panasz érkezett hozzánk, hogy a programkiszármakhoz gyenge és rossz leírások vannak.

H. F. Erre nem mondok semmit, inkább megmutatom azokat.

M. M. Amint látom, egyértelmű, hogy játékprogramokat ismertető füzetekben a leírások meglehetősen kurták és egyes esetekben érthetetlenek, illetve elnagyoltak. A vásárlók szoktak erre panaszkodni?

H. F. Elég gyakran. Itt jegyzem meg, hogy kérdéseire mindig a vásárlók észrevételei alapján válaszoltam, szinte az ő véleményüket mondtam el. Engem nem elsősorban mint kereskedőt, hanem mint embert bánt, hogy a vásárlók többsége nem számítógépet akar vásárolni, hanem „játékautomatát”. Nem is tudják, mennyi lehetőség és szépség rejlik egy ilyen gépben. Ők valószínűleg a szakmai folyóiratokat sem olvassák, ezért jó lenne különböző fórumokon erre a figyelmüket felhívni. Mert különben hatalmas lehetőségek maradnak kihasználatlanul.

Pinke György

Mi a manó?

A 0., azaz a látható lapra felrajzoljuk a kezdeti helyzetet (145-ös sor). A 152-es sorban jelezzük, hogy grafikus képernyőn akarunk dolgozni és (154-es sor) 16 színnel.

156-os sor: megnyitjuk az 1. csatornát, ez lesz az első videolap.

158-as sor: kijelöljük az első nyolc szint az 1. lapra.

160–190-es sorok: a téglalapokat az 1. lapra rajzoljuk.

205-ös sor: megjelenítjük az 1. lapot.

210–260-as sorok: változtatjuk az 1. lap színeit, és így megjelennek a képernyőn a téglalapok.

Tovább javítjuk a programot. Ezúttal minden fájszót más-más lapra rajzolunk (17. lista). Hét lapra van szükségünk, ezért már takarékoskodnunk kell a memóriával. A lapok méretét kisebbre vesszük. Magasságuk legyen 9 sor, szélességük 26 karakter (156–158-as sorok). Miután megnyitjuk a T. lapot (162-es sor), beállítjuk a ceruza színét (164-es sor), majd megrajzoljuk a „dominót” (180–182-es sorok). A lapokat egymás után megjelenítjük a képernyő 5. sorától kezdve (220-as sor). Jól látszik a lapmért, ha kiadjuk a SET BORDER 2 parancsot.

A „dominó” elhelyezéséhez tudnunk kell, hogy egy karakter 32 x 36 képernyőponton áll és a 26 karakter szélességű lapok — miután a képernyő közepén jelennek meg — a 7. karakterpozíciótól kezdődnek.

A 18. listán látható programmal a színek keverését mutatjuk be. Egy körlapot három körcíkkre osztunk, és a körcíkket befestjük a három alapszínnel. Ezután a körlapot megforgatjuk. A körlap fehér színűnek látszik. Futtassuk le a programot, majd változtassuk a színek arányát és összehajlítjuk. Ezek után talán könnyebb megérteni az RGB függvény jelentését (F–85. old.).



Tanítványom, Nagy Ákos, a miskolci Földes Ferenc Gimnázium II. E osztályos tanulója készítette az alábbi két programot, amelyek ugyancsak a videolapokat használják.

Mindenki ismeri azt a játékot, hogy ha egy füzetnek jobb oldali lapjaira egy mozgássorozat egyes elemeit laponként lerajzoljuk, és a lapokat gyorsan lepergetjük, mozgó képet kapunk. Ezzel a módszerrel készíthetünk az Enterprise-on gyorsan mozgó grafikus ábrákat. Az előbb leirt játékot úgy valósíthatjuk meg a számítógéppel, hogy a papírlapok helyett videolapokat használunk, és arra rajzoljuk a mozgás fázisait. A legegyszerűbben például (19. lista) integető embereket készíthetünk.

Ugyanerre a módszerre épül a következő, a 20. lista programja is, amellyel forgó kockákat varázsolhatunk a képernyőre. A program első részében megnyitjuk a 14 videolapot (160–180-as sorok). Előtte megadjuk a videolapok típusát és méretét. A 200–210-es sorokban beállítjuk a „nagy kockák” adatait, majd meghívjuk a kockarajzoló függvényt (220-as sor). Ezután ugyanezt csináljuk a „kis kockákkal” (230–260-as sorok) is. A kockarajzoló függvény a program végén található (350–610-es sorok). Amikor befejeződik a kockák rajzolása, elkezdődik a forgatás, azaz a videolapok megjelenítése (270–310-es sorok).

A nagy kockák rajzolása és a kis kockák rajzolása című részben megadott adatok átírásával megváltoztathatók a kockák méretei. Az a változóval jelezzük, hogy melyik kockáról van szó. Videolaponként mintegy 6,4 fokos az elforgatás, így a 14 lapon 90 fokkal fordulnak el a kockák. A videolapokat négyszer kell megjelenítenünk ahhoz, hogy a kockák egy teljes fordulatot tegyenek.

Dusza Árpád

A Téglatest a képernyőn című cikkben (1988/11. szám) lévő lista apró módosítása általánosabbá teszi a program tudását. A módosított sorok a következők:

$$150 \text{ LET YA2} = Z - X/2 + W$$

$$170 \text{ LET YA3} = Z - (X + Y)/2 + W$$

$$180 \text{ LET YA4} = W - (X + Y)/2$$

$$230 \text{ LET YB3} = Z - Y/2 + W$$

Enterprise panaszok. Wágner Miklós budapesti olvasónk többek között a következőket panaszolja. A Spectrum-emulátort és a lemez meghajtót nem lehet egyszerűen használni, így választás elé kerül, hogy melyiket csatlakoztassuk. Hasonló választást elé állították szerinte a felhasználót a kapható programokkal is. Ugyanis, ha a géphez csatlakozókártyát illesztünk, akkor a kazettán levő programokat nem lehet használni. Az első fájl betöltése után a gép lemezzel akar tovább tölteni, és mivel ez nem sikerül neki, leáll. Ha a programot átmásolja lemezzre, a betöltésnél az első fájl után szintén leáll a gép, mivel a periféria más-más hosszúságú fájlnevet fogad el, és így a rövidebb néven

a gép megint csak nem találja a folytatást. Ezek után szerinte a következők lehetőségek közül választhat:

— Szükség szerint felsereli a csatlakozókártyát, várva, hogy a csatlakozó meddig bírja.

— Addig megy, amíg a lemezre átírt programokat nem tudja lemásolni.

— Megfogadja a Novotrade Rt. eddig ki nem mondott tanácsát: aki annyira nem ért a számítógéphez, hogy egy nyavalyás játékgépprogramot sem tud átírni, az maradjon meg a Commodore-jánál.

Olvasónk panaszait azért tesszük közzé, hogy hátha valaki praktikus, könnyen megvalósítható tanáccsal segíteni tud neki.

Enterprise klubok. Rovatunk szeretné felvenni a kapcsolatot az országban megalakult Enterprise klubokkal. Kérjük, jelentkezzenek a szerkesztőség címén.



Gaetsch Günterné rajza

Hardver

A sorozat alapgondolata — azon a régi felismerésen túl, hogy az elektronika és a számítástechnika felválaszthatatlan 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 áramkört hardverismerettel is. Megértheti ezt, hogy szaporodik az olyan berendezések, mikroprocesszorok alkalmazású rendszerek száma, amelyek programvezérelten működnek. Az ilyen rendszerek tervezőinek és fejlesztőinek is szükségük van integrált hardver- és szoftverismeretekre.

Nagy bonyolultságú periféria-áramkörök

A mikroprocesszoros rendszerek alkalmazásának kezdetén a gyártó cégek még nem terveztek és nem is gyártottak semmiféle kiegészítő elemet. Nagyszámú kis- és közepes bonyolultságú elemet — kapukat, tárolókat, számlálókat — használtak fel, amelyekkel körülvették a mikroprocesszort: így teremtették meg a rendszer kapcsolatát a külvilággal. Ilyen módon illesztették az érzékelők és kapcsolók, nyomógombok — azaz a bemenetek — és a kijelzők, kiírók vagy végrehajtott elemek — a kimenetek — jeleit a mikroprocesszorhoz.

A jelek tényleges be-, illetve kivitelét a processzor végezte. Többek között olyan feladatokat is ellátott, hogy figyelt egy adott billentyűzetet, vagyis érzékelt egye billentyű megnyomott állapotát, aminek alapján egy programot elindított. Ez a megoldás igen sok időt vett el a processzortól, ami miatt a bonyolultabb feladatok megoldásánál már működési-időzírtési problémák is felmerültek.

Igen sok, külvilággal kapcsolatos feladat hasonló volt, ezek azonban önállóan, kis célhardverrel is megoldhatók lettek volna. Az LSI technológia felhasználásával a gyártók az egyszerű kiegészítő funkciókra egységeket állítottak elő, amelyeknél a mikroprocesszorral való összeköttetés a processzor buszjeleinek — cím-, adat- és vezérlőbusz — közvetlen rácsatlakozásával jött létre. Következésképpen szinte valamennyi mikroprocesszor-típushoz megalkották azokat a kiegészítő elemeket, amelyekkel a rendszerem közötti és az adott feladattal

való kapcsolat könnyebben és egyszerűbben létrehozható. Az 1. ábrán több ilyen alapfeladatkört mutatunk be, amelyek megvalósító LSI áramkörök többé-kevésbé a processzortípustól független, szabványos formában jelentek meg.

A fejlődés során megjelenő újabb periféria-vezérlők már programozhatók voltak. Ezáltal a processzorban lévő programmal olyan kódokat adhatunk ki a periféria-vezérlő áramköröknek, hogy azok tényleges működése és funkciója rugalmasan, a feladathoz még jobban igazodva változtatható. Ennek folytán ugyanaz a vezérlő áramkör — csupán a felprogramozása segítségével — több hasonló, de egymástól kísérletben eltérő feladatra is felhasználhatóvá vált. Ezeknél a processzor saját cím-, adat- és vezérlőjelei segítségével információcsere-t folytat a periféria-vezérlő valamelyik regiszterével (2. ábra).

Ezek a regiszterek funkciójuk szerint a következő módon csoportosíthatók:

- bemeneti regiszter,
- kimeneti regiszter,
- parancsregiszter,
- állapot- vagy másként státuszregiszter.

A be- és kimeneti regiszterek a külvilág felől/felé irányuló adatokat tárolják.

A parancsregiszterbe küldött információ a periféria-vezérlő működésének módját és paramétereit szabja meg a processzortól kapott kódoknak — parancsbjtoknak — megfelelően. Ilyen módon állítható be például az előző részben részletesebben bemu-

tatott 8255-ös párhuzamos be/kimeneti áramkör konkrét üzemmódja. Vagy így programozható például egy soros átvitel vezérlő áramkörnél az átvitel típusa (szinkron vagy aszinkron), az átvitt karakter hossza (bitszám), az átvitel sebessége (Baud-rate), a hibajelzést szolgáló paritásképzés módja (páros, páratlan, nincs) stb.

A vezérlő mindenkori állapotát az állapot- vagy szokásosan használt idegen szóval a státuszregiszter tartalmazza. A processzor ennek kiolvasásával szerezhet tudomást például az átvitt információ meghibásodásáról és ezen belül a meghibásodás módjáról. Ilyen történetesen az aszinkron soros adatátvitelnél a keretkezési hiba, ami nem más, mint hogy a zárt STOP bit helyén nem a megfelelő bit érkezik, azaz „elcsúszik” az információ.

Végül is tehát mind a tényleges adatok, mind a vezérlő működésének módját és állapotát meghatározó információk hasonló formában, bájtkokban vannak kódolva, és csupán a rendeltetési hely címe különbözteti meg ezeket.

Mint azt már a bevezetőben említettük, a legtöbb mikroprocesszor-típushoz kifejlesztettek saját, a processzorhoz optimálisan illeszkedő perifériaillesztő elemkészletet.

A táblázatban a világ talán három legnépszerűbb 8 bites mikroprocesszorához kapcsolódó család elemeit soroljuk fel.

A Z80 és 8080/8085-ös közös eredete és a részleges kompatibilitása lehetővé tette, hogy a két család elemeit akár keverve is használjuk. A gyakorlatban általában a Z80-as processzorhoz alkalmazzuk az Intel család elemeit.

Sajátos „karriert” futott be a Motorola cég display-vezérlője. A jól megtervezett, sokfunkciós és színeket is kezelő áramkör igen népszerűvé vált: ezt alkalmazzák az IBM PC színes grafikus adapterében (CGA), valamint sok 8080 és Z80 alapú mikroszámítógépes rendszerben, mint a Videoton TVC-ben is.

Igazán általános célú periféria-vezérlők az úgynevezett egytökös mikroszámítógépek felhasználásával hozhatunk létre. A következőkben összefoglaljuk a velük kapcsolatos legfontosabb ismereteket.

Funkció	18080/8085	Z80	Motorola
Párhuzamos be- és kimenet	8255 (24 bit)	Z80—PIO (16 bit)	MC6820 (16 bit)
Soros be- és kimenet	8251 USART	Z80—SIO	MC6850 (ACIA)
TV/Display vezérlő	8275	Z80—VCU	MC6845
DMA	8257	Z80—DMA	MC6844
CTC	8253, 8254	Z80—CTC	MC6840
Megszakításvezérlő	8259, 8214	—	MC6828
Lemezmeghajtó vezérlő	8271	—	MC6843

EGYTOKOS MIKROSZÁMÍTÓGÉPEK

Az egytocos mikroszámítógépek a számítástechnika és az integrált áramkörti technológia együttes fejlődésének köszönhetőek létezésük. Felépítésükben sok a közös a mikroprocesszorokkal és a kapcsolódó egyéb áramkörökkel, de néhány tulajdonságuk — különösen be/kimeneteik — az adott feladat körülményeitől függenek. Az egytocos mikroszámítógépeknek nagy jelentőségük van a műszaki alkalmazásokban. Ezeket használják az elektronikus játékokban, automata mosógépekben, autókban, televíziókban és a számítógépek perifériális berendezéseiben.

Általánosan fogalmazva: az egytocos mikroszámítógép a számítógépes részegységek egy tokba való integrálását jelenti. Ezek a részegységek a következők.

Csak olvasható memória (ROM vagy PROM, EPROM)

A ROM egy fix tartalmú, nem törlődő memória. A programtárba kerülő program a tok előállításakor írják be a tárolóba. A gyártók a felhasználó által programozható memóriával (PROM, EPROM) ellátott mikroszámítógépeket is forgalmazznak.

Írható-olvasható memória (RAM)

A RAM memória a program változóinak és az adatoknak a tárolására szolgál. Nagysága a típustól függ.

Központi egység (CPU)

A CPU megegyezik a mikroprocesszorban alkalmazható központi egységgel. A CPU legtöbbször képes be/kimeneti vonalnak és a memória egyes bitjeinek tesztelésére, beállítására és törlésére, mivel az alkalmazások során sokszor szükség van egy-egy kimeneti vonal állapotának megváltoztatására vagy egy-bemeneti vonal beolvasására. Ezek általában a külső, kétállapotú készülékekhez, kapcsolókhöz, termosztátokhoz, szilárdtest-relekhez, szelepekhez, motorokhoz kapcsolódnak.

Párhuzamos be/kimenetek

A legtöbb típusban az adott be/kimeneti vonal bemenetként vagy kimenetként is használható, ami a vonalak rugalmas felhasználását teszi lehetővé. Bizonyos be/kimeneti vonalak közvetlenül alkalmaznak nagyobb áramú vagy speciális igényű meghajtásra; ilyenek például a fluorescens kijelzők.

Soros be/kimenet

A terminálokkal való soros kapcsolat megszokott megoldás, amely kevés összekötő vonalat igényel. A soros kapcsolatot speciális áramkörök vagy több mikroszámítógép összekapcsolására is szokták használni. Az ismert aszinkron és szinkron adatátviteli megoldások olyan adatátviteli előírást — protokollt — igényelnek, amelyekben benne van a start és stop információ is. Ez hardverben is megvalósítható az U(S)ART (Universal Synchronous Asynchronous Receiver/Transmitter = Univerzális

Szinkron/Aszinkron vevő/adó) áramkörként, tehermentesítve a processzort és a programozót ettől a feladattól. Így csupán az adatátviteli sebesség (Baud-rate) és a megfelelő opciók (stop bitek száma, paritás stb.) kiválasztása szükséges, a megfelelő értékeknek a soros adó vagy vevő vezérlésgiszterébe írásával. A megfelelő formátumú soros adat előállítását ezek után a hardver áramkör végzi és kezeli.

Időzítő/számláló egységek

Az egytocos mikroszámítógépek alkalmazása sok esetben valós idejű (real time) végrehajtást kíván. Ez megoldható a végrehajtási időt és a programhurkok gondos egyeztetésével, ami azonban a legegyszerűbb feladattól eltekintve, nem hatékony. Megoldásként leggyakrabban időzítő áramköröket, timereket alkalmaznak. Az ilyen áramkörök a program futásától függetlenül számolják az eltelt időt, és megszakítást generálnak, ha egy előzőleg beprogramozott időtartam eltelt. Ezek az időzítők adott értékkel feltölthető számlálók. Az időzítő lefelé számol, és amikor eléri a nullát, egy jelzőbitet állít be, vagy megszakítást generál. Bonyolultabb időzítők képesek ezután az eredeti értéket újra betölteni és ismét számolni. Ilyképpen tehermentesítik a programozót az ismételt újratöltés és az időzítő újraindításának a programozásától, ami a folyamatos, pontos időközönkénti megszakításgeneráláshoz például óránál szükséges. Néha az időzítőt eseményszámlálónak használják, amihez egy külön be-

menet kell, és a használat a külső esemény létrejötte lépteti.

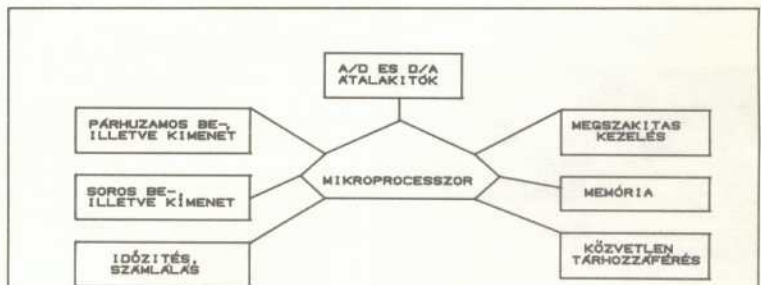
Az időzítést beállító elemek

A legtöbb egytocos mikroszámítógép órajel-generátora egyszerű időzítő elemekből épül fel. Ha a mikroszámítógépet maximális órajel-frekvenciával akarjuk működtetni és a nagyfokú pontosság nélkülözhetetlen, kvarckristályt használunk. Sok órajel-generátor csak ellenállással és kondenzátorral is működik, vagy külső órajelet alkalmazhatunk. Ez az utóbbi megoldás akkor hasznos, ha a mikroszámítógép külső szinkronizációt is igényel. Az ilyen felépítésű perifériavezérlők már a legújabb generációknak tekinthetők. Segítségükkel az új, modern felépítésű mikroprocesszorok rendszerekben megjelenő feladatok oldhatók meg, amelyek közül most csupán kettőtől, mintegy illusztrációként teszünk említést.

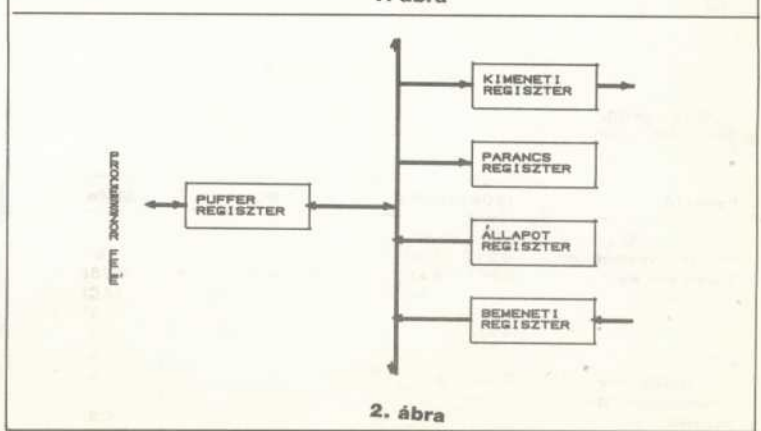
Az új típusú, nagy — millió bájtt — címtartományú mikroprocesszorok és a korszerű operációs rendszerek alkalmazásához elengedhetetlen a nagy számítógépekhez már megszokott tárkezelési forma. Az erre a célra kifejlesztett egység az MMU (Memory Management Unit: tárkezelő egység) a memória változtatható felosztását (szegmentálását), dinamikus áthelyezését (relokálását) és a kijelölt szegmensek védelmét látja el.

Többprocesszoros rendszerekben a sírrendszer osztott idejű felhasználásából adódó konfliktushelyzeteket oldja meg a BAU, a Bus Arbiter Unit, vagyis a sínvezérlő egység.

Dr. Könyö László



1. ábra



2. ábra



COBRA

ELEKTRONIKAI ÉS SZOLGÁLTATÓ KISSZÖVETKEZET
1097 Budapest, IX. Illatos ut 7. Telefon: 476-160/388

KISSZÖVETKEZETEK!

Egyedülálló kínálatunk:	
Számlakészítő program	19 900,— Ft
Számlanyilvántartó program	24 900,— Ft
Bér- és jövedelem-számfejtő program	24 900,— Ft
Főkönyvi könyvelőprogram	44 900,— Ft
	114 800,— Ft
COBRA—CONTO programcsomag	99 000,— Ft
IBM—XT kompatibilis számítógép	169 000,— Ft
STAR LC—10 nyomtató	49 000,— Ft
	317 000,— Ft

helyett mindezt már
299 000,— Ft-ért is megvásárolhatja!

TUTTI

ELECTROCOOP
KISSZÖVETKEZET

- IBM PC kompatibilis gépek
- HARDVERTELEPÍTÉS SZERVIZ ÉS GARANCIA
- SZOFTVERES TÁMOGATÁS
- RÖVID HATÁRIDŐ

Cím: Bp., Üllői út 81. 1091
Tel.: 334-354



ECOSOFT
Számítástechnikai
Szolgáltató
Kisszövetkezet

IBM PC/AT
kompatibilis számítógép
ár: **229 000 Ft-tól**
TURBO/32

32 bit, 24 MHz
ár: **479 000 Ft-tól.**
Az általunk forgalmazott
eszközök lizingelhetők is.
Tel.: 863-677



IRODA:
VI., Nagymező u. 51.
TEL.: 325-768

ADATÁTVITELI
RENDSZEREK
HARDVER/SZOFTVER
GRAFIKUS MUNKAHELY
ADATRÖGZÍTÉS
SZERVIZ



Telefon:
415-166

Kereskedelmi és Szoftveriroda
1061 Bp., Liszt F. tér 10.
Telex: 22-4378

- **ASY—16 szupermikro számítógép**
 - 12 terminál
 - VME busz
 - UNIX

— **CRT TERMINÁLOK**
VT—52, QUT—102, Siemens
8160

- **BILLENTYÜZETEK**
- **MONITOROK**
- **INTEGRÁLT VÁLLALATI INFORMÁCIÓS RENDSZER**

UNIX környezetben üzemeltethető.



PERIFÉRIA

Elektronikai
Fejlesztő
és Szolgáltató
Kisszövetkezet
Bp. VII., Peterdy u. 30.
Telefon: 213-588

ajánlata:

- **P—XT:** 140 E Ft-tól + áfa
- **P—AT:** 200 E Ft-tól + áfa
- igény szerinti konfigurációk
- **FX—1000 PRINTER**
75 E Ft + áfa

procontrol



Kisszövetkezet
megnyitotta

COMPUTER SZAKÜZLETÉT

- hardver, szoftver
- elektronikai elemek
- integrált áramkörök számítógépek, perifériák
- blokkolóórák
- vonalkódeszközök, biztonsági rendszerek

SEGED
Kazinczy u. 8. Tel.: 62/12-259
Berzsenyi u. 2. Tx.: 82-726

2,2 m-es parabola-
antenna

Tel.: 323-332
TT—2164

FALIVEZETÉK-KUTATÓ

Tel.: 314-575
ERIKA
İRÓGÉP

Bp. VI.,

Népköztársaság útja 2.



1146 Bp., AJTÓSI DÜRER
SOR 10.
Levélcím:
1393 Pf.: 319.
Telefon: 421-974
Telex: 22-6544

február havi kínálata

a 80 386-os
mikroprocesszorra
alapozott, multiuser
üzeműdű

ACER SYS 32

rendszer, amely
széleskörűen,
változtatható kiépítésben,
alap és alkalmazói
szoftverekkel együtt
kapható.

10th Annual
Volume 6 Number 6

COMDEX **4**

SHOW DAILY

The Greatest Show Daily on Earth. Published by THE INTERFACE GROUP, Inc.
Thursday, November 17, 1988

★ ★ **110,000...PLUS** ★ ★

A LEGNAGYOBB BEMUTATÓ

Honnan indult és merre tart ez a kiállítás?

A kiállítást a THE INTERFACE GROUP cég szervezte. A cég története egyrészt egy jellegzetes mai, amerikai karriertörténet, másrészt mikroszámítógép történelem is.

Sheldon G. Adelson különféle üzleti vállalkozásai során 1971-ben vásárolt egy kis számítógép-ipari kiadványt. Rájött, hogy „egy kiállítás olyan élő kiadvány, ahol a hirdető, kiállító, olvasók, látogatók és az írott szavak a konferencián előadottaknak felelnek meg”. Az elgondolást két év múlva az INTERFACE kiállítással valósította meg először. Ez az adatátvitellel foglalkozó kiállítás akkor csak harmincöt kiállítót és ezeröttszáz látogatót vonzott. Bár ez az első kiállítás ráfizetéses volt, Adelson nem tájított — bízott a számítástechnika világában —, folytatta.

Hat évvel később Adelson összehívott egy szakértőkből álló csoportot, akiknek előadta, hogy a mikroszámítógép-ipar részére szervez egy kiállítást. Az ötletet meglehetősen hűvösen fogadták. Végül azt mondták, hogy gondolkozzunk ezen még egy évet, majd talán jövőre. Adelson azonban nem hagyta magát lebeszélni, mert attól tartott, hogy más megelőzi.

Miért is tartott ettől? — Elég volt áttekintenie a számítástechnika akkori jelenét és jövőjét. Ekkor már négy éve létezett a West Coast Computer Faire (Nyugati Parti Számítógép Kiállítás) San Franciscóban, két éve a Northeast Computer Faire (Észak-keleti Számítógép Kiállítás) Bostonban, tehát már voltak mikrogépes kiállítások, de ezek az amatőrök részére jöttek létre, nem

pedig a kereskedőknek, az iparnak, a fejlesztőknek, és persze a közönségnek is. Még nem lépett a porondra az IBM, még nem voltak milliós példányszámú gépek, de már a Tandy cég itteni eladásai elérték a negyedmilliárd dollárt, az Apple már százmillió dolláros cég lett. Adelson a növekedésből a jövőbeni nagy üzletet vette észre, és feltételezte, hogy más is.

A COMDEX (Computer Distribution EXposition — Számítógépterjesztők Kiállítása) sorozat 1979-ben egyetlen szállódban kezdődött összesen

A tizedik Las Vegas-i COMDEX mindenben felülmúlta az eddieket. (1700 kiállító, 110 000 látogató, 8 kiállítási csarnok és szálló, 85 ország, 52 szekció, 1000 kiállítási dolgozó, 1800 szállítókész, 650 taxi, 60 000 szállodai szoba, 70 ingyenes szállítóbusz, 1000 rendőr és biztonsági dolgozó, 1000 segítő kiállítási dolgozó, 1300 (!) ideiglenes telefonvonal, 138 millió dollár bevétel Las Vegasnak.)

157 kiállítóval és 4000 látogatóval. — Érdekességként jegyzem meg, hogy a cégtulajdonos a kiállítás jobb szervezése érdekében később megvásárolt egy utazási ügynökséget, egy kis légitársaságot — amelyből kifejlesztett egy tengeren túli utazási vállalatot is —, idén pedig megvett egy nagy Las Vegas-i szállót, amelynek telkén fel akarja

1. kép





2. kép

építeni a világ legnagyobb szállodáját hatezeröttszáz szobával, a szállodához tartozó 100 000 négyzetméteres kiállító csarnokkal és 10 000 négyzetméternyi konferencia-teremmel.

A cég együtt nőtt a kiállításokkal, csakúgy mint a mikroszámítógép-iparral. Jelenleg hatszáz dolgozója van öt országban.

A COMDEX/FALL történetét számokban az 1-4. ábrákon mutatom be. Látható, hogy 1983-ig gyors növekedés volt jellemző, de — a külföldi kiállítók számától eltekintve — ezt követően 2-3 éves visszaesés következett, majd újra erős növekedés. Miért? — Az erős növekedésnek a mikrogépek gyors terjedése a magyarázata. A visszaesésnek pedig az, hogy ebben az időben azoknak a vásárlóknak a száma, akik a gépeket csak játékokra használták, már nem nőtt, míg a profi felhasználók száma még nem volt elég nagy. Azóta viszont az utóbbiak váltak uralkodóvá és egyre csak gyarapodnak.

A kiállítók, látogatók tájékoztatására öt éve jelent meg először a jelenleg 176 színes, nagyalakú lapon (!) megje-



3. kép

lenő, a COMDEX alatt kapható napilap, a Show Daily, és az utóbbi két évben egy saját tv-adó(!) is.

Miért olyan népszerű ez a kiállítás? Kétségtelen, hogy a számítástechnika népszerűsége állandóan nő, de akkor miért buktak meg mások, miért lett ez a kiállítás népszerűbb, mint például a korábban kezdett, és akkor legnépszerűbb West Coast Computer Faire? Elsősorban a kitűnő helyszínválasztás miatt. Mitől jobb ez a helyszín, mint az előző? A magyarázatot a hátsó borítónk képei adják. — A kaszinók ebben a városban mindent meghatároznak, minden ezekért történik. A szállodák árai például rendkívül alacsonyak. Igen olcsók a szállodák vendéglői is. A legolcsóbb komplett reggeli mintegy 25 Ft-nak megfelelő összeg, míg a legolcsóbb ebéd ára hozzávetőlegesen 130 Ft. A repülőjegyek ára is viszonylag alacsony. Mindez azért, hogy az emberek menjenek oda, töltsenek el néhány napot és játsszanak. És az emberek mennek... és játsszanak. Bár az ottani lapok tele voltak azzal, hogy valaki félmilliót nyert, de ettől függetlenül egyre csak újabb és újabb szállodákat építenek, a régiakat bővítik, tehát jól megy a szállodák tulajdonosainak, s biztos, hogy a játékosok többet veszítenek, mint amennyit nyernek.

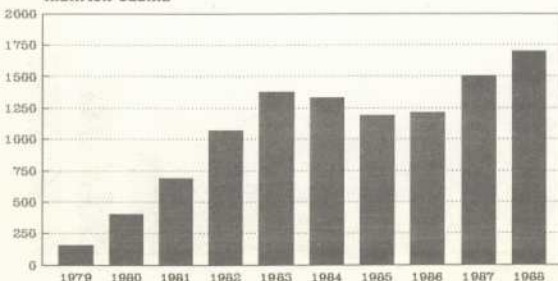
S a kiállítás? — A hátsó borító utolsó képen és e cikk 1-2. képen — melyek a kiállítás két fő helyszínén készültek —, látható, hogy minden nagy cég megjelent. (A két fekete-fehér képen az is, hogy a felkapottabb helyeken nagy volt a tömeg.) A belső borító

első két képen látható, hogy milyen minőségű képeket lehet előállítani a NEC Multisync 2A és MACSync monitorain. (Az elsőn egyébként a Las Vegas környéki Spring Mountain — Tavasz Hegység látható.) Ugyanezen oldal *első két képen* egy közönséges nyomtatóból az Eotron cég EOGraph nevű rendszere által plotterre alakított eszközzel készített két kép van. A rendszer egyetlen, az IBM PC/XT/AT és a kompatibilis gépekhez használható hosszú kártyából és szoftverből áll. A kártyán 0.5-1M RAM, négy soros csatorna, ROM-ba égetett szoftver van, amellyel emulálják a Hewlett-Packard HP-GL-, és Houston DM/PL-plotternyleveit. Így a számítógép a nyomtatót plotterként „látja”. A rendszer egyik különleges előnye, hogy sokkal gyorsabb, mint a plotterek. A nyomtatókkal szembeni előnye a nagy rajzterület — például öt méter hosszú is lehet —, illetve a rendkívüli felbontás, mintegy húsz pont milliméterenként.

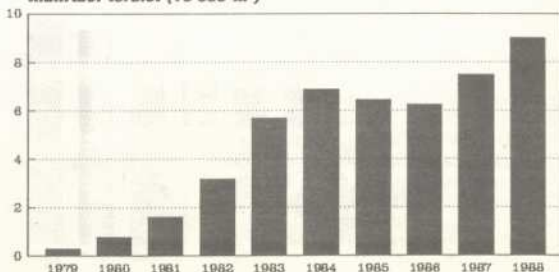
A kiállítás és a konferencia öt napig tartott, de ez az idő csak a kiállítás futó áttekintésére és egy-egy érdekesebb rész alaposabb megnézésére volt elég. A konferenciára, bármilyen csábító külsejű robot csalogatót (hátsó borító belső oldal *legfelső kép*), alig maradt idő.

Az említett két hardverhez is kapcsolhatók a grafikus szoftverek a Macintosh gépekhez készítette az Informix cég a WING Z nevű táblázatkezelőjét. A rendkívül rugalmas, egyszerűen kezelhető színes szoftver egy-egy

Kiállítók száma



Kiállítási terület (10 000 m²)



dalon található **többi képen** látható háromféle, külön-külön is igen szemléletes grafikus megjelenítését az eredménynek. Erre mondhatjuk azt a szójátékok, hogy: A SZÁRNY Z szárnyakat ad az Ön Macintosh-ának (The WING Z gives wings for your Mac!)

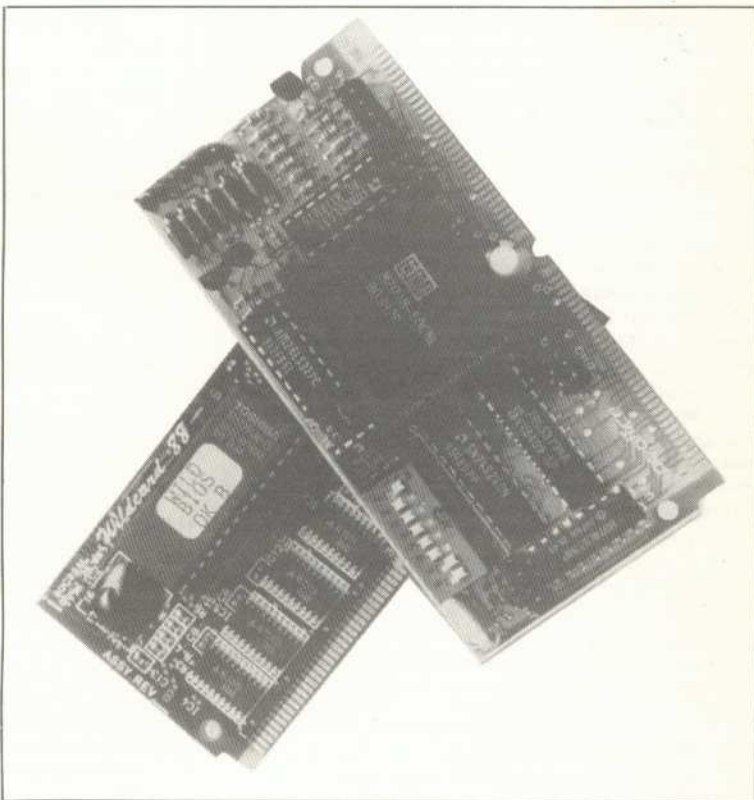
Az IBM 625, más cég által a PS/2-höz tervezett eszközről tett említést. Legjelentősebb ezek közül a Hewlett-Packard és az AT&T által kiállított OS/2 LAN Manager. Ez azt jelenti, hogy az IBM engedélyezte ezt a tevékenységet. A COMDEX Show Daily erre a következő kérdést tette fel: Ez a gláznoszty nemsokára peresztrojkiát eredményez az IBM-nél? (Vagyis azt remélik, hogy az IBM engedni fog a termékeit védő merev stratégiájából.)

A cikksorozatban eddig nem volt szó az ún. POS-ról, azaz a Point of Sale pénztárgép rendszerről, ami rendelésnyilvántartási, eladás/vásárlás könyvelési, napi eladáslisztázási, pénzkezelési, raktározási, számlakészítési alrendszerek összessége. A 3. és 4. képen az egyik ilyen szoftverárúsító cég (CASHTRAX) érdekes reklámja látható. A szoftvert PC/XT rendszerhez optikai kódolvasóval és egy bizonyos pénztárgéptípushoz tervezték.

Ezen, de mint több, más területen is, alapvető változást hoz az 5. képen látható felső kártya. Ez az INTEL cég Wildcard 88 technológiájával készített, és az XT alapkártyáját — a memória nélkül — helyettesítő kártya. A mindössze ötven négyzetcentiméter felületű kártya (a méretre jellemző, hogy a kártyacsatlakozón egy osztásköz — a két szomszédos vonal közti távolság — mintegy kettőegész öttized milliméter), ami a másikkal együtt egy teljes LAN (Local Area Network helyi hálózat)-kártyát és terminált helyettesíti. Így két ilyen kicsi kártyát a pénztárgépbe telepítve, azokat összekapcsolva egy hálózat hozható létre. Várhatóan ezt a kártyát alkalmazzák majd a laboratórium- és folyamatirányításban is, ahol ezek a kisméretű, nagy kapacitású kártyák intelligens irányítórendszerek helyi központi egységei lehetnek.

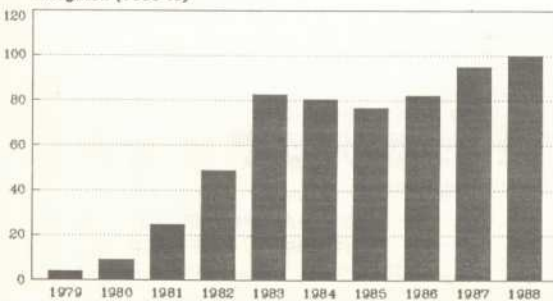
Simonyi Endre és Simonyi Márton

4. kép

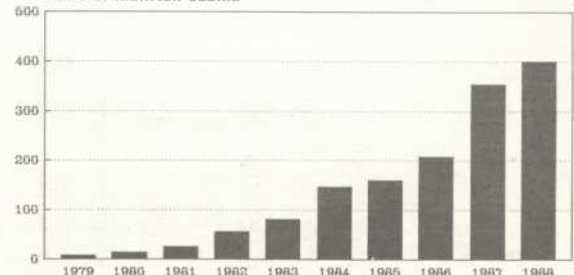


5. kép

Látogatók (1000 fő)



Külföldi kiállítók száma



A MOTOROLA 68000-ES

MIKROPROCESSZOR ASSEMBLY PROGRAMOZÁSA

A Motorola cég által 1979-ben kifejlesztett M68000-es mikroprocesszor alkalmazóinak tábora a chip megjelenése óta világszerte nő. Így nálunk is, hiszen egyre több Commodore Amiga, Atari ST, Apple Macintosh típusú mikroszámítógép kerül be az országba. És a Sinclair QL-tulajdonosoknak is érdekes lehet az alábbi leírás, hiszen a QL M68000-as központi egysége szoftver szinten teljesen kompatibilis az M68000-rel.

Ennek a mikroprocesszornak 16 bites adatbusza, 24 bites címbusza és 32 bites regiszterei vannak. Teljesítőképessége a Zilog cég Z8000-es processzorával csaknem azonos szintű. Mikroprogramozott felépítésű. A fejlesztők a 68000-es bővíthetőségét, átalakíthatóságát tartották a legfontosabbnak.

Az M68000-es szoftver szempontjából legfontosabb paraméterei a következők:

- 32 bites regiszterek,
- 16 Mbájtos, közvetlenül címezhető memóriataromány,
- 14 címzési mód,
- 56 utasítástípus (a teljes utasításkészlet a 14 címzési mód miatt 1000-nél több elemű!),
- 5 fő adattípus,
- memóriába ágyazott B/K,
- valódi kétcímű processzor (vagyis akár mindkét operandus a memóriában lehet).

Az M68000-es regiszterkészlete (lásd az ábrát):

D0—D7: adatregiszterek

A0—A6: címregiszterek

USP (User Stack Pointer): felhasználói veremmutató

SSP (Supervisor Stack Pointer): supervisor veremmutató

SP (Stack Pointer): az aktuális veremmutató (tulajdonképpen ez lenne az A7-es címregiszter)

PC (Program Counter): programszámláló (vagy ha jobban tetszik: utasításslámláló)

SR (Status Register): állapotregiszterek (flagok)

CCR (Condition Code Register): az állapotregiszter alsó, felhasználói bájttja

Az állapotregiszter (SR) alsó, úgynevezett felhasználói bájttja (CCR) a következő flageket tartalmazza:

0. bit: Carry (C) — átvitelkapcsoló

1. bit: overflow (V) — túlszordulás-kapcsoló

2. bit: Zero (Z) — zérókapcsoló

3. bit: Negative (N) — negatívkapcsoló

4. bit: eXtension (X) — bővítőkapcsoló

5—7. bitek: nem használt (értékük 0)

Az állapotregiszter (SR) felső, ún. rendszerbájttjának felépítése:

8—10. bitek: Interrupt mask (I0—I2) — megszakítási maszk

11., 12. bitek: nem használt (értékük 0)

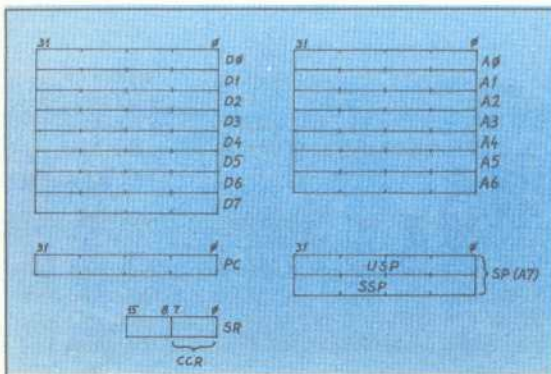
13. bit: Supervisor (S) — felügyelői állapot

14. bit: nem használt (értéke 0)

15. bit: Trace mode (T) — nyomkövetési üzemmód

A rendszerbájt csak felügyelői állapotban módosítható közvetlen úton! Az M68000-es az S flag értékétől függően felhasználói vagy felügyelői üzemmódban működik. Felügyelői állapotban a processzor valamennyi utasítása végrehajtható, míg felhasználói állapotban nem. Az ilyen állapotfüggő utasítások az úgynevezett privilegizált utasítások (mint például az egész állapotregisztert módosító utasítások).

A processzor a T flag 1-re állításával nyomkövetési üzemmódba állítható, ami azt jelenti, hogy egy tesztelendő program viselkedése lépésenkénti végrehajtással nyomon követhető.



Az M68000-es ötféle adattípust ismer:

- bit,
- BCD számjegy (4 bit),
- bájt (8 bit), jele: B
- szó (16 bit), jele: W (Word)
- kettős szó (32 bit), jele L (Long word).

Az M68000-es mikroprocesszor a következő címzési módokat használja:

Direkt regisztercímzés

- adatregiszter direkt címzése: az effektív címet egy adatregiszter (D0—D7) tartalmazza
- címregiszter direkt címzése: az effektív címet egy címregiszter (A0—A7) tartalmazza

Abszolút címzés

- abszolút rövid címzés: az effektív cím az utasítás kódja után áll, és szó hosszúságú
- abszolút hosszú címzés: az effektív cím az utasítás kódja után áll, és kettős szó hosszúságú

PC-relatív címzés

- relatív címzés ofszettel: az effektív cím a PC tartalmának és az utasításkód után álló szó hosszúságú ofszetnek az összege
- relatív címzés ofszettel és indexszel: az effektív cím a PC tartalmának, az utasításkód után álló, bájt hosszúságú ofszetnek és az indexregiszter tartalmának az összege

Regiszter indirekt címzés

- címregiszter indirekt: a címregiszter arra a címre mutat, ahol az effektív cím található
- címregiszter indirekt a címregiszter utólagos növelésével: ugyanaz, mint a regiszter indirekt, azzal a kiegészítéssel, hogy befejezésésként a címregiszter tartalmához hozzáadódik az operandus bájtokban vett hossza
- címregiszter indirekt a címregiszter előzetes csökkentésével: ugyanaz, mint a regiszter indirekt mód, de itt az utasítás végrehajtása előtt a címregiszter tartalmából kivonódik az operandus bájtokban vett hossza
- címregiszter indirekt ofszettel: a címregiszter és az utasításkód után álló bájt hosszúságú ofszet összege arra a címre mutat, ahol az effektív cím található
- indexregiszter és címregiszter indirekt ofszettel: az indexregiszter, a címregiszter és az utasításkód után álló bájt hosszúságú ofszet összege arra a címre mutat, ahol az effektív cím található

Közvetlen címzés

- közvetlen: az operandus közvetlenül az utasításkód után áll
- gyors közvetlen: az operandust maga az utasításkód meghatározza

Implicit címzés

- az operandust az utasítás implicit módon tartalmazza (például egy veremkezelő utasítás implicit módon tartalmazza a veremmutatót)

A processzor működése során úgynevezett rendkívüli események következhetnek be, amelyek oka lehet címzésihiba, nullával való osztás, közvetlen utasítások, nyomkövetési üzemmód, megszakítás, buszhiba vagy reset. Ilyen esetekben a processzor — sajátos elnevezésű — kizárási (exception) állapotba kerül. A kizáráskor a supervisor verembe kerül az állapotregiszter és a programszám-láló tartalma, majd a supervisor bit 1 értékre áll és a nyomkövető bit visszaállítódik 0-ra. Módosulhat a megszakítási maszk is. A PC új értéke a megfelelő, úgynevezett kizárási vektor lesz. Ezeket a vektorokat külön táblázatban ismertetem.

Az utasításkészletet ábécé sorrendben tartalmazó táblázat nem a tanuláshoz nyújt segítséget, hanem csupán a mindennapos programírást megkönnyítő, gyors áttekintés.

A táblázat jelöléseinek értelmezése:

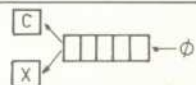
"ea"	effektív cím
Dx, Dy	adatregiszter
Ax, Ay	címregiszter
Rx, Ry	adat- vagy címregiszter
# "data"	közvetlen adat
- / /	előzetes csökkentés
/ / +	utólagos növelés
. X	az operandus hossza (X helyére B, W vagy L helyettesíthető be)
?	a flag állapota nem definiált
↑	a flag beáll az utasításnak megfelelően
-	a flag állapota nem változik

A kétoperandusú utasítások mnemonikja után mindig az első (bal oldali) operandus a forrás operandus, a második (jobb oldali) pedig a cél operandus!! Ezt leginkább azok vessék jól az eszükbe, akik eddig Intel vagy Zilog mikroprocesszorokat használtak.

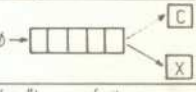
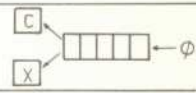
Leírásomban — a korlátozott terjedelem miatt — nem adhattam részletes információkat. Céлом csupán az volt, hogy a kezdőknek felvázoljam a 16 bites mikroprocesszor igen jó képességeit, az alkalmazók munkáját pedig egy jól használható segéd-eszközzel — a táblázatokkal — megkönnyítsem.

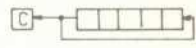
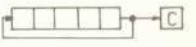
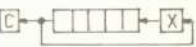

Tóth Zoltán


Szintaxis	Név	Operand.hossza	X	N	Z	V	C	Megjegyzés
ABCD Dx, Dy ABCD -(Ax), -(Ay)	Add BCD with extend	B	†	?	†	?	†	BCD összeadás az extend flag hozzáadásával.
ADD.X Dx, "ea" ADD.X "ea", Dx	Add binary	B,W,L	†	†	†	†	†	Bináris összeadás.
ADDA.X "ea", Ax	Add binary Addressreg.	W,L	-	-	-	-	-	Bináris összeadás egy címregiszterrel.
ADDI.X# "data", "ea"	Add Immediate	B,W,L	†	†	†	†	†	Bináris összeadás közvetlen adattal.
ADDQ.X# "data", "ea"	Add Immediate Quick	B,W,L	†	†	†	†	†	A konstans max. 3 bites lehet. Gyors összeadás közvetlen adattal.
ADDX.X Dx, Dy ADDX.X -(Ax), -(Ay)	Add binary with eXtend	B,W,L	†	†	†	†	†	Bináris összeadás az extend flag hozzáadásával.
AND.X Dx, "ea" AND.X "ea", Dx	logical AND	B,W,L	-	†	†	φ	φ	Logikai ÉS.
ANDI.X# "data", "ea"	logical AND with Immediate value	B,W,L	-	†	†	φ	φ	Ha az operandus W vagy az SR, privilegizált utasítás!
ASL.X Dx, Dy ASL.X# "data", Dx ASL.X "ea"	Arithmetic Shift Left	B,W,L	†	†	†	†	†	Dy-t Dx helyel Dx-et "data" helyel "ea"-t 1 helyel
ASR.X Dx, Dy ASR.X# "data", Dx ASR.X "ea"	Arithmetic Shift Right	B,W,L	†	†	†	†	†	Dy-t Dx helyel Dx-et "data" helyel "ea"-t 1 helyel
Bcc "címké"	Branch if Condition - Codes are true	B,W						T (true) 1 F (false) φ HI (high) $\bar{C} \bar{Z}$ LS (low or same) C+Z CC (carry clear) \bar{C} CS (carry set) C NE (not equal) \bar{Z} EQ (equal) Z VC (overflow clear) \bar{V} VS (overflow set) V PL (plus) \bar{N} MI (minus) N GE (greater or equal) $NV + \bar{N}V$ LT (less than) $N\bar{V} + \bar{N}V$ GT (greater than) $NV\bar{Z} + \bar{N}VZ$ LE (less or equal) $Z + NV + \bar{N}V$
BCHG Dx, "ea" BCHG# "adat", "ea"	test Bit and CHanGe it	B,L	-	-	†	-	-	Az első operandus a vizsgálandó és negálandó bit számát tartalmazza.
BCLR Dx, "ea" BCLR# "adat", "ea"	test Bit and CLear it	B,L	-	-	†	-	-	Az első operandus a vizsgálandó és törölendő bit számát tartalmazza.
BRA "címké"	BRAnch	B,W	-	-	-	-	-	Relatív ugrás.
BSET Dx, "ea" BSET# "adat", "ea"	test Bit and SET it	B,L	-	-	†	-	-	Az első operandus a vizsgálandó és 1-be állítandó bit számát tartalmazza.
BSR "címké"	Branch to SubRoutine	B,W	-	-	-	-	-	Relatív ugrás szubrutinra.
BTST Dx, "ea" BTST# "adat", "ea"	Bit TeST	B,L	-	-	†	-	-	Az első operandus a vizsgálandó bit számát tartalmazza.
CHK.X "ea", Dx	CHeck register	W	-	†	?	?	?	Ha a Dx φ és az "ea" címen lévő értéken kívül van, akkor TRAP #φ6 jön létre.



Szintaxis	Név	Operand. hossza	X	N	Z	V	C	Megjegyzés
CLR.X "ea"	CLeaR	B, W, L	-	Φ	1	Φ	Φ	Az operandus minden bitjét Φ-ra állítja.
CMP.X "ea", Dx	CoMPare	B, W, L	-	↑	↑	↑	↑	Kivonást végez, de nem írja felül az operandusokat.
CMPA.X "ea", Ax	CoMPare Addressreg.	W, L	-	↑	↑	↑	↑	CMP az egyik címregiszterrel.
CMPI.X # "adat", "ea"	CoMPare Immediate	B, W, L	-	↑	↑	↑	↑	CMP egy közvetlen adattal.
CMPM.X(Ax)+, (Ay)+	CoMPare in Memory	B, W, L	-	↑	↑	↑	↑	CMP a memóriában lévő operandusok között.
cpGEN "paraméter"	CoProc. GENERAL Comm.	?	?	?	?	?	?	Vezérlésátadás társprocesszornak.
DBcc Dx, "címké"	Decrement and Branch if Condition Codes are true	W	-	-	-	-	-	Relatív ugrás, ha a feltétel teljesül és Dx a dekrementálás után ≠ -1 (Lásd még: Bcc)
DIVS "ea", Dx	DIVide Signed	W	-	↑	↑	↑	Φ	$Dx = \frac{Dx(L)}{ea(W)}$ Előjeles osztás. Ha "ea" = Φ, akkor → TRAP# Φ S Ha "ea" > Dx, akkor → V = 1
DIVU "ea", Dx	DIVide Unsigned	W	-	↑	↑	↑	Φ	Ugyanaz, mint DIVS-nél, de ez előjel nélküli osztás.
EOR.X Dx, "ea"	logical Exclusive OR	B, W, L	-	↑	↑	Φ	Φ	Logikai KIZÁRÓ-VAGY. (EOR, XOR)
EORI.X # "adat", "ea"	logical EOR with Immed	B, W, L	-	↑	↑	Φ	Φ	EOR egy közvetlen adattal.
EXG Rx, Ry	EXchanGe registers	L	-	-	-	-	-	Két tetszőleges regiszter tartalmának cseréje.
EXT.X Dx EXTB.L Dx	Sign EXtend	W, L	-	↑	↑	Φ	Φ	B-ről W-re, vagy W-ről L-re terjeszti ki az operandus méretét, és vele együtt az előjelét.
ILLEGAL	ILLEGAL instruction	-	-	-	-	-	-	PC és SR → (SP), majd 4-es kizárás.
JMP "ea"	JuMP absolute	L	-	-	-	-	-	Abszolút ugrás.
JSR "ea"	JuMP to SubRoutine	L	-	-	-	-	-	Abszolút ugrás szubrutinra.
LEA "ea"	Load Effective Address	L	-	-	-	-	-	Címregisztert memóriából feltölt.
LINK Ax, #Localstack	LINK local basepointer		-	-	-	-	-	Ax → -(SP) SP → Ax SP + Localstack → SP
LSL.X Dx, Dy LSL.X # "adat", Dx LSL.X "ea"	Logical Shift Left	B, W, L	↑	↑	↑	Φ	↑	Dy-t Dx helyet Dx-et "adat" helyet "ea"-t 1 helyet
LSR.X Dx, Dy LSR.X # "adat", Dx LSR.X "ea"	Logical Shift Right	B, W, L	↓	↓	↓	Φ	↑	Dy-t Dx helyet Dx-et "adat" helyet "ea"-t 1 helyet
MOVE.X "ea", "ea"	MOVE source to dest.	B, W, L	-	↑	↑	↑	Φ	Adatmozgatás egyik címről a másikra.
MOVE "ea", CCR	MOVE source to CCR	W	↑	↑	↑	↑	↑	SR-nek csak az alsó byte-ját (CCR) állítja (a felső byte nem változik).
MOVE "ea", SR	MOVE source to SR	W	↑	↑	↑	↑	↑	Privilegizált utasítás.
MOVE SR, "ea"	MOVE SR to dest.	W	-	-	-	-	-	SR-t a megadott címre tölti.
MOVE Ax, USP MOVE USP, Ax	MOVE to or from USP if supervisor	L	-	-	-	-	-	Privilegizált utasítások.
MOVEA "ea", Ax	MOVE to Addressregister	W, L	-	-	-	-	-	Címregiszter feltöltése memóriából.



Szintaxis	Név	Operand. hossza	X	N	Z	V	C	Megjegyzés
MOVEM X "reg.listá,"ea MOVEM X "ea,"reg.lista	MOVE Multiple registers	W, L	-	-	-	-	-	W esetén minden regiszter előjelhelyesen kibővül 32 bitre (L formátumra).
MOVEP Dx, offs.(Ax) MOVEP offs.(Ax), Dx	MOVE to or from Peripherals	W, L	-	-	-	-	-	
MOVEQ # "adat," Dx	MOVE immediate Quick	L	-	†	†	φ	φ	Az "adat" max. 8 bites lehet. Közvetlen adat gyors töltése.
MULS "ea," Dx	MULTiply with Sign	W	-	†	†	φ	φ	Előjeles szorzás.
MULD "ea," Dx	MULTiply without Sign	W	-	†	†	φ	φ	Előjel nélküli szorzás.
NBCD "ea"	Negate BCD-byte	B	†	?	†	?	†	Az operandus és az X bit nullából kivonásra kerül.
NEG X "ea"	NEGate	B, W, L	†	†	†	†	†	2-es komplementes képzése.
NEG X "ea"	NEGate with eXtend	B, W, L	†	†	†	†	†	Elsőként elvégző egy nullára vonatkozó vizsgálatot is!
NOP	No OPERATION	-	-	-	-	-	-	PC+2 → PC
NOT X "ea"	logical NOT	B, W, L	-	†	†	φ	φ	1-es komplementes képzése.
OR X Dx, "ea" OR X "ea," Dx	logical OR	B, W, L	-	†	†	φ	φ	Logikai VAGY.
ORL X# "adat," "ea"	logical OR with Immed.	B, W, L	-	†	†	φ	φ	Ha az operandus W, vagy az SR, akkor privilegizált utasítás.
PEA "ea"	Push Effective Address	L	-	-	-	-	-	A verembe a kiszámított effektív cím kerül.
RESET	RESET	-	-	-	-	-	-	124T ciklusig tart.
ROL X Dx, Dy ROL X# "adat," Dx ROL X "ea"	ROtate Left	B, W, L	-	†	†	φ	†	Dy-t Dx helytel Dx-et "adat" helytel "ea"-t 1 helytel 
ROR X Dx, Dy ROR X# "adat," Dx ROR X "ea"	ROtate Right	B, W, L	-	†	†	φ	†	Dy-t Dx helytel Dx-et "adat" helytel "ea"-t 1 helytel 
ROXL X Dx, Dy ROXL X# "adat," Dx ROXL X "ea"	ROtate through eXtend Left	B, W, L	†	†	†	φ	†	Dy-t Dx helytel Dx-et "adat" helytel "ea"-t 1 helytel 
ROXR X Dx, Dy ROXR X# "adat," Dx ROXR X "ea"	ROtate through eXtend Right	B, W, L	†	†	†	φ	†	Dy-t Dx helytel Dx-et "adat" helytel "ea"-t 1 helytel 
RTE	ReTurn from Exception	-	†	†	†	†	†	Privilegizált utasítás. (SP) → SR
RTR	ReTurn and Restore Condition Codes	-	†	†	†	†	†	RTE user-megfelelője. (SP)+ → CCR
RTS	ReTurn from Subroutine	-	-	-	-	-	-	Visszatérés szubrutinból.
SBCD Dx, Dy SBCD -(Ax), -(Ay)	Subtract BCD with extend	B	†	?	†	?	†	BCD kivonás az extend flag kivonásával.
SCC "ea"	Set byte according Condition Codes	B	-	-	-	-	-	Ha a feltétel teljesül: #FF → "ea" ha nem teljesül: φ → "ea" (Lásd még: Bcc)
STOP # "adat"	STOP with condition code loaded	B, W, L	†	†	†	†	†	Kilépés: RESET, vagy kizárás adat → SR útján lehetséges.

Szintaxis	Név	Operand. hossza	X	N	Z	V	C	Megjegyzés
SUB X Dx, "ea" SUB X "ea", Dx	SUBtract binary	B, W, L	↑	↑	↑	↑	↓	Bináris kivonás.
SUBA X "ea", Ax	SUBtract from Address	W, L	-	-	-	-	-	Bináris kivonás címregiszterrel.
SUBL X #adat, "ea"	SUBtract Immediate	B, W, L	↓	↑	↑	↑	↑	Bináris kivonás közvetlen adattal.
SUBQ X #adat, "ea"	SUBtract immed. Quick	B, W, L	↑	↑	↑	↑	↑	A konstans max. 3 bites lehet. Gyors bináris kivonás.
SUBX X Dx, Dy SUBX X -(Ax), -(Ay)	SUBtract binary with eXtend	B, W, L	↑	↑	↑	↑	↑	Bináris kivonás az extend flag kivonásával.
SWAP Dx	SWAP register halves	W	-	↑	↓	∅	∅	
TAS "ea"	Test byte And Set always bit 7	B	-	↑	↑	∅	∅	Vizsgálat: nullára és előjelre. A 7-es bitet 1-be állítja.
TRAP #vektor	software interrupt	-	-	-	-	-	-	Vektor: #∅∅. #∅F Kizárás kezdeményezése.
TRAPV	TRAP on Verflow	-	-	-	-	-	-	#∅7-es interrupt, ha V=1.
TST "ea"	TeST byte	B, W, L	-	↑	↑	∅	∅	Vizsgálat: nullára és előjelre.
UNLK Ax	UNLink local area	B, W, L	-	-	-	-	-	Ax → SP (SP)+ → Ax

EXCEPTIONS

Szám	Cím	Funkció
\$ ∅	\$∅∅∅	RESET esetén SSP (Supervisor Stack - Pointer) kezdőcíme
\$ 1	\$∅∅4	RESET esetén PC (Program Counter) kezdőértéke.
\$ 2	\$∅∅8	BUS ERROR: külső készüléktől jön, a BERR aktív L szintjével.
\$ 3	\$∅∅C	ADDRESS ERROR: ha W vagy L operandus esetén páratlan címet akarunk elérni.
\$ 4	\$∅1∅	ILLEGAL INSTRUCTION: nem \$Axxx vagy \$Fxxx kódú nemlétező utasítás, vagy nemlétező címzés mód. Megoldás: \$4AFC
\$ 5	\$∅14	DIVISION BY ZERO
\$ 6	\$∅18	CHK utasításnál Dx értéke az intervalumon kívülre esett.
\$ 7	\$∅1C	TRAPV utasítás (software megszakítás, ha V=1).
\$ 8	\$∅2∅	Privilegium megsértése: user üzemmódban privilegizált utasítást kapott a processzor.
\$ 9	\$∅24	TRACE rutin kezdőcíme: ha a trace-bit 1, akkor minden utasítás után erre a vektorra adódik a vezérlés.
\$ A	\$∅28	Nem implementált utasítás, melynek kódja: \$Axxx

Szám	Cím	Funkció
\$ B	\$∅2C	Nem implementált utasítás, melynek kódja: \$Fxxx.
\$ C	\$∅3∅	Nem használt, foglalt.
\$ D	\$∅34	Nem használt, foglalt.
\$ E	\$∅38	Nem használt, foglalt.
\$ F	\$∅3C	Nem inicializált megszakítás.
\$ 1∅- \$ 17	\$∅4∅- \$∅5C	Nem használt, foglalt.
\$ 18	\$∅6∅	Hibás megszakítás: ha egy megszakítás visszaigazolása során BUS ERROR lép fel.
\$ 19- \$ 1F	\$∅64- \$∅7C	Megszakítási autovektorok (1-től 7-es szintig).
\$ 2∅- \$ 2F	\$∅8∅- \$∅BC	TRAP vektorok (#∅ - #15): software megszakítások, lehetőséget nyújtanak a supervisor módba átlépéshez.
\$ 3∅- \$ 3F	\$∅C∅- \$∅FC	Nem használt, foglalt.
\$ 4∅- \$ FF	\$1∅∅- \$3FC	192 db nem autovektor (felhasználói megszakítás).

A KÖNYVRŐL ÉS A ZENRŐL

A regény cselekménye látszólag egyszerű: a főhős — az apa — hosszú motorkerékpár-túrára indul fiával — Chrisszel — és egy baráti házaspárral. Keresztülutazzák — főleg mellékutakon, elkerülve a forgalmas autópályákat — az Egyesült Államokat. Az apa, a történet egyes szám első személyű elbeszélője, ütközben mindenfelől medítál: az emberek és a technika viszonyáról, a tudományos megismerés nehézségeiről, a jó gépkönyvirásról, az analitikus gondolkodásról, a kanti filozófiáról stb. A könyvben amerikai tájak, emberek, városok, benzinkutak, éttermek, motelek leírásai váltakoznak ezekkel a meditációkkal, melyekből lassan, fokozatosan összeáll egy régebbi, „mögöttes” történet is. Rájövünk: az apa és a fia hosszú utazása tulajdonképpen menekülés az életért, lelki egészségükért — menekülés az örület goethei „Rémkirálya” elől.

Megtudjuk: az apa annak idején korán érő tehetség lehetett, aki különleges engedéllyel, már tizenöt éves korában elvégezhetette a tudományegyetem biokémiai szakának első évfolyamát, de már tizenhét éves korában el is kellett hagynia az egyetemet. A hivatás indoklás szerint „éretlenség és a tanulmányok elhanyagolása” miatt. Fokozott érzékenységre, tehetségre túl korán és roppant erővel neheztek rá a tudomány nagy kérdései. Szembesülésében kétével merültek fel benne a megismerés általánosan elfogadott módszereivel szemben. (Mi is irtnak ilyesmirel rovatunk 1987/11. számában.) Idegrendszere túlterhelhető, konfliktusba került környezetével. Végül már kötelező sokkos kezelést írtak elő számára. Ettől újra „normális” lett, de emlékezetéből kiesett múltjának legnagyobb része. „Hőstünk” az alaptörténet, az utazás idején már mint műszaki író gépkönyvek, műszaki leírások készítésével-szerkesztésével keresi kenyérét. Egyéni töretteseit tetézi, hogy az orvosok fiánál is a kezdődő elmebetegség tüneteit vélék felfedezni. A tájleírásokkal váltakozó meditációk súlyát a korábbi énjét, saját zonnosságát és gyermekével közös gyógyulásukat kereső menekülés „véres komolysága” növeli.

Pirsig műve jó könyv. Elhisszük az elbeszélőnek, valóban nagyon fontos számára az igazság megtalálása. Mielőtt a könyv technikával és tudománnyal kapcsolatos eszefuttatásaiból izelítőt adnánk — és közben még egy kis kitérőt is tennénk a dokumentálásról —, néhány szót szólnunk a könyv címében lévő zenről is. A zen japán szó — magyarul *meditáció* jelent. A zen egyúttal Japán mai legnagyobb — a 12. században alapított — buddhista szektájának elnevezése is. A könyv, melyről úgy hírlík, esetleg magyarul is megjelenik majd, *nem* valamiféle zen-buddhista mű. A könyv főhőse ugyan a történet szerint tíz évet is töltött Indiában, a beneasi egyetemen keleti filozófiát tanulta, de *nem* lett a buddhista tanok híve. Mint az empiria (a tapasztalatokon alapuló ismeretek) elsőlegességét valló tudós nem Indiába és így is tért vissza — némiképpen csalódottan és kissé ingerülten is, meg egy fontos tanulsággal gazdagabban: úgy találta, a hinduizmus, a buddhizmus és a tao közötti doktrínális különbségek kisebbek, mint a „nyugatibb” vallások közötti különbségek, mert ott, a Távol-Keleten a verbalizált, szavakban megfogalmazott állításokat soha nem tekintették azonosnak az azokban és a velük leírt realitásokkal.

A könyvben — mint említettük — sok a „felszín alatti”, ún. „mögöttes formát”, tartalmat kereső *meditáció*. Erre utal a címbeili *zen* szó.

A meditációk tehát a technika és az ember viszonyára, egyes emberek technikailenségének okaira vonatkozó egyszerű gondolatmenetektől a tudományos megismerés határait feszegető kérdésekig ívelnek. A könyv persze elsősorban regény, és nem tudományos értekezés. Mégis, aki veszi a fáradságot és elolvassa, egy sajátos nézőpontból, mellyel nem kell feltétlenül egyetérteni, napjaink műszaki-tudományos kérdéseiről *olvasmányos* formában viszonylag sokat megtudhat. A szerzővel együtt például elgondolkozhat a valóság és a valóságról kialakított tudásunk viszonyáról, arról, hogy mit értett Kant történetesen „*a priori*” tudáson. A könyv egy másik helyén a *nem euklideszi geometriáról* találunk érdekes eszefuttatásokat.

A szerző gondolatai mindvégig azt sugallják, hogy *nincs két kultúra*, a tudományt és a technikát a ma emberének nemcsak elviselni, kényszeredetten tűrni kellene, hanem meg kellene tanulnia, hogy integrálja és humanizálja azt.

A könyv nagyon *gazdag* gondolatokban. E rovat keretében sajnos nem vállalkozhatunk másra, mint izelítőtől két rövid részlet felvillantására és közbeszúrva egy „meditációra” arról, miént csen nálunk elegendő mérnök-iró.

MŰSZAKI LEÍRÁSOK ÉS A „LÁTHATATLAN” KÉS

Saját munkám közben — medítál a főhős — gyakran elgondolkozom azon a gondatlanságon is, amit a számítógépek műszaki leírásainál tapasztalok. Az év tizenegy hónapjában ugyanis műszaki leírások készítésével és szerkesztésével kereselem meg a kenyeremet és tudom, hogy ezek is mennyire tele vannak hibákkal, kértéltelműségekkel, követhetetlen ugrásokkal és annyira összekutyult információval, hogy gyakran hatszor is el kell olvasnom ezeket, amíg megtalálom bennük az értelem. De ami már az első alkalommal mellbe vágott bennük, az a javítóműhelyekben is tapasztalt, szinte megegyezően *külvilálli*, nézgelődő magatartás (alapalás), amivel ezeket a gépkönyveket írták. Mert szinte sugallják: „ez itt a gép, időben és térben *elszigetelve* a mindenség összes többi dolgától”.

Maga a szemlélő — a leendő olvasó személye — teljesen hiányzik a képből. Az ilyen leírás elefajti például megmondani: ahog, hogy mondjuk dugattyút láthassunk a motorkerékpárnál, *le kellene szerelni* a hengerfejet. „Én”, az olvasó, az ilyen típusú leírásoknál sehol sem vagyok a képbén.

Nem tudatosul az olvasóban az, hogy itt az író kezében egy *láthatatlan kés dolgozott*. Egy meglehetősen halálos kés, egy intel lektuális szike, olyan gyorsan és olyan élesen, hogy időnként a mozgását se láttad. Az az illúziód támadhatott, hogy mindazok a részek, amikről a leírás szól, egyszerűen ott vannak, ahol vannak, máshol nem is lehetnek, és *csak* úgy hívhatják őket, ahogy a gépkönyvben elnevezték őket. Csakhogy ezeket egészen másképp is el lehetett volna nevezni, sőt, az egész motorkerékpárt teljesen másképp is szét lehetett volna részekre vágdosni.

Mondjuk a motorkerékpár *leírásában* mint önálló egység szerepel egy ún. *visszacsatoló mechanizmus*, amelyik magába foglalja a büttyök tengelyt, a vezérlő láncot, a gyújtászáblayozót, a megszakítókat stb. Ez a „része” a motorkerékpárnak kizárólag azért „létezik”, mert az előbbi — a gépkönyvvezetésnél használt — analitikus késnek *éppen így tetszett* szétvágni a teljes rendszert. Ha például bemennél egy motorkerékpár-szaküzletbe és megpróbálnád egy „*visszacsatoló szelvényt*” venni, akkor ott egyszerűen nem is tudnák, hogy mi az ördögről beszélsz. Ott ugyanis *nem így vágják szét* a motorkerékpárt. Sőt, nem találsz két gyártót sem, amely leírásában *egyformán* vágna szét a motorkerékpárt. Minden szerelő jól ismeri a formát: azért nem tud megvásárolni egy-egy alkatrészt, mert *nem találja meg* azt, mert a gyártó a dokumentáció készítésekor úgy gondolta, a kérdéses alkatrész valamilyen *más szelvény* része, mint amire a szerelő gondolt.

Ezért fontos, hogy műszaki leírások olvasásakor lássuk mi is ezt a kést. Oda tegyük, ahová való, és ne engedjük becsapni magunkat. Ne higgyük el, hogy a motorkerékpárt vagy bármi más *csak így* lehetett volna leírni. Ezért fontos, hogy dokumentációt olvasva erre a kérsre is koncentráljunk. — Eddig a figyelemfelkeltőnek szánt első részlet a könyvből.

E rovat írója napi munkájából adódóan maga is rendszeresen olvas különböző számítástechnikai műszaki leírásokat, kezelési útmutatókat, termékismertetőket is, és gyakran elgondolkozik azon, miért adnak a fejlesztők, a gyártók, a forgalmazók egyébként még jó gépeket, programjaikkal mellé is gyenge dokumentációt, olyat, melyből — ha egyáltalán — csak nagy nehézségek árán, indokolatlanul sok munkával lehet kideríteni a szükséges tudnivalókat. Úgy gondolja, az egyik ok az, hogy

és olvasmányaink közül ma Robert M. Pirsig: „Zen and the Art of Motorcycle Maintenance” (magyarul: „Zen és a motorkerékpár karbantartásának művészete”) című regényét emeljük ki (Bantam Books, 8. kiadás, 1974. április, 406 oldal), elsősorban a technikára és a tudományra vonatkozó eszmefuttatásai-meditációi miatt.

HIÁNY VAN „MÉRNÖK-ÍRÓKBAN”

A jó tervező nem feltétlenül tud jól írni is, sőt, általában inkább nem tud, ugyanis másfajta képességek szükségesek az egyik tevékenységhez, mint a másikhoz. Mégis elég tipikus eset, hogy nálunk azokkal íratják meg a — karbantartói, üzemeltetői, kereskedelmi célú stb. — dokumentációkat, akik a termékeket, a számítógépeket, a programokat *konstruálják*. Az „eredmény”: az embernek sokszor az az érzése, hogy pénzéért nem tudást, biztonságot, hanem, *kisebbségi érzést* szeretnének eladni neki. „Hadd lássa a vevő, hogy a tervező milyen okos, milyen *bonhólt* szerkezete kigondolására képes.” Pedig a fejlesztőnek nem ez volt a szándéka. Ő csak nem tudott kilépni megszokott, neki való szerepéből, nem tudta *beleélni magát* az olvasó helyzetébe, ez eszébe sem jutott.

A dokumentációkészítés lehetne társadalmilag megbecsült tevékenység, foglalkozás, és egyúttal a fejlesztés valamiféle *minőségi ellenőrzése* is. Az a termék ugyanis, melyet az erre szakosodott hivatalos szakírók nem tudnak sem megérteni, sem szépen, érthetően *leírni*, az *nem lehet igazán jó sem*.

Ott, ahol a dokumentálást a fejlesztés valamiféle utólagos — pótlólagos, kényszerű — járulékanak tekintik csupán, ott nem nagyon születhetnek igazán jó, piacképes termékek sem. (A hiánygazdaságok „piacán” persze sok minden eladható még, ami a valódi piacon nem!)

Valószínű, hogy a kialakult helyzetet *csak* szavakkal segíteni nem lehet. A valódi piac — ha lesz! — ki fogja kényszeríteni ennek a szakmának, a mérnök-íróinak az elismerését és ezzel elterjedését is.

Felmerül még az a kézenfekvő kérdés is, hogy *dokumentálhatlan-e a szoftver?*

A jó dokumentálhatóság összefügg azzal is, hogy a dokumentáció tárgya jól *strukturálható-e*. A mérnöki tervezőmunka hagyományos végeredményei fizikailag megfogható, „kézbe vehető” gépek, eszközök stb. voltak. Itt a dolgok természetes struktúráját rendszerint az adta, hogy „*mi van benne miben?*” Ami magában *foglalt* valamit, az állt a hierarchia magasabb szintjén. Például egy *komplett* sebességválogató szerelvény *részeinek* tekintetében benne lévő tengelyeket, a fogaskerekeket, és nem fordítva.

A számítógépeknél — elsősorban a programoknál — ez az egyszerű „*bennfoglalási*”, strukturálási elv már nem ennyire egyszerűen követhető: igaztven lehet úgy is tervezni, hogy tulajdonképpen „*semmi sincs szoftver semiben*”. Ez a hagyományos gépeknél szinte lehetetlen volt. Tipikusan nem „haraphatnak egymásba” a gépek alkotóelemei, mert akkor nem lehetne rendszeres szétszerelési és összeszerelési éket. — A mai ún. *strukturált* programozási technikák tulajdonképpen ezen a helyzeten kívánnak segíteni, amikor azt írják elő, hogy az alkotóelemek a szoftvernél is legyenek csak egymásba ágyazva, *egymásba foglalva*, az egyes részek „*ne harapjanak egymásba*” (lásd például az ún. „*goto utasítás nélküli*” programozási technikákat). Még sok időnek kell azonban elteltie ahhoz, hogy megtaláljuk a célszerű strukturálás és a jó dokumentálhatóság közötti kompromisszumos *szoftvertechnológiát*.

A HIPOTÉZISEKRŐL

A dokumentálás mellett hitvallás után térjünk vissza a cikkünk elején bemutatott könyv egy másik, erősen vitatható, de egyben gondolatébresztő eszmefuttatására, mely a *hipotézisekről* szól. A főhősben akkor következett be a fiatalkori törés, amikor laboratóriumi kísérletei kapcsán a hipotéziseknek a kérdésével kezdett foglalkozni. Újból és újból arra a következtetésre jutott, hogy amit

korábban a tudományos munka tudomány legnehezebb részének tartott — a hipotézisek felállítását —, az bizonyult a legkönnyebbnek a gyakorlatban. Szinte maguktól következtek abból, hogy kísérleteire vonatkozóan mindent következetesen, világosan leírt. Miközben az első hipotézisét ellenőrizte, új hipotézisek egész sora ötlött eszébe. És amikor ezeket is elkezdte ellenőrizni, ismét újabb hipotézisek kezdtek körvonalazódni. Ez így ment tovább és tovább, egészen addig, ameddig „világos” nem lett számára, hogy miközben ellenőrzi és szelektálja a korábbi hipotéziseit, a megvizsgálandó hipotézisek *száma* nem akar *csökkenni*, hanem inkább egyre *nő*.

Kezetben még szórakoztatónak is találta ezt a „felfedezést”. Parkinson nyomán meg is fogalmazott egy vicces törvényt: „*egy jelenség racionális magyarázatára alkalmas hipotézisek száma végtelen*.” Még mulattatta is a gondolat: sohasem fog ki a hipotézisekből. Még ha egy időre úgy is látszott, a kísérleti munkája kapcsán zsákutcába került, nincs kiút, tudta, ha egyszer hosszabb időre le tud ülni és nyugodtan tud gondolkodni, akkor biztosan felbukkan egy újabb hipotézis. És ez menetrendszerűen mindig be is következett. Csak hónapokkal azután, hogy humoros kedvében megfogalmazta „törvényét”, merült fel benne kétség az íránt, hogy a dolog valóban olyan humoros-e.

Ha a megfogalmazása, felfedezése igaz — gondolta —, akkor itt nem csupán a tudományos érvelés egy kisebb, elhanyagolható *szépséghibájáról* van szó. Ez a „törvény” *teljesen nihilista*. Ezzel elérkezett volna minden tudományos módszer általános érvényességének katasztrófális csúfolathoz!

Ha igaz az: a tudományos módszer célja, hogy hipotézisek sokaságából kiválassza azt, amelyik igaz, és ha ezzel egyidejűleg igaz lenne az is, hogy a hipotézisek száma a kísérletek előrehaladtával *gyorsabban nő*, mint ahogy azok igazolhatók vagy cáfolhatók, akkor nyilvánvaló, hogy az összes hipotézist *soha sem lesz módunkban ellenőrizni*. Ha pedig az összes hipotézist nem lehet ellenőrizni, akkor minden kísérleti módszer következtelen és az egész tudományos megközelítésünk alkalmatlan arra, hogy kipróbált, igazolt tudományos ismereteket szerezzünk általa.

Visszaemlékezett arra, hogy minderről Einstein a következőket mondta: „A fejlődés megmutatta, hogy egy adott pillanattal az összes elképzelhető konstrukcióból *egy* mindig magasban a többi felett állónak bizonyult.” De főhősünk számára ez nagyon gyenge válasz volt. Az „*egy adott pillanattal*” kitétel valósággal megráztta őt. Vajon Einstein azt gondolta-e ezzel a megfogalmazással, hogy az igazság az *idő függvénye* lenne? Ha ezt akarta ezzel mondani, akkor ez nem jelentené-e a tudomány összes alapvető feltételezésének a semmivé tételét?!

De mit mutat a tudomány egész története? — Az nem más, mint régen ismert tények új és időben változó magyarázata. Azok az időtartamok, ameddig egy-egy törvényt, egy-egy magyarázatot elfogadottak tekintettek, véletlenszerűen oszlanak meg. A főhős ebben nem látott semmiféle rendszert. Egyes tudományos magyarázatok akár évszázadokig is változatlanul maradtak, míg másoknál ez az időszak nem volt több, mint egy év. A tudományos igazság nem dogma, nem olyasmí, ami az idők végtelenségéig elfogadott marad, hanem *időleges*, amelyet ugyanúgy lehet *tanulmányozni*, további vizsgálatok tárgyává tenni, mint bármí mást. És éppen elfogadott magyarázatok „élettartamát” a felmerülő újabb magyarázatok — hipotézisek — száma rövidíti: minél több hipotézis merül fel, annál rövidebb lesz a magyarázat — az „igazság” — élettartama. És éppen a tudományos módszer az, amelyik korunkban a felkínált hipotézisek számát folyamatosan növeli. (Minél inkább növekszik, annál többet látsz!) Ahelyett, hogy *szekletáld* a helyes magyarázatot a hipotézisek sokaságából, inkább az ellenőrizendő *hipotézisek sokaságát növeled*. Hogy megpróbálsz előrehaladni a változatlanul hitt igazság felé a tudományos módszer alkalmazásával, úgy tulajdonképpen *még csak nem is közeledhetsz* felé. Éppenséggel távolodsz tőle! És ennek oka — a végül is ebbe a „felfedezésébe” beleőrült főhős szerint — magában a tudományos módszerben keresendő! Mert szerinte az eredmény tudományosan előállított ellentudomány. És ez az eredmény nem más, mint a káosz.

Természetesen nem kell egyetérténi a könyv minden gondolatmenetével, a hipotézisekről szólóval sem. A szerző végül is nem „tanokat hirdetett”, hanem (jó) regényt írt.

A sakkprogramok értékelőfüggvényének nélkülözhetetlen alkatrésze a centrumkontroll. A program értékelőfüggvénye ennek felhasználásával tudja eldönteni, hogy mely figurák és milyen mértékben hatnak a centrumra. Ez az összetevő nagyon fontos, ugyanis a sakkban a legtöbbször az a kezdeményező fél, aki a centrumot uralja, és mint tudjuk, a kezdeményező fél mindig aktívabb ellenfelénél, emiatt előnyben van.

A centrumkontrollt a jelenlegi sakkprogramok kivétel nélkül felhasználják az állásértékelésnél. Igaz, a különböző programok más-más értékelést alkalmaznak, mindegyik más súllyal viszonyítja a függvény többi tagjához, de abban mindegyik algoritmus megegyezik, hogy használ egy olyan táblázatot, amely a tábla minden mezőjéhez egy súlyozó értéket rendel. A sakkprogramok „őskorában” figuratípustól és szintől függetlenül, azonos táblázatot használtak minden figurára. A modern programok fejlettebb, precízebb számítási módot alkalmaznak, ugyanis a különböző figuratípusok centrumértékét különböző táblázatok alapján számítják.

A fejlődés nem áll meg soha. Már ennél a módszernél is precízebbet használnak, olyat, amely a figuratípuson kívül a figura színét is megkülönbözteti. Vagyis különböző táblázatot használ például a világos és a sötét huszár centrumértékének kiszámításánál. Ezzel lehetőség nyílt arra, hogy egyszerű módon támadásra készítsük a programot. Ehhez nem kell más, csak hogy a táblázatban az ellenfél térfelelő centrummezőkhöz és az ellenfél királyát körülvevő mezőkhöz nagyobb pontértéket rendeljünk.

Vannak olyan programok is, amelyek az elemzésben az összes figurára azonos táblázatot használnak a mezők értékeinek megállapítására, de az összesítésnél a különböző figurák centrumértékét különböző kvócienssel szorozzák meg. Ezzel az eljárással a legősibb technikát ötvözik a legmodernebbel.

A most bemutatott elemzésnél az egyszerűség kedvéért a legősibb technikát alkalmaztuk azzal a módosítással, hogy kiválasztjuk azt

az esetet, amikor egy mezőn saját figura van, azaz védjük a mezőt, és azt, amikor ellenséges figura van azon, vagyis támadjuk azt. Ha a megadott mezőkön a figurák védik egymást, akkor azoknak a mezőknek a centrumértéke kétszer annyit ér esetünkben, mint ha az ellenséges figurák támadnák azokat. Ezzel a számítási móddal programunk kerülni fogja az agresszivitást, és figuráinak elhelyezkedését összehangolja, harmonikusan játszik.

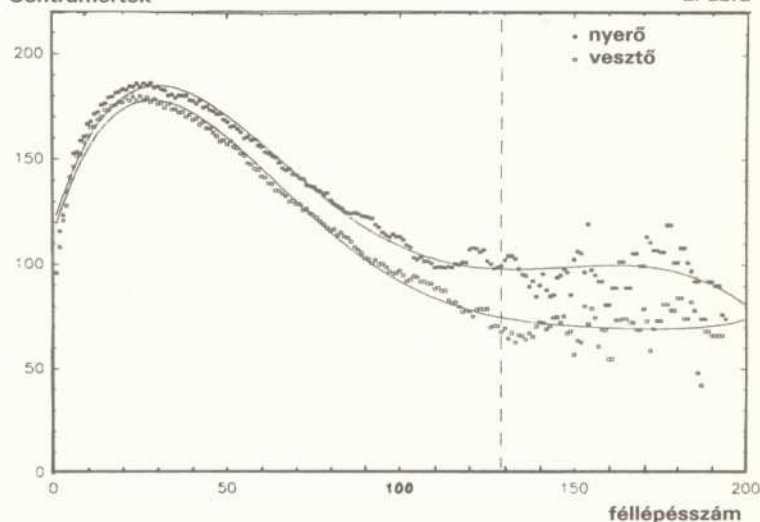
A statisztikai adatok az 1. ábrán látható táblázat alapján készültek, a következő képletet figyelembe véve:

$F_{cen} = B \times (\text{támadók száma} + 2 \times \text{védők száma})$, ahol B a mező megfelelő centrumértéke.

A 2. ábrán és a 3. ábrán 832 mesterjátszma minden lépésenkénti centrumértékének átlagát láthatjuk a nyerő és a vesztes félre vetítve.

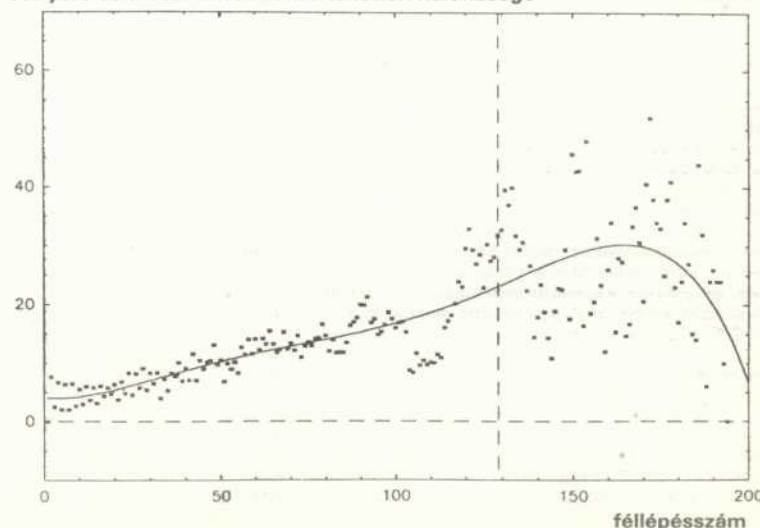
Kovács P. Attila

Centrumérték



2. ábra

A nyerő és a vesztes centrumértékének különbsége



3. ábra

A centrumérték mezők szerinti eloszlása

1. ábra

8	0	0	0	0	0	0	0	0
7	0	2	2	2	2	2	2	0
6	0	2	4	4	4	4	2	0
5	0	2	4	8	8	4	2	0
4	0	2	4	8	8	4	2	0
3	0	2	4	4	4	4	2	0
2	0	2	2	2	2	2	2	0
1	0	0	0	0	0	0	0	0
	a	b	c	d	e	f	g	h

Az Olvasó írja

Néhány változatos tartalmú levelet válogattam a rovata. Remélem, a kedves Olvasók megírják ezzel kapcsolatos véleményüket is.

Dudás Gyula, Göd

Örülök, hogy kedvenc lapom profilja gazdagodott, s ezentúl „jóirodalmi” fordításokat is közöl. Szívesen hozzájárulok a paletta színesítéséhez egy bolgár írással.

Örülök, hogy örül a „jóirodalmi” fordításoknak, és köszönöm a bolgár írást, amelynek csak egyetlen baja van, nem „elég jó”. Másoknak is tanácsolom, hogy mielőtt nagyobb munkába kezdenek, konzultáljanak a lap valamelyik szerkesztőjével, ugyanis még azt is nagyon fontos tudnunk, hogy közölhető-e az írás, azaz „copyright-tiszta”-e, kinek a szellemi tulajdona stb.

Kolozsvári Károly, Pilisvörösvár

Elnézést, hogy levelemmel zavarok, de volna egy ötletem az önök számára. Többször olvastam már, hogy az olvasók kifogásolják a lapban szereplő listák olvashatóságát. Nos, a javaslatom segít ezen. Nem tudom, ismerik-e az orosz nyelvű gyerekújságát, a „Kolobok”-ot. Ebben mindig van olyan mese, vers, ami meg is szólal. Ezt úgy oldották meg, hogy a szöveget (zenét) hajlékony műanyag lemezre veszik fel. Ez lehet kismértékben hajlítani (amíg meg nem török). Ez a kis lemez egész jó hangminőséget produkál. Gondolom, sejtik már, mire gondolok. Ilyen lemezre felvehetnének olyan programokat, amelyek a lapban megjelennek. Igaz, hogy a lemezjátszó nem számítógépperiféria, de a lemez átjátszható kazettára, vagyis már a magnóba helyezni számítógépi felhasználásra alkalmas. A lap sem lenne sokkal drágább, mivel a Kolobok is csak 10–12 forint, és négy ilyen lemezt tartalmaz. Remélem, használni tudják javaslatomat, amely már külföldön bevált.

Az ötlet majdnem jó. Azt kell ugyanis tudni, hogy a hanglemez meglehetősen drágán előállítható médium, amit csak akkor lehet olcsón adni, ha azt több százszoros példányszámban terjesztik. Nem ismerem a Kolobok c. újságot, ismerve viszont a Szovjetunió méreteit, nem tartom lehetetlennek, hogy a lap példányszáma elérje esetleg a milliót is. Ilyen példányszámban a beletett hanglemez árfekvése viszonylag kedvező lehet.

Ha ugyanezt a Mikroszámítógép Magazin 22–25 ezres példányszámára vetíti, a helyzet már egyáltalán nem ilyen kedvező.

A hanglemezről kazettára másolás, mint ötlet, nem rossz, de azt hiszem, hogy az alkalmazása nagyon kényelmetlen, és ezért nem tartom valószínűnek, hogy akadna vállalkozó, aki ezt megvalósítja. Bár ki tudja?!

Pákozdy Pál, Szolnok

A májusi Magazinban az olvasók véleményét kérték arról, hogyan vélekedünk a játék- és felhasználói programleírásokról, illetve közléséről. Ennyire egyszerűen és csatlakozom Boros Péterhez abban, hogy különböző játékokhoz és felhasználói programokhoz lehetne részletes leírást, térképet, ötleteket közölni. Jömagamnak összesen több mint ezer programom van, és ilyen mennyiségnél már nem lehet igazán átlátni a programokat, fejben tartani a kezelésüket, főleg nem az olyanokat, amikhez nem öt perc kell, hogy kiismerje az ember. Nagyon sok jó játékom átsiklottam azért, mert nem jöttem rá a nyitjára, illetve nem tudtam mit kezdeni velük. Időnként segítségemre vannak a könyvek és a Sinclair újság, de az ilyen kiadványok száma kevés, s a játékok is csak feltételesen egyeznek meg.

A leírásokért, a beküldött anyagokért, rajzokért a fizetséget nem tartom fontosnak. Azért gyűjt az ember játék- és felhasználói programokat, hogy szórakozzon, kikapcsolódjon velük, s ha már ott a lehetőség, leírja és lerajzolja amit csinál. Így saját magát is segíti az ember. Nézetem szerint a Mikroszámítógép Magazinak kellene felkarolnia a „játék-leírást”, vállalkozást. Természetesen nagyon függne attól, hogy melyik géptípusra jönne be a legtöbb leírás, mert ennek függvényében lehetne őket kiadni. Ezt pluszoldalakon a Magazinban vagy mint a Magazin különszámát lehetne eljuttatni az olvasókhöz. Hiszem, hogy 30–40 forintos áron gazdára találnának az újságok, mert ki ne szeretné a programjait megismerni, és főleg játszani velük.

Más! Adok — veszek — cserélek rovat. Mélyélesen elítélem azokat, akik programokat hirdetnek „olcsón, nagy választékban” az újság hasábjain, eladásra. Magam is írok „pénzes” programokat eladásra, a Szerzői Jogvédő Hivatallal karöltve. De sértőnek tartom, hogy a lapban nyíltan lehessen kalóz úton játék- és felhasználói programokkal üzletelni.

Levelének közzétételére egy kicsit később került sor, de tartalma még most is aktuális. Azt hiszem, hogy nincs közöttük nézeteltérés. Mi mindig és mindenhol a programlopást, az illegális programmásolást és terjesztést ítéljük és ítélnék el. Nem tartjuk etikusknak, ha valaki más szellemi produktumából csinál üzletet, annak a másikkal a tudta és engedélye nélkül. Persze, ha valakinek az üzlethez és nem a fejlesztéshez van tehetsége, akkor üzleteljen, annak is megvan a törvények által megszabott módja, sőt a díjazása is. Kössön szerződést a szerzővel és árulja a programját. Ma az eladók általában 50 százalékban részesednek az árbevételekből, de láttam már olyan üzletet is, hogy egy vállalkozó 4–5000 példány eladására vállalt garanciát, és az ezért járó közvetítési díjat (kommissiót) előre kifizette a szerzőnek. Biztosan mindenki meglepődik, ha elárulom, hogy ilyenkor, amikor az eladó és nem a szerző vállalja a kockázatot, a szerző díja 10 százalék és az eladóé 90. Mindezt azért írtam le, mert egy ügyes eladó egy keveset kockázatot szerzón tényleg meg tud gazdagodni, de legálisan! Ami az első javaslatát illeti, gondolkozunk rajta.

Bódi László, Egyek

Primo A—64 '84.1-es kiadású gépem van, így egyes IC-k, botkormány-csatlakozók nincsenek beépítve. Kérem, írja meg, mit kell — s hogyan — megépíteni, beépíteni ahhoz, hogy használni tudjam. Ha beütem az IC-eket és néhány más elemet, működik-e azonnal, vagy még illesztőt kell építeni, esetleg segédprogramot kell betölteni? A Rádiótechnika '87/5. számában megjelent egy illesztőegység leírása, illetve egy betöltőprogram, ezt a Primo „BIE” 2×25 pólusú kimenetűre kell csatlakoztatni minimális belső módosítás után. Ennek megépítése esetén szükséges-e az IC-eket beütemíteni?

Levelére Simonyi Endre válaszol: Az „A” és a „B” jelű csatlakozók bekötése a Primo hardver könyvében van leírva. Ezenkívül még illesztőegységre is szükség van a botkormány működtetéséhez. Ha ezt a változatot kívánja megépíteni, akkor az illesztőegység rajzát elküldjük.

A Rádiótechnika 1987/5. számában leírt megoldás más kiegészítést nem igényel, az ott közölt kapcsolás működőképes.

A legjobbakat kívánom minden Olvasónknak:

Kovács Győző

PROGRAMTERMÉK

SZUMMA

Immár egy éve faggatjuk az oktató-programokat. Ideje összegezni az eddigi tapasztalatainkat. — A cikksorozat úgy született, hogy kaptunk egy Assembler oktatóprogramot bevizsgálásra. A szerkesztőség véleménye az volt, hogy ne álljunk meg egyetlen program bemutatásánál, mert ezzel a Magazin feltételül jó ügyet szolgálja. Az évi 12 megjelenés persze a piacon található nem csekély számú programhoz mérten nem túl sok lehetőség volna, de ha megvalósítunk valamelyes előszelkeket, akkor reménykedhetünk abban, hogy felhívjuk a figyelmet néhány kiemelkedő termékre. A sorozat ennek szellemében indult el. A programokat úgy választottuk ki, hogy megkerestük a potenciális programforrásokat és válogattunk a termésükből.

Az elemzések módszereiként a szakfolyóiratok tesztelési szisztemáit alkalmaztuk, hozzáigazítva azokat sajátosságainkhoz. Az év folyamán egyeztetünk megartani ugyanazt a rendszert. A szisztematikusan vizsgált adatokon kívül egyeztetünk feltárni egy sereg heurisztikus megfigyelhető adatot is, amelyeket a termékekhez igazodó módon kellett megválasztani. Fő célunk az volt, hogy számot adjunk egy-egy termék olyan előnyös vagy netán hátrányos tulajdonságairól, amelyek — az osztályzatok kiegészítéseként — hasznosan informálhatják a potenciális vásárlókat.

A programok vizsgálatánál erősen esett latba a dokumentáltság és az, hogy a termék mennyire alkalmas az önképzésre. Sajnos, általános tapasztalat, hogy a programok készítőinek a termék dokumentálására mindig jóval kevesebb erejük marad, mint a programhibák elhárítására és a „bolondbiztos” kezelhetőség megoldására. A programok kezelhetőségével kapcsolatban általános probléma, hogy a programírók „elvakultságban szenvednek”, és nem érzik, hogy a vevők mit nem fognak megérteni a program használatakor. Ez egyrészt azért fordulhat elő, mert szinte rájuk erőltetik saját, megszokott számítógép-kezelési stílusukat, de ugyanilyen baj az is, hogy túltengő tantárgyszeretből kiindulva többet tételeznek fel a vevőről, mint amennyi reális lenne.

Böngésszünk az elemzett termékek táblázati adatai között.

Forgalmazók

Az eddig elemzett termékek az alábbi forgalmazóktól származtak: IHB (1), Magiszter (1), Novotrade (3), TII (6). Ebből a listából több érdekes dolog derül ki. Az oktatóprogramok két fő forrása a Novotrade és a TII. Kettőjük közül is talán az utóbbinak van nagyobb programgyártása. Ez a TII korábbi oktatási program-pályázatának eredménye. Programpályázatot a Novotrade is hirdetett, de választékában mégis a rendszeres belső fejlesztések képviselik a nagyobb részt. Ezek a termékek a piacon ma a legnagyobb fajsúlyúak.

Terméknév

Az eddig boncolgatott termékeknek felsorolása haszontalan volna. Inkább az érintett témaköröket tekintsük át. Voltak olyan termékek, amelyek kifejezetten csak a számítástechnikai háttérrel foglalkoznak (Assembler-oktatás). Mások szerzőknek adnak a pedagógus kezébe (feladat-generátorok, ábrarajzolók stb.). A legjelentősebb halmaz az volt, amelyik valamilyen konkrét ismeret oktatását segíti (nyelvoktatás, földrajz, fizika, történelem, csillagászat stb.). Az elemzett programok kapcsán a szerző elismeréssel adózik sok szerzőnek azért, hogy milyen szellemesen rugaszkodott el a bemutatandó problémától a számítógép adta nyelvezet követelményeihez igazodva.

Szerzők

A szerzők említése még kevésbé lehet cél, bár a fizika programcsomag (két Zátonyi) és az Évezredek programcsomag szerzőinek munkája e cikk szerzőjének nagyon tetszett. Ami ennél tanulságosabb, az az, hogy középiskolai diákoktól nyugdíjasokig terjed a szerzők korhatára. Egyébként néhány tanár talán már ma sem tudja elképzelni az oktatást számítógépek nélkül. Még jelentősebb, hogy megértették: a programozás csak eszköz, mint az írásvetítő vagy a videomagnó.

Géptípus

A vizsgált programok körében a legelterjedtebb géptípus a Commodore Plus/4. Utána következik a TVC. Ennek a két iskola-számítógépnek legjobbak a paramétere. Egyes programoknak vannak más gépre készült változatai is, de a jobb programoknál ez ma már elenyésző. Még nem jelentek meg IBM PC kompatibilis gépre készült programok, bár néhány középiskola már hozzájutott ilyen gépekhez. Hallottuk, hogy már ezekre is fejlesztenek oktatási programokat, de az alkalmazásuk egyelőre, a felsőoktatást kivéve, várat magára.

Hordozó

Az első iskolagépénél, a HT-nál az adathordozó eszköz szinte egyedül a kazetta volt. Csak kevés HT-hez illesztettek mágneslemezeket. A Plus/4-nél és a TVC-nél ilyen korlát már nincs. A programok alapesetben most is megkaphatók kazettán, de a forgalmazóktól a vevő általában kérheti, hogy a megvásárolt programokat lemezen szállítsák neki. Sok program táro-

lásának legolcsóbb eszköze mégis ma is a kazetta. Persze mióta a turbó betűtöket feltalálták, a sebességszökkenésből eredő hátrányoknak sincs nagy jelentőségük.

Dokumentáció

Ez az oktatóprogramok Achilles-sarka, akárcsak más programtermékeké. Mint már említettük, a szerzőknek kevés energiájuk marad jó dokumentáció készítésére. Ennél nagyobb baj, hogy a forgalmazó költségeit ez és a csomagolás folyamatosan terheli, ezért ezekkel igyekszik takarékoskodni. A dokumentáció kivitelezése olykor eléggé kritikanál aluli. Hamar szét hullnak, emiatt elveszhetnek egyes lapjaik, a másolat gyengén olvasható, a papír a tartós használatot nem sokáig bírja. Valami javulás azért érezhető volt, sosem egyik cikkünknek köszönhetően javítottak egy program dokumentációján a TII-ben.

Árak

Az árak elvileg kalkuláció alapján alakulnak ki. Mégis az az érzése az embernek, hogy hihetetlenül eltérő teljesítményeket mutató termékekért forgalmaznak közel azonos áron, ugyanakkor igen eltérő ára van közel ugyanolyan teljesítményű termékeknek. Bele kell nyugodni, hogy a számítástechnikában mindmáig nincsenek kialakult technikai normák, ezért az árakat elsősorban a kifizetett szerzői díj és a forgalmazó remélt haszna határozza meg. A termékek hasznosságának erre nincs hatása. Mégis jó lenne, ha a forgalmazók az árakat az iskolák maximális vásárlási „küszöbértékéhez” igazítanák. Az árhoz tartozó probléma, hogy a TII csak iskoláknak ad el közvetlenül programokat, magánosoknak nem. A távtanulási tendenciák fényében ez nem biztos, hogy jó politika.

Minősítő adatok

A cikksorozat olvasói bizonyára nyúltszívűséggel vádolják a szerzőt az elmúlt egy évben kiosztott sok jó jegy miatt. Itt azonban figyelembe kell venni, hogy az osztályozás mindig szubjektív tevékenység, és több szerepet tölt be, mint ahogy az iskolában is. Minthogy mi előválogatást végeztünk, a jobb jegyek eleve indokoltak lehetnek. Persze, szükségét láttuk és látjuk a buzdításnak is. Az

Összbenyomás

egy év alapján végül is az, hogy az iskolában lévő számítógépek ma már nem szükségképpen csupaszon heverő, használhatatlan eszközök, mert igenis van elég sok színvonalas program a hasznosításukhoz. Csak legyen elég pénz a programvásárlásra.

Zsadányi Pál

Egytökös mikroszámítógépek.
Szerk. P. F. Lister.
(Budapest, 1988.
Műszaki Könyvkiadó, 243 oldal.
Ára: 164,— Ft.)

Az egytökös mikroszámítógépeknek nagy jelentőségük van a műszaki alkalmazásokban. Az ilyen áramkörök értékesítése jelentősen túlszárnyalja az általános célú mikroprocesszorok eladását. Az egytökös mikroszámítógépek gyártási üteme minden más integrált áramkörnél gyorsabb.

A félvezetőgyártók széles választékban állítanak elő egytökös mikroszámítógépeket, az egyszerű, olcsó 4 bites mikrokontrollerektől a nagy teljesítőképességű típusokig, amelyek jellemzői versenyeznek a legjobb általános célú mikroprocesszor-típusok tulajdonságaival. A mikroelektronikai gyártmányokban megjelenő áramkörök legtöbbször mikroszámítógépek és ezekhez tartozó kiegészítő áramkörök. Ezeket alkalmazzák az elektronikus játékokban, automata mosógépekben, autókban, televíziókban és a számítógépek periferiáiban.

A könyv alapvető célja a különféle típusok bemutatása, tulajdonságai ismertetése és alkalmazási példákon keresztül a felhasználási lehetőségeik illusztrálása. Az alkalmazási rész szerepe jelentős, mert míg az áramköri leírás csak az adott típusra vonatkozik, addig az alapvető megvalósítási elvek, mint például az interfész megoldások, általánosan alkalmazhatók. Az egytökös mikroszámítógépek sok potenciális felhasználója viszonylag kevés elektronikai ismerettel rendelkezik. Várhatóan a könyvben leírtak az ilyen felhasználók érdeklődését is fel fogják keltetni.

Bentley, Jon Louis:
A programozás gyöngyszemei.
(Budapest, 1988.
Műszaki Könyvkiadó, 175 oldal.
Ára: 95,— Ft.)

Ebben a kötetben a szakma látványosabb oldalaival foglalkozó tanulmányok szerepelnek, amelyek a biztonságos tervezőmunkánál túl a hozzáértés és kreativitás területéről származó programozási gyöngyszemeket veszik nagytól alá. Az igazgyöngyöt a kagylóból növesztik a testüket irritáló homokszemek köré, ezek a programozási gyöngyszemek ugyanígy a programozókat irritáló valóságos problémákból nőttek ki. Érdekes programok jönnek létre így, amelyekből fontos programozási technikát és alapvető tervezési elveket tanulhat az olvasó.

A könyv egyes fejezetei önmagukban is olvashatók, bár a könyvben logikailag csoportosítva vannak. A kötet első részében a programozás alapelemeit, a feladatmeghatározást, az algoritmusokat, az adatszerkezeteket és a programellenőrzést tanulmányozhatja az olvasó. A következő rész a programhatékonyságot tárgyalja, az utolsó fejezetek a korábban leírt technikák felhasználását mutatják be a különböző problémák megoldásában.

Ez a könyv programozóknak készült. Ötleteivel, megoldásaival és az ajánlott irodalommal hasznára válhat a hivatásos és az amatőr programozóknak egyaránt.

Kiss László:
1001/3 játék C—64 és C—128.
(Budapest, 1988.
LSI ATSZ, 161 oldal.
Ára: 155,— Ft.)

A kötet felépítésében követi a korábbi 1001 játéksorozatot. Ebben a kötetben az olvasó újabb érdekes játékokkal ismerkedhet meg, egyben bepillantást nyerhet az ART STU-DIO VI. 1.-rajzolóprogram kezelésébe, és hasznos információkat találhat a különböző TOOLKIT-ek és monitorok felhasználására is.

Üry László:
SYMPHONY 1.
(Budapest, 1988.
LSI ATSZ, 132 oldal.
Ára: 160,— Ft.)

A kiadvány egy hat füzetből álló sorozat első kötete. A füzetek várhatóan kéthavonta fogják megjelenni. Céljuk a SYMPHONY programrendszer használatának ismertetése.

Az első rész a SYMPHONY rendszer installálásával, az elektronikus lappal, valamint a SYMPHONY makróival és programozásával foglalkozik.

„Irodaváltás”

Az Ad-oktat program kifejlesztésének az a célja, hogy segítséget nyújtson azoknak a PC-felhasználóknak, akiknek nincsenek számítógépes ismeretek, akik kezdetben idegenkedéssel fogadják az új technika bevezetését. A programmal megszerezhető azok az alapok, amelyek a számítástechnikát emberközelivé teszik, és további ismereteket is nyújtanak.

Az Ad-oktat egy háromrészes oktatási sorozat első része, amely bevezet az új iroda műszaki eszköztárának a rejtelmeibe az információközlő berendezésektől a személyi számítógépekig. Az Ad-oktat 12 fejezetben tárgyalja az anyagot. Ennek el-sajjaitása nagyfokú önállóságot ad a tanulónak, és szabadságot az ismétlésekhez vagy a már megértett tematika kihagyására. A program különlegessége az első fejezet, amelyben a hétköznapi irodai berendezések alaposabb tárgyalásával vezet be az olyan alapfogalmakat, mint az információ, a bit, az algoritmus. A későbbiekben a program áttekintést ad az üzemeltetés nélkülözhetetlen fogalmairól, mint a lemezformattálás, bemutatja a DOS-t, a hálózati rendszereket, majd kitér az irodai alkalmazások leggyakoribbjára, a szövegszerkesztésre és az adatfeldolgozásra. Az Ad-oktat utolsó fejezete szá-

mitástechnikai miniszortár, amellyel a tanult fogalmak könnyen átismételhetők. Az Adatrend Kiszívóvetkező által kifejlesztett Ad-oktat elsősorban azoknak ajánlható, akik a hagyományos írodát az „új” iroda eszközparkjával kívánják felváltani, és ehhez meg akarják nyerni a számítástechnikai ismeretek híján levő munkatársaikat is.

ABV-hibajavítás

A cirill ábécé első három betűjéről neveztek el az Interprogramma bolgár—szovjet közös intézet új szoftvertermékét, amely a bolgár nyelvű szövegek helyesírás ellenőrzését és — kérésre — automatikus hibajavítását végzi. Amennyiben nem automatikus javítást kérünk tőle, akkor a hibás szót invertált formában írja ki a képernyőre. A program minimum 512 kb-át operatív tárú IBM PC-vel kompatibilis gépeken fut, s a szintén az Interprogramma által kidolgozott Dokument—16 nevű szövegszerkesztővel feldolgozott állományokat vizsgálja át. A helyesírást egy kétszintű szótárral ellenőrzi. A leggyakoribb 200—300 ezer bolgár szó az operatív tárban van, a winchester-táron pedig további 800 ezer szó található.

A szótár a felhasználó is bővítheti, például az általa használt szaknyelvi terminológiával. Az ABV rendszer — melynek orosz nyelvű változatán már dolgoznak — az első helyesírási ellenőrzést végző program, amit szocialista országban kifejlesztettek.

Grafológia

Számítógépes szakemberek és grafológusok együttműködésének eredményeként immár négy éve jelennek meg grafológiai célú szoftverek, melyek a napijainkra elérhetően alacsonyra mérséklődött: például a Prologic 13 500 francia frank, az Abax 20 ezer. Ezzel összevetve egy hagyományos grafológia 250—1500 frankba kerül. A szoftvereket a nagyvállalatok személyzeti osztályai is vásárolják, ezzel egészítik ki a személyzetisek munkáját. Alkalmazásukat azonban egyesek fenntartással fogadják. A grafológia nem csak az egymástól elkülönült jellemzőket veszi figyelembe, hanem az elemek egymás közötti összefüggéseit is. Egy írás elemzéséhez komoly grafológiai ismeretekre van szükség. Aki nem rendelkezik ezzel, hibázhat. Mindenki egyetért azonban azzal, hogy a grafológiai szoftverek is egyre tökéletesebbé válnak. A felhasználók képzésével pedig az alkalmazások megbízhatósága javítható. De azt is látni kell, hogy a grafológia — akár számítógépes, akár nem — csak a módszerek egyike a jelölt megítélésében.

A csökkentett utasításkészletű számítógépek (RISC=Reduced Instruction Set Computer) használatakor a gépi utasítások 20 százalékát a felhasználók aktívalják az idő 80 százalékában. Így az utasításkészlet folyamatok csökkenthetők a szoftvermunka növelésével. Hagyományos mikroprocesszorok tartalmaznak az összetett utasításkészletű számítógépek (CISC=Complex Instruction Set Computer). A RISC fő előnye a sebessége: 2-5-ször gyorsabb a hagyományos változatoknál. A RISC-áramköröket ma már sok vállalat gyártja: az Intel, a Motorola, a

Sun Microsystems, az Integrah stb. A becslések szerint ezeknek az eszközöknek a száma 1992-re várhatóan eléri a 32 bites mikroprocesszorok piaci forgalmának 20 százalékát. Más becslések szerint ekkorra az arányuk 35 százalékra nő, mégpedig olyan összetételben, hogy a munkahelyeken a gépek 20 százalékában, az irodai és személyi számítógépek 5 százalékában és az egyéb eszközök 10 százalékában RISC-mikroprocesszor fog működni.

A RISC-technológia egyik érdekes mellékterméke a CISC-processzorok hibride-

sítése. A Motorola és az Intel új termékeket nagyobb RISC-hasonlósággal hirdetik. Várhatóan a jövőben a két termék összeillesztéséből olyan integrált áramkörök születtek, amelyek szerencsésen egyesítik a kétfajta architektúra előnyeit. Jelenleg azonban a hibridesítést sokan csak reklámfogásnak tartják. Mindig szükség van a kompatibilitásra a DOS alapú rendszerekkel, ezért a CISC-processzorok nem tünhetnek el a RISC megjelenése miatt, de használhatják azokat a módszereket, amelyeket a RISC-gépektől „tanultak”

ADOK — VESZÉK — CSERÉLEK

Ebben a rovatban rövid, szűveges, a mikroszámítógépekkel kapcsolatos híreket közlünk. A díjazás: közléseknek gépielt soroként (60 karakter) 100,- Ft., magánzemélyeknek az első sor 50,- Ft., minden további sor 20,- Ft. Az NJSZT tagjainak az első három sor ingyenes. Hirdetéseiket a szerkesztőség címre várjuk.

ADOK

Amiga 500 számítógép előadó. 64'er, Happy Computer, CHIP, 68000'er, Amiga, Macintosh, PC Plus és egyéb NSZK magazinnal felírók vásárolhatók, vagy más lapokra cserélhetők. Amiga felhasználókkal kapcsolatban keressen. Raiz Tamás, Budapest, Győri u. B. 1123. Tel.: 566-941

Atari 800XL + XF 551 lemezegység kb. 300 programmal, szakirodallal, joystickokkal előadó. Szinclair II csere is érdekel. Érdeklődni: 06-42-12539 egész nap.

Atari 651E (amerikai gyártmány) típusú számítógép árul aludni. Érdeklődni: Nagy Jánosné, Vácpalota, Pf.: 7B vagy a 71-950-es telefonon.

Atari program idegen nyelvek tanulásának megkönnyítéséhez. Válaszboríték ellenében ingyenes tájékoztató! Cím: Dr. Gerő László, Szeged, Budapesti krt. 4/b. VIII/31. 6723

C16-os gépet (Plus/4-es bővíttével), 2 joystickkal, magnóval, kb. száz programmal, szakirodallal eladnám. Ár 18000,- Ft. Érdeklődni leveleiben lehet. Cím: Péntek Levente, Budapest, Havanna u. 33. I/1. 1181

C16, Plus/4 programokat előadó (8 Ft/db) magnókazettára. Cím: Pauló Tamás, Békéscsaba V., Máriássy u. 51. 5600
Plus/4-re mindentudó személyi, jövendölésmódszámító program '88 ószi változásokkal előadó. Ár: 400 Ft + kazetta, utóvétvelet is. Bogdás Lajos, Derecske, Dózsa Gy. u. 15. 4130

C Plus/4, C16 tulajdonosoknak új 151-es floppy előadó! Irányár: 15000 Ft. Somodi Géza, Győr, Rákóczi F. u. 10. 9024

C64 programokat előadó, csere is érdekel. Listát kérek és küldök. Rózsa István, Budapest, Kápolna u. 25. I/17. 1105. Tel.: 159-507 este.

C64 (rég) + SHOOT'EM UP CONSTRUCTION KIT kazettán (500) előadó. Keresem Amiga 500 tulajdonosok ismerettségét. Kótai Balázs, Sopron, Margitányi u. 37/A. 9400 "D7MEGA".

C64 + 1541 + 1531 + joystick + Lightpen + 50 lemez + 1500 program + Irodalom olcsón előadó! Sürgető! Michailov Mihály, Csorna, Vörös-H. u. 7. 9300. Tel.: 335

C64 program - 600 db kazettán és lemezen, átlag 17 Ft-ért - előadó. Kérésre listát küldök. Cím: Sasvári Gábor, Lenti, Petőfi u. 33. 8960

Commodore 64 magnóval, több mint 400 programmal, joystickokkal, jelentős mennyiségű dokumentációval - cartridge (gyorsító), monitor, reset, másoló) sűrűsűen előadó. Az ár: 20000 Ft. Cím: Péntek Balázs, Balassagyarmat, Rákóczi u. 79. 2660

C64 + 1541C floppy (1 éves) + datsette + cartridge (turbo, copy, stb.) + 1 db joystick + kb. 60 lemez programokkal + könyvek (megkímélt) előadó. Irányárát és árajánlatokat kérek. Salgótarján (06-32) 10-146, 16 órátkor.

Enterprise programokat adok-veszek-cserélek (kazettán). Racsó Zsolt, Nyíregyháza, Háman Kató u. 1. 4400
Enterprise programok előadó! (10-12 30 Ft-ig.) Válaszborítékért listát küldök. Zemen László, Budapest, Kada u. 141. fsz. 9. 1104

Enterprise programok olcsón előadó! Válaszborítékért elküldöm listám. Sala István, Budapest, Hársfa utca, 17. 1203
JOYSTICK SZERVIZ a Florián Áruházhoz közel! Javítás, magnófej-összes. Játékprogramok kazettán és floppy. Budapest, III. Kérlek u. 36. IV/24. Szerdán és pénteken 17-19-ig.

Janoszty televízió videomonitorral alakított (kép+hang) az eredeti funkciók megtartása mellett. Minden videomemóriával rendelkező számítógéphez alkalmazható. C12B esetén 80 oszlopos megjelenítésre is alkalmas. Páll Miklós, Budapest, Rákóczi F. u. 345. K 7-10 C lépcsőház IV. em. 9. 1214

Magyar nyelvű dokumentáció előadó az alábbi programokhoz: AMICA PAINT, EGA V3.2 stb. Kérésre listát küldök. Honti József, Cskváry, Május 1. u. II. 8083

MEM 5/1/4 (Videonon TVC) és Olivetti AI 090 B" lemezmaghajtó előadó. Nagy Jenő, Győr, Tákó u. 2/A. 9025

Plastic programcsomag Enterprise-ra kapható (adatbevitel, szótár, táblázatkezelés, diagramkészítés, ékezetes és cirill betűk), Varga Zoltán, Székesfehérvár, Lovdó u. 9/C. 8000

Primo A32 tápegység, demo-kazettával, könyvekkel, MK 29-es magnóval 4000 Ft-ért előadó. Machlik János, Debrecen, Sasvári Pál u. 10. IV/12. 4031

Primo SSD-hez 32 K-s komplett felhasználói programcsomag égetése. Vidékre utóvétvelet. Előzetes megbeszélés: Horváth László 583-544 délelőtt.

Spectrum (48/128 k) és C64 programok előadó. Lemezen és kazettán is megrendelhető. Régi és a legújabb szoftverek egyaránt beszerezhető. Kias Gábor, Győr, Venyige u. 31. 9028

Spectrum programok (48 k) előadó. Magyarországon a legolcsóbban, 12 Ft/db. Válaszborítékért katalógust küldök. Megfelelő partner esetén cserével is foglalkozom. Ertvoss Levente, Debrecen, Cserépes u. 22. 4026

ZX-Spectrum (48 k) gépet több száz programmal, kézikönyvekkel, esetleg magnóval előadó. Irányár: 16000 Ft. 1115 Tamás, Nyíregyháza, Fészek u. 200. 4400
TVC programokat adok-veszek-cserélek (550 org.). Listát kérek. Molnár János, Szolnok, Jászai F. u. 10. VI/25. 5000. Tel.: 56/31-085

Tudja Ön, hogy milyen egyszerű a FASTLOAD CARTRIDGE-el dolgozni? Vállalom bővített karakterkészletet, Reset gomb, IRQ (pause) kapcsolat beépítését. Seikosha SP120VDC - 180Vc nyomtatók magyarisítását, FASTLOAD, SPEEDTAPE programsegítő modulok készítését. Szívesen küldök tájékoztatót! Bártfai Barnabás, Budapest, Mörzicz S. u. 22. 1193. Tel.: 668-411/64 (délelőtt).

VESZÉK

C16-hoz fényceruzát keresek. Ajánlatot a következőkre: Szabó Péter, Mészalka, Alkotmány u. 26. 4700
C64-ra, Pascal fordítóprogramot keresek megvételre, új kazettán. Machlik János, Debrecen, Sasvári P. u. 10. IV/12. 4031

C64-re stratégiai programokat keresek. Mészáros Tamás, Miskolc, Kondor Béla u. 18. 3535

C64-es programokat veszek és cserélek lemezen és kazettán egyaránt. A válaszokat az alábbi címre kérem: Papp Roland, Kapuvár, Gumázy János Fő-tér 25. 9330

CSERÉLEK

Atari 800XL játék- és felhasználói programokat cserélek lemezen. Listát adok és kérek. Uhrin János, Békéscsaba, Madóc u. 40. 5600

C16, Plus/4 színvonalas játékokat cserélek kazettán és lemezen. Listát kérek. Cím: Bóta Ákos, Felsőtárkány, Fő u. 266. 3324

C16, Plus/4 programokat cserélek kazettán. Keresem: A-07, Winter Events névű programokat. Emmer Zsolt, Budáors, Lévali u. 21. III/12. 2040

C16, Plus/4 és C116 játékok és felhasználói programokat cserélek kazettán. Listát kérek. Griecs György, Gábor-telep, Vasut u. 25. 5664

C64-en szuper programokat cserélek kazettán. Baráth Lajos, Zalaegerszeg, Puskás T. u. 1-3/B. 8900

Enterprise programcsere! Több száz program! Válaszboríték ellenében lista! Gyors csere, olcsó eladás. Sándor József, Bonyhád, Bezerédi u. 41. III/5. 7150

Enterprise programokat cserélek. Listát kérek! Szekeres Péter, Szolnok, Juhász Gy. u. 44. 5000

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

Az elmúlt évi pályázatunkat, a PONTVADÁSZATOT befejeztük. Közöljük az októberi, a novemberi és a decemberi számban megjelent feladatok kiértékelését, az utóbbiak megoldását is. Egyben most tesszük közzé a legjobbba névsorát is, akik a megérdemelt dicsőségen túlmenően könyvtalványt is kapnak. Gratulálunk a nyerteseknek a nem kis teljesítmény eléréséhez!

Dr. Hoffmann Tibor

AZ 1988/12. SZÁM FELADATAINAK MEGOLDÁSA

1. FELADAT

a) Jelöljük a t időpontban a tengerben jelenlévő A típusú halak számát $a(t)$ -vel és a B típusúakét $b(t)$ -vel. Tekintsünk el attól, hogy a szaporodás „kvantálva” van, és így tekintsük ezeket a függvényeket folytonosoknak. Ezek változása így a differenciálhányadosokkal írható le.

A folyamatot most két szakaszra kell bontani.

I. szakasz. Ha $a(t)/b(t)$ kisebb, mint $4/5$, akkor minden A hal életben maradhat a B típusú halak rovására.

II. szakasz. Ha a fenti hányados nagyobb, mint $4/5$, akkor az A halak egy része elpusztul, mégpedig hetente $a(t) - 4b(t)/5$.

Mint hogy a kiinduláskor ($t=0$ időben) a feladat szerint $a(0)/b(0) = 1$, ez nagyobb $4/5$ -nél, és így a II. szakasszal kezdődik a folyamat. Ebben az esetben a B halak változása az időegység (hét) alatt

$$\frac{db(t)}{dt} = 2b(t) - b(t) = b(t)$$

ahol a heti 2-vel való szaporodáson kívül figyelembe vettük, hogy az A halak a héten jelenlévő B halat mind felfalják, s így a szaporodás ennyivel csökken.

Az A halak változása ugyanezzel

$$\frac{da(t)}{dt} = \frac{1}{2}a(t-1) - \left[a(t) - \frac{4}{5}b(t) \right].$$

Itt figyelembe vettük az A hal szaporodásához szükséges 1 hét érési időt. A folytonosnak tekintett $a(t)$ függvény-nyel most

$$a(t-1) = a(t) - 1 \cdot \frac{da(t)}{dt}$$

közelítéssel ez az összefüggés

$$\frac{3}{2} \frac{da(t)}{dt} = \frac{4}{5}b(t) - \frac{1}{2}a(t)$$

alakban írható és ebből $b(t)$ kifejezhető

$$b(t) = \frac{15}{8} \frac{da(t)}{dt} + \frac{5}{8}a(t).$$

Ezt a $b(t)$ előbb kapott differenciálegyenletébe behelyettesítve azt kapjuk, hogy

$$\frac{15}{8} \frac{d^2a(t)}{dt^2} + \frac{5}{8} \frac{da(t)}{dt} = \frac{15}{8} \frac{da(t)}{dt} + \frac{5}{8}a(t),$$

vagyis

$$\frac{d^2a(t)}{dt^2} - \frac{2}{3} \frac{da(t)}{dt} - \frac{1}{3}a(t) = 0.$$

Ismeretes, hogy ennek a *másodrendű differenciálegyenletnek* a megoldása

$$a(t) = C_1 e^{\lambda_1 t} + C_2 e^{\lambda_2 t}$$

alakú, ahol C_1 , C_2 , λ_1 és λ_2 állandók. λ_1 és λ_2 a

$$\lambda^2 - \frac{2}{3}\lambda - \frac{1}{3} = 0$$

egyenlet két megoldása. Ezek:

$$\lambda_1 = 1 \text{ és } \lambda_2 = -\frac{1}{3}.$$

Vezessük be a kezdeti A halmennyiségre az

$$a(0) = a_0$$

jelölést, amely a feladat feltételei szerint egyúttal

$$b(0) = a_0$$

-t is jelenti. Ekkor a C_1 és C_2 állandókra azt kapjuk, hogy

$$C_1 + C_2 = a_0 \text{ és } \frac{5}{2}C_1 = a_0,$$

melyek segítségével

$$a(t) = \frac{a_0}{5} (2e^t + 3e^{-\frac{t}{3}})$$

és

$$b(t) = a_0 e^t.$$

Mindaddig e függvények szerint nő (mert nő) a halak száma, amíg a II. szakasz feltétele teljesül. A halak aránya ebben az esetben az

$$\frac{a(t)}{b(t)} = \frac{2}{5} + \frac{3}{5} e^{-\frac{4t}{3}}$$

függvény szerint változik. A II. szakasz feltétele addig a t_1 időpontig teljesül, amikor

$$\frac{2}{5} + \frac{3}{5} e^{-\frac{4t_1}{3}} = \frac{4}{5},$$

vagyis amíg

$$t_1 = \frac{3}{4} \ln \frac{3}{2}.$$

ami közelítőleg

$$t_1 = 0,304099 \text{ hét}$$

vagy

$$t_1 = 2,13 \text{ nap.}$$

Ekkor a_1 -gyel jelölve az $a(t_1)$ értéket

$$a_1 = \frac{6}{5} \sqrt[3]{\frac{2}{3}} a_0,$$

vagyis közelítőleg

$$a_1 = 1,084322 a_0$$

Ettől kezdve a folyamat átlép az I. szakaszba. Ebben az esetben a B halak változása időegység alatt

$$\frac{db(t)}{dt} = 2b(t) - \frac{5}{4}a(t),$$

minthogy ebben az esetben az aránylag kisebb számú A hal nem pusztítja el az összes B halat, hanem csak annyit, amennyit felfalni képes. Ebből

$$a(t) = \frac{8}{5}b(t) - \frac{4}{5} \frac{db(t)}{dt}.$$

Az A halak változása pedig

$$\frac{da(t)}{dt} = \frac{1}{2}a(t-1),$$

minthogy most a B halak száma nem okoz kihalást az A halak között. Az előbbi megmondolás szerint figyelembe véve az 1 heti érési időt

$$\frac{da(t)}{dt} = \frac{1}{2}a(t) - \frac{1}{2} \frac{da(t)}{dt},$$

vagyis

Pontvadászat

$$\frac{3}{2} \frac{da(t)}{dt} = \frac{1}{2} a^2(t).$$

$a(t)$ fent kiszámított kifejezését behelyettesítve most azt kapjuk, hogy

$$\frac{12}{5} \frac{db(t)}{dt} - \frac{6}{5} \frac{d^2b(t)}{dt^2} = \frac{4}{5} b(t) - \frac{2}{5} \frac{db(t)}{dt},$$

vagyis

$$\frac{d^2b(t)}{dt^2} - \frac{7}{3} \frac{db(t)}{dt} + \frac{2}{3} b(t) = 0.$$

Ennek megoldása

$$b(t) = D_1 e^{\mu_1(t-t_1)} + D_2 e^{\mu_2(t-t_1)}$$

alakú, ahol D_1 , D_2 , μ_1 és μ_2 állandók. Itt μ_1 és μ_2 a

$$\mu^2 - \frac{7}{3}\mu + \frac{2}{3} = 0$$

egyenlet két gyöke. Ezek

$$\mu_1 = 2 \text{ és } \mu_2 = \frac{1}{3}.$$

A kezdeti érték ($t = t_1$ -nél)

$$D_1 + D_2 = \frac{5}{4} a_1,$$

és

$$\frac{4}{3} D_2 = a_1,$$

amivel ebben a szakaszban

$$a(t) = a_1 e^{\frac{1}{3}(t-t_1)}$$

és

$$b(t) = \frac{a_1}{2} \left[e^{\frac{2}{3}(t-t_1)} + \frac{3}{2} e^{\frac{1}{3}(t-t_1)} \right].$$

Most tehát

$$\frac{a(t)}{b(t)} = \frac{4}{3 \times 2 e^{\frac{5}{3}(t-t_1)}},$$

ami azt mutatja, hogy hosszú idő múltán az A halak a B halakhoz képest kihalnak. (8 pont)

b) Ha a B típusú halak szaporodása hetenként csak 1 új hal, akkor a két szakaszra való bontás ugyanaz, mint eddig, de a kezdeti II. szakaszban a B halak változása

$$\frac{db(t)}{dt} = b(t) - b(t) = 0,$$

vagyis a létrejövő halakat az A halak mind felfalják. Így a $b(t)$ állandó marad. Az A halak változása ekkor a

A Pontvadászat befejeződött. Az utolsó öt szám megfejtéseinek összesítő kiértékelése a következő.

A feladat megnevezése	Maximális pontszám	Megfejtések száma	Megoldási százalék
1988/ 8. 1.	3	6	88,9
1988/ 8. 2.	18	3	77,8
1988/ 9. 1	4	5	85,0
1988/ 9. 2.	8	2	31,3
1988/10. 1.	5	2	60,0
1988/10. 2.	8	3	100,0
1988/11. 1.	6	2	75,0
1988/11. 2.	10	2	80,0
1988/12. 1.	12	2	83,3
1988/12. 2.	5	2	90,0

$$\frac{3}{2} \frac{da(t)}{dt} = \frac{4}{5} a_0 - \frac{1}{2} a^2(t)$$

összefüggés szerint történik, vagyis

$$\frac{da(t)}{dt} + \frac{1}{3} a(t) = \frac{8}{15} a_0$$

elsőrendű lineáris differenciálegyenlet szerint. Ennek megoldása most

$$a(t) = \frac{a_0}{5} (8 - 3e^{-\frac{1}{3}t})$$

és így

$$\frac{a(t)}{b(t)} = \frac{1}{5} (8 - 3e^{-\frac{1}{3}t}).$$

Mint látható, ez az arány a kezdeti 1-től kezdve folytonosan növekszik 8/5-ig, tehát a II. szakasz feltétele mindig teljesül, és így az állandó B halak mellett az A halak száma aszimptotikusan 60%-kal növekszik meg végtelen idő múlva, s ugyanakkor a B halak száma 62,5%-a lesz az A halak számának. (4 pont)

Érdeemes megjegyezni, hogy olyan folyamatok is létrejöhetnek bizonyos adati paramétereknél, amelyeknél periodikusan változnak a halak száma, és ez a halászat szervezéséhez ad felvilágosításokat. Az eljárás az ilyen biológiai folyamatok leírásán kívül a gazdasági élet folyamataira, sőt a társadalmi változások folyamataira is alkalmazható a leírásban lévő paraméterek értelemszerű változtatásával.

2. FELADAT

Ismeretes, hogy

$$\ln(1+y) = y - \frac{y^2}{2} + \frac{y^3}{3} - \dots$$

ha

$$|y| < 1.$$

Ez egyúttal

$$\ln(1-y) = -y - \frac{y^2}{2} - \frac{y^3}{3} - \dots - t$$

és így

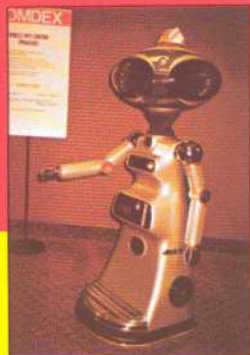
$$\frac{1}{2} \ln \frac{1+y}{1-y} = y + \frac{y^3}{3} + \frac{y^5}{5} + \dots - t$$

is jelenti. Az adott függvény így 10^{-8} pontossággal $\ln x$ -et adja meg. (5 pont)

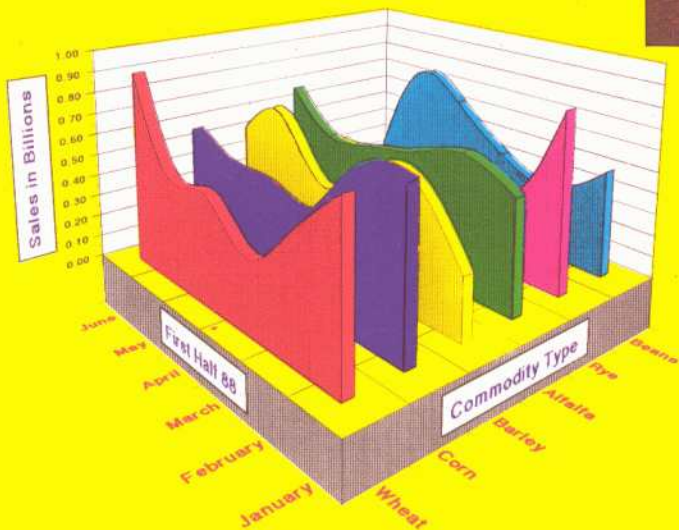
A teljes pontvadászat kiértékelését elvégezve, a lehetséges 144-ből a legtöbb pont megszerzői az alábbiak:

Nagy D. István, Csikszereda, Románia	105,5 pont
Sterczer Ödön, Tatabánya	102 pont
Gáspár László, Budapest	83 pont
Nagy-Imecs Vilmos, Székelyudvarhely, Románia	68 pont
Zilinyi Vilmos, Debrecen	64 pont
Hirschler András, Budapest	61 pont
Jakab Sándorné, Zirc	55,5 pont
Pozsgay László, Budapest	50 pont
Vámos János, Mezőkövesd	44 pont

Ezeket a legjobb megfejtőket 200 forint értékű könyvtárral jutalmaztuk, amelyet postán küldünk el nekik.



Commodities Sales



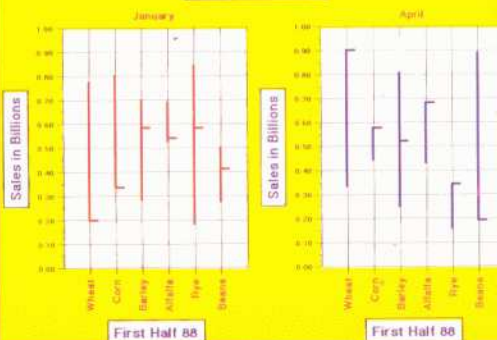
■ Wheat
 ■ Corn
 ■ Barley
 ■ Alfalfa
 ■ Rye
 ■ Beans

Commodities Sales



■ Wheat
 ■ Corn
 ■ Barley
 ■ Alfalfa
 ■ Rye
 ■ Beans

Commodities Sales





Amerikából jöttem...

