

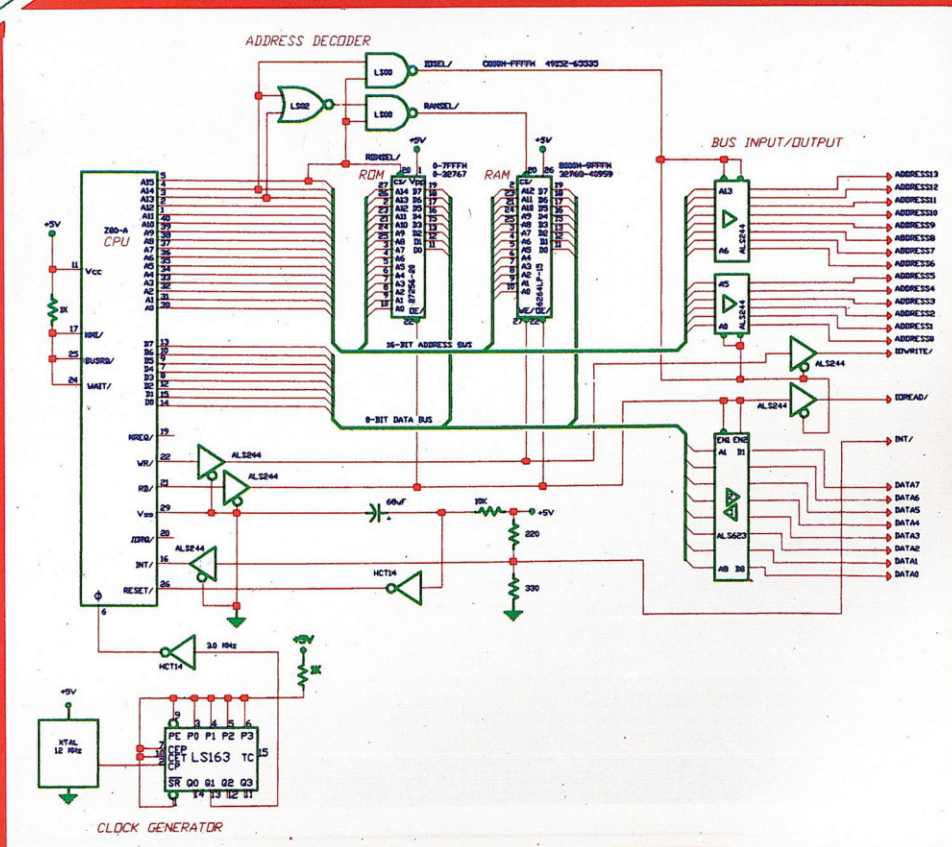
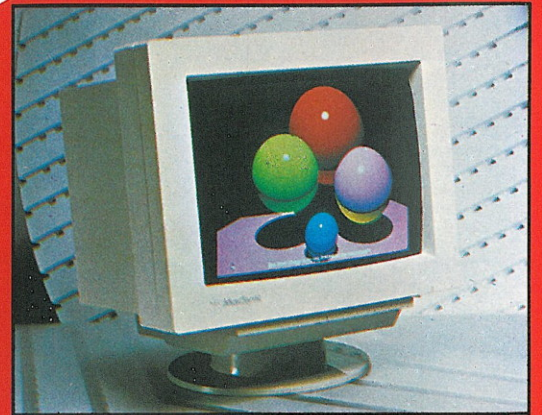
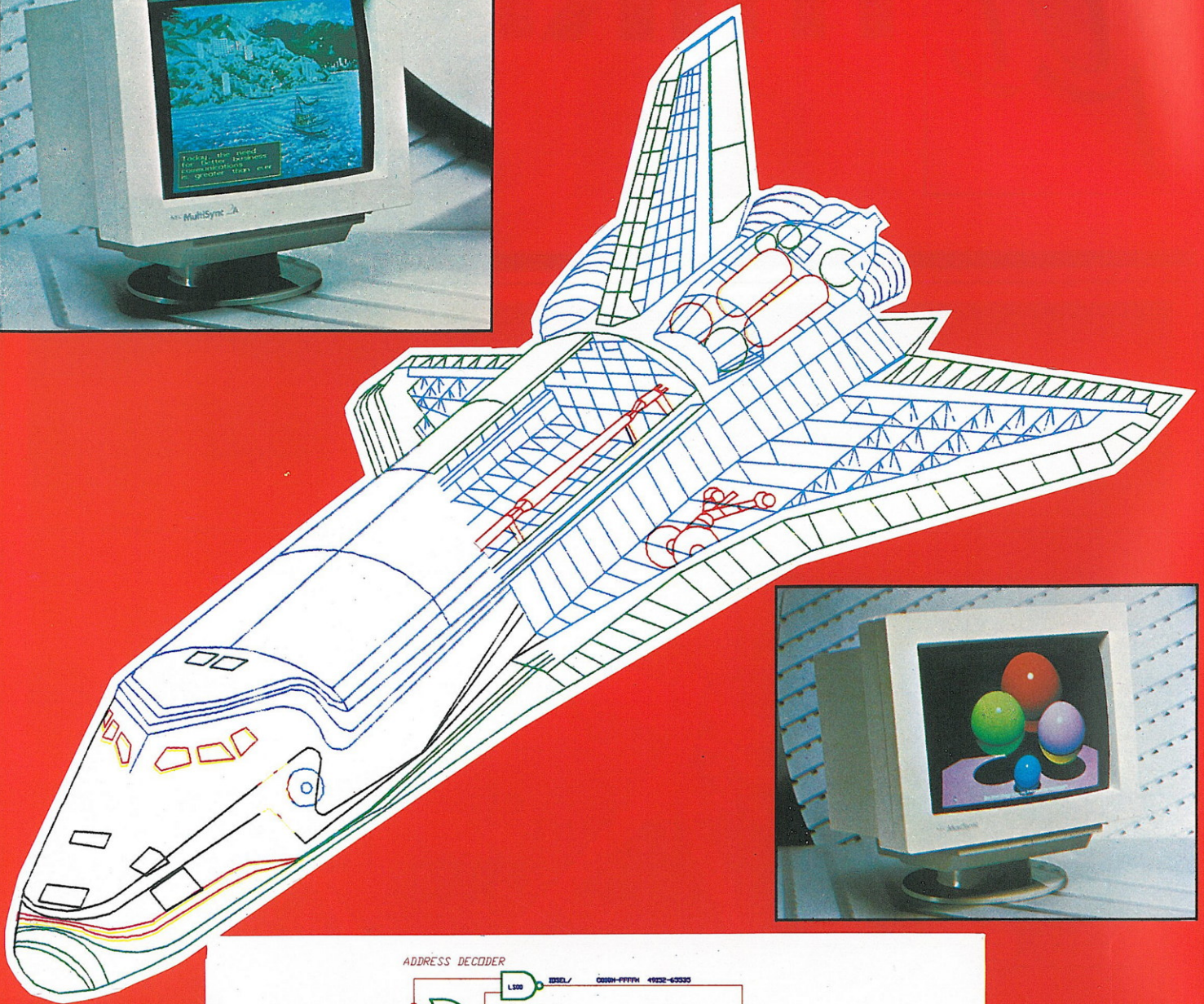
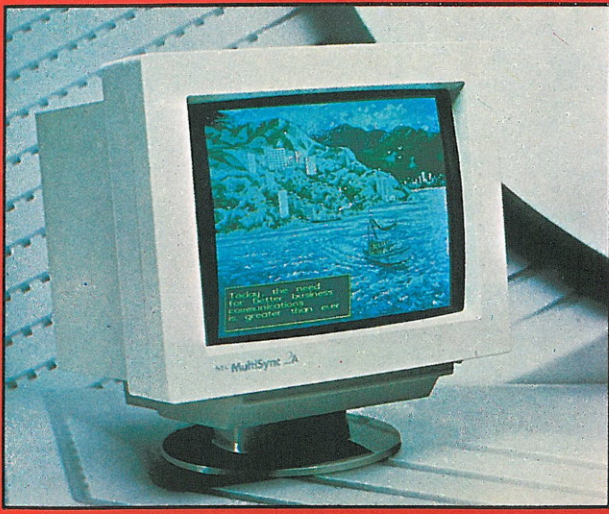
mikro

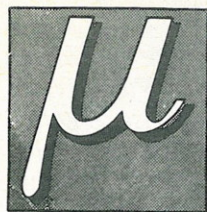
számítógép

magazin

Ára: 30 Ft







mikro számítógép magazin

7. ÉVFOLYAM
1989/2. SZÁM

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
(tervezőszerkesztő)
Simonyi Endre
Szebenzki Sándor
Szulyovszky Csaba
Tamásné Lakó Erika
Terebessy Ákosné

Címképünk:
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 jelzőszámmal.

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



Szika Lapnyomda
Budapest (88-1966)
Felelős vezető:
Csöndes Zoltán vezérigazgató

INDEX: 25 629
ISSN 0236-6088

TARTALOM

2	Technologizált 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 rejtelmei
6	Verseny a javából!
6	Hangos fejbeállítás
7	Ne bánts a magnót!
8	Hova tovább?!

CSIPEGETŐ

12

12	Kilépés a sprite-térbe
12	Bájt sorozatok tömörítése
13	Örökélet
13	Képcsere – bitenként
14	Video-RAM
15	Az ugrálgató
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	Dőlnek a dominók
27	Mi a manó?

SAKK

42

42	Centrumkontroll
----	-----------------

AZ OLVASÓ ÍRJA

43

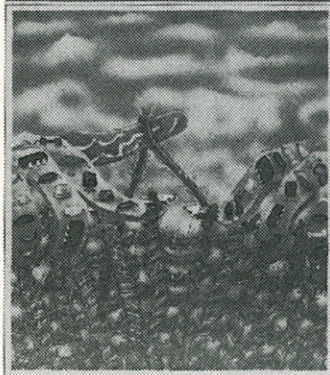
KÖNYVEK – HÍREK – ÉRDEKESSÉGEK

45

PONTVADÁSZAT

47

**μ mikro számítógép
magazin**



MEGOLDÁS ÉS KÖTÉS

1989/2

Nyugat-Európa konferenciáit és kiállításait járva, az ember egyre több olyan vállalkozással találkozunk, amelyeknek a fő terméke a számítógépre készített tananyag, vagy ahogyan 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 meg tanulni, akkor jobb, ha a CAL (Computer Assisted Learning), vagyis a számítógéppel segített tanulás kifejezést alkalmazzuk. A kompromisszumokra hajlóknak van még egy harmadik lehetőségük 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 távtanuláshoz (distance learning) a híve vagyok.

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

„Az oktatásban elért eredményeink közismertek, és ezek az eredmények igen jelentősek a legfejlettebb országokhoz viszonyítva. Mégis végrehajjtuk 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 hatásos voltak oktatási rendszerünkre is: itt is túrtuk 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 régen jelent meg először az országban.

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

kesztési trükkjei. Mindezeket vastos és kevésbé vastos 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 tapasztalattal gyűjtötték 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 rendszerekkel 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 rendszer alkalmazásaként behívva lehet például a kezeléshez szükséges ismereteket megtanítani.

Ennek az írásomnak nem az a célja, hogy a számítógépes tananyagfejlesztés technológiájáról részletes ismertetést adjak. De talán sikerült érzékeltetnem mennyire fontos számunkra, hogy ezen a területen is felzárkózzunk az európai élmezőnyhöz, hiszen különben 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 valamit a technológizált távtanulás bevezetése érdekében. A virágozzék 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 irigylem a művelődési minisztert, hiszen az iskolák meg vannak rakva a technológizált tananyag készítésére alkalmatlan házi 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 technológizált rendszerek bevezetése és az európai, illetve a nemzetközi élvonalhoz való felzárkózás nélkül meg lehetne valósítani.

Kovács Győző

Technológizált tanulás

kásos évi összefoglalót a hollandiai központú 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 képviselője, 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 társulajdonosaként tagja lesz ennek a nemzetközi szervezetnek. A csatlakozásnak az a célja, hogy elsősorban az informatikai képzésben nagyobb legyen a világra való tekintés, és lehetőség 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 iparilag fejlett országok, de jó néhány volt „fejlődő” ország, mint Tajvan, Dél-Korea vagy akár Thaiföld mögött is. Sokáig dőngöttük 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 üzleteket, 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 szellemű Kereskedelmi Minisztériumtól, hogy illúziók helyett a valóságos értékeket fogja védeni, és a jelenlegi korlátozások helyett az áruforgalom, és így a szoftverforgalom liberalizálását is meg fogja teremteni.

vel vitatkoztak 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. Kivétel persze a számítástechnika tanulása, de ott sem közömbös, hogy a tananyag készítőinek van-e pedagógiai érzékük és ismeretük.

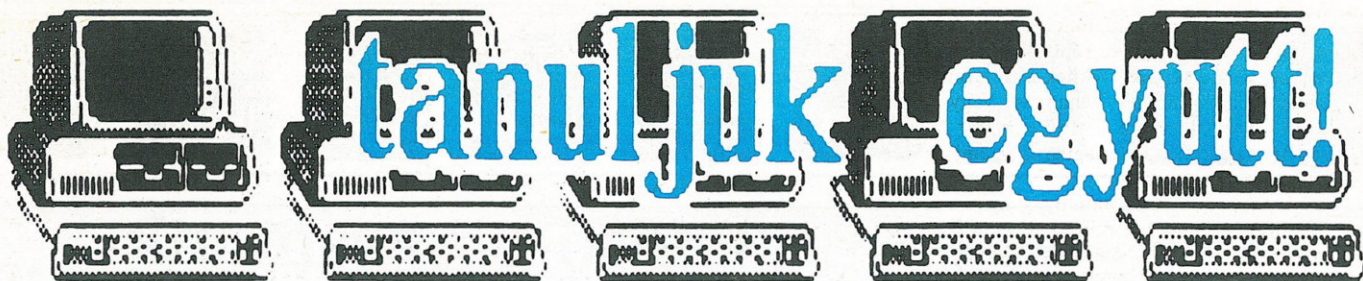
A számítógépes tananyag éppen ezért — pedagógiai értelemben — nem számítógépes program. Sokkal közelebb áll a tankönyvhöz, mint egy programhoz, de ugyanakkor a tananyag nem elektronikus tankönyv, mert akkor az már nem „courseware”.

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

A szerzői nyelv utasításokkal írja le a képernyő tartalmát (szöveg, ábra, fotó, grafika, animáció, visszakerdezés/teszt stb.). A szerzői rendszerekkel közvetlenül a képernyőn lehet dolgozni, legtöbbször ablaktechnikával, egerrel vagy tablettel 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 összehasonthasson vagy szétdarabolhasson.

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 jártas programozónak — például a formátum-, grafika-, szín-, fájlkezelést támogató nagyszámú beépített eljárása révén —, de kezelése sok tanulást, gyakorlást, első ránézésre „pilótavizsgát” 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 rendszerfájloktól.

A 3.XX verziók valamelyike a Pascalal 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 hajlékonylemezen (floppyn) 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 aktíválhatók — az MS/DOS hozzáfér az azokat tartalmazó könyvtárhoz —, az indítás a TURBO paranccsal 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 tárba.

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 paranccsal léphetünk be. A szerkesztő a 3. ábrán látható képpel jelentkezik be.



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 Pascal 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.

3. Első programunk

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 alpműveletet, 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 lemezes fájlra, hanem az ún. standard perifériákra vonatkozik — többek között a billentyűzet és a képernyő ilyen —, a fájlneveket nem szükséges kiírni. Ha ugyanis az eljárásnévben 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 feltételezi, 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 ' (apostrof) 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 közöltek). A szerkesztésből a Ctrl-K Ctrl-D-vel kilépve megjelenik a 2. ábrán bemutatott parancs főmenü. A programot az R paranccsal futtathatjuk. A fordító a programot először lefordítja, majd elindítja.

Az adatok bevitele 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 eljárásnevekben 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 bájtban 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= 3.3333333333E-01
>

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.

```

6. ábra

7. ábra

Első gyakorló programunk
a=1
b=3
a+b=4
a-b=-2
a*b=3
a/b= 3.3333333333E-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. Akiük e téma érdekel, 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 kereszteződésnek kell kezelni. A kereszteződéseket a település térképének bal felső sarkától kiindulva sorfolytonosan számozzuk. Utca a két kereszteződés közötti útszakasz, melyet tehát a két kereszteződés közötti ződés 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 tonnánál nehezebb gépjárművel behajtani tilos
 3. Kötelező 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 ház-tömb 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 gépjármű súlyát,
- a gépjármű indulási és érkezési helyét a kereszteződés számával.

A program határozza meg azt a keresztezőssorozatot, 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 említjük meg, hogy a javasolt programnyelv a Pascal vagy a C volt, továbbá az értékelésnél szempont a változókkal (változónevekkel) való takarékoskodás is.

Igen sok panasz hangzott már el a C16, C Plus/4 gépek kazettás egységeinek inkompatibilitásáról. Magyarán mondva: 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 csúcscsokoz 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.

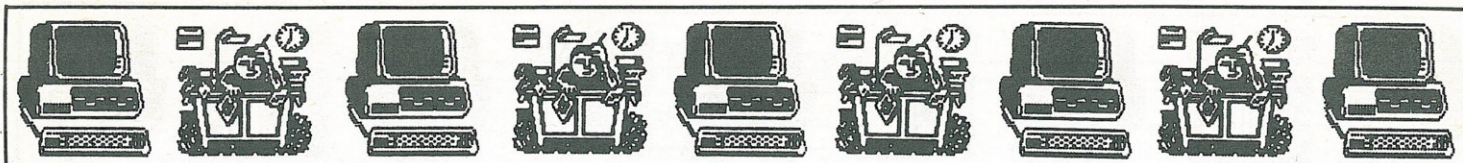
A fejbeállítás elve roppant egyszerű. Az adatmagnótól műszaki szempontból ugyanazt várjuk el, mint más, zenét, beszédet visszaadó készüléktől: minél szélesebb frekvenciasávot vigyen át, azaz jól

vigye át a magas hangokat. Ebben a megközelítésben tehát a jó minőségű adatbejátszást erősítőn keresztül hallgatva nagyon szép, éles hangzású nyivakolást hallunk. 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 árnyékoló vezetékhez kötött tucheldugó vagy a magnó házába csavarozott Jackdugó-aljzat.

Ehhez a művelethez először természe-

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 készült kivezetésről vehetjük le a jelet, ide kell az árnyékoló kábel belső erét egy 4,7–10 nF-os kondenzátoron keresztül kötni. A panelon ez a pont így is felismerhető, hogy ide csatlakozik alulról 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 erősítőn — ami lehet egy táskarádió magnóbemenete is — a kazetta jellegzetes hangját, de műsoros kazettát is lejátszhatunk így. Az alsó fedél visszaszerelése után a csavarokat húzzuk meg jól, mert ezek is befolyásolják a fejjállást! Az új árnyékolt vezeték a régi mellett a gumiszemen keresztül bújthatjuk ki.

Ezek után már szinte gyerekjáték a fejjállítást bármilyen programhoz. A fejjállító csavart forgatva figyeljük a hang magas-

ságát, élességét — vájt fülűek esetleg a szalagzajt is —, mert ezeket kell maximumra állítani. Ha jó a program, bárhol is származik, az ekként beállított fejjel 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 magánra menteni. A túl gyakori fejjállítgatás nem használ a rögzítőrendszernek, ezért

javasolom csak egy magnó felhasználását.

Ez az eljárás egyébként nagyobb biztonságot ad a tape-to-tape rendszerű programátvitelhez 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 „fűlre” megtalálhatjuk, s egy memóriatörlés után indíthatjuk a bejátszást.

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 működtető program megírásánál abból a feltételezésből indultunk ki, hogy a saját magnóval kimentett programok ugyanazzal be is tölthetők.

Hogy mindenki mindenkor a saját magnóját használhassa, egy program átvételéhez két gépet és két kazettás egységet alkalmazunk. A két gépet egy szabványos CBM csatlakozókábelrel a soros buszcsatlakozókon összekapcsoljuk. Az átmásolandó programot — programokat — tartalmazó kazettát az egyik (ezentúl 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 másik — vevő — géphez kapcsolt magnóban lévő szalagra íródik. Ehhez sem külön hardverre, sem szoftverre nincs szükség, ez a gép áramköri 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-

```
>3000 20 1A 30 A9 FD 25 01 85 : #FF00FF
>3008 01 20 83 30 C9 7F F0 79 : #FF00FF
>3010 C9 EF D0 F5 20 98 30 4C : #FF00FF
>3018 00 30 A2 00 A9 93 20 D2 : #FF00FF
>3020 FF B0 2E 30 30 07 20 D2 : #FF00FF
>3028 FF E8 4C 21 30 60 43 4F : #FF00FF
>3030 50 59 20 56 41 47 59 20 : #FF00FF
>3038 56 49 53 53 5A 41 20 42 : #FF00FF
>3040 41 53 49 43 20 42 41 3F : #FF00FF
>3048 20 28 43 2F 3C 52 55 4E : #FF00FF
>3050 53 54 4F 50 3E 29 11 11 : #FF00FF
>3058 11 11 41 20 40 41 53 4F : #FF00FF
>3060 4C 41 53 20 4D 45 47 53 : #FF00FF
>3068 5A 41 4B 49 54 41 53 41 : #FF00FF
>3070 20 41 20 3C 43 54 52 4C : #FF00FF
>3078 3E 20 47 4F 4D 42 42 41 : #FF00FF
>3080 4C 21 A0 A9 00 20 70 DB : #FF00FF
>3088 60 58 A9 93 20 D2 FF A9 : #FF00FF
>3090 FD 25 01 85 01 4C 0A 80 : #FF00FF
>3098 78 A9 EF 2D 06 FF 8D 06 : #FF00FF
>30A0 FF A9 0F 85 00 A9 F7 25 : #FF00FF
>30A8 01 85 01 20 E0 30 AD 00 : #FF00FF
>30B0 FF 8D 19 FF A0 00 8C 08 : #FF00FF
>30B8 FF 20 EE 30 8C 30 FD AD : #FF00FF
>30C0 08 FF C9 FB D0 E5 A9 10 : #FF00FF
>30C8 0D 06 FF 8D 06 FF A9 08 : #FF00FF
>30D0 05 01 85 01 A9 EE 8D 19 : #FF00FF
>30D8 FF A9 80 8D FC 07 58 60 : #FF00FF
>30E0 A9 FD 25 01 AA A9 10 25 : #FF00FF
>30E8 01 D0 FA 86 01 60 A9 02 : #FF00FF
>30F0 05 01 AA A9 10 25 01 F0 : #FF00FF
>30F8 FA 86 01 60 00 FF 80 00 : #FF00FF
```

1. lista

2. lista

```
. 3000 20 1A 30 JSR #301A
. 3003 A9 FD LDA #FFD
. 3005 25 01 AND #01
. 3007 85 01 STA #01
. 3009 20 83 30 JSR #3083
. 300C C9 7F CMP #7F
. 300E F0 79 BEQ #3089
. 3010 C9 EF CMP #EF
. 3012 D0 F5 BNE #3009
. 3014 20 98 30 JSR #3098
. 3017 4C 00 30 JMP #3000
. 301A A2 00 LDX #00
. 301C A9 93 LDA #93
. 301E 20 D2 FF JSR #FFD2
. 3021 B0 2E 30 LDA #302E,X
. 3024 30 07 BMI #302D
. 3026 20 D2 FF JSR #FFD2
. 3029 E8 INX
. 302A 4C 21 30 JMP #3021
. 302D 60 RTS
```

3. lista

```
. 3083 A9 00 LDA #00
. 3085 20 70 DB JSR #DB70
. 3088 60 RTS
. 3089 58 CLI
. 308A A9 93 LDA #93
. 308C 20 D2 FF JSR #FFD2
. 308F A9 FD LDA #FFD
. 3091 25 01 AND #01
. 3093 85 01 STA #01
. 3095 4C 0A 80 JMP #800A
. 3098 78 SEI
. 3099 A9 EF LDA #EF
. 309B 2D 06 FF AND #FF06
. 309E 8D 06 FF STA #FF06
. 30A1 A9 0F LDA #0F
. 30A3 85 00 STA #00
. 30A5 A9 F7 LDA #F7
. 30A7 25 01 AND #01
. 30A9 85 01 STA #01
. 30AB 20 E0 30 JSR #30E0
. 30AE AD 00 FF LDA #FF00
. 30B1 8D 19 FF STA #FF19
. 30B4 A0 00 LDY #00
. 30B6 8C 08 FF STY #FF08
. 30B9 20 EE 30 JSR #30EE
. 30BC 8C 30 FD STY #FD30
. 30BF AD 08 FF LDA #FF08
. 30C2 C9 FB CMP #FB
. 30C4 D0 E5 BNE #30AB
. 30C6 A9 10 LDA #10
. 30C8 8D 06 FF ORA #FF06
. 30CB 8D 06 FF STA #FF06
. 30CE A9 08 LDA #08
. 30D0 05 01 ORA #01
. 30D2 85 01 STA #01
. 30D4 A9 EE LDA #EE
. 30D6 8D 19 FF STA #FF19
. 30D9 A9 80 LDA #80
. 30DB 8D FC 07 STA #07FC
. 30DE 58 CLI
. 30DF 60 RTS
. 30E0 A9 FD LDA #FFD
. 30E2 25 01 AND #01
. 30E4 AA TAX
. 30E5 A9 10 LDA #10
. 30E7 25 01 AND #01
. 30E9 D0 FA BNE #30E5
. 30EB 86 01 STX #01
. 30ED 60 RTS
. 30EE A9 02 LDA #02
. 30F0 05 01 ORA #01
. 30F2 AA TAX
. 30F3 A9 10 LDA #10
. 30F5 25 01 AND #01
. 30F7 F0 FA BEQ #30F3
. 30F9 86 01 STX #01
. 30FB 60 RTS
```



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 kívánnak látni, esetleg azt saját igényüknek és ötleteiknek megfelelően át akarják alakítani.

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

a program beírása előtt is elolvasható a hexdump lista melletti, 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 „csíkozta”).

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 részlet megszünteti a képernyő elsötétítését, kikapcsolja a kazettamotort, a keretet 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-tól 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

Csány 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 számítástechnikananár,
programozó matematikus

Budapesti Műszaki Egyetem,
Villamosmérnöki Kar

Budapest, Egy József u. 18.

Műszer- és irányítástechnika szak,
informatika szak

Nehézipari Műszaki Egyetem

Kohó- és Fémipari 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

Vasútüzemti 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	
Archiv 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

— M E G O L D Á S O K

Sorozatunkat elsősorban középis-
kolásoknak szánjuk, de reméljük,
hogy minden olvasónknak tanulási
lehetőséget és szórakozást nyújt.

A feladatok a Nemes Tihámér or-
szágos számítástechnikai verseny
színpadának felelnek meg. Minden
esetben olyat választunk, amely rö-
viden, gyorsan megoldható, de a
megoldáshoz ötletre van szükség.
A megoldást mindig a következő
számban közöljük.

Mivel változatosságra törek-
szünk, különböző programozási nyel-
veket használunk. Az is előfordul
majd, hogy egy feladatra több pro-
gramnyelven is közlünk megoldást,
ezzel is elősegítve az ismeretszer-
zést.

A szerkesztőség várja az olvasók,
a versenyzők leveleit. A legötlete-
sebb program beküldőjét könyvtal-
vánnyal jutalmazzuk. Ne feledjenek
azonban a programhoz leírást is mel-
lékelni!

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, tekint-
sük át röviden megint.

A tábla belső ábrázolását
t\$(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 je-
lentőségű, koordinátáit tx és ty
tartalmazza.

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

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

A tábla kirajzolása után a
főprogram (6000—6140) egy
végtelen ciklusban figyeli a bil-
lentyűzetten beadott parancso-
kat. A kurzormozgató nyilak-
kal vagy a botkormánnyal le-
het a mezőket a táblán tologat-
ni (6050—6080). A tényleges
mozgatást a le (2000), fel
(2100), a jobbra (2200) és balra
(2300) rutinok végzik el, mint
ahogy a t\$ megfelelő változta-
tását is. A tábla összekeverése
a 0 vagy a tüzgomb lenyomá-
sával kezdeményezhető. Ekkor
a kever szubrutin (3000—3180)
indul el, és újabb billentyű le-
nyomásáig véletlenszerűen to-
logatja a mezőket.

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

Vizsgáljuk meg először,
hogy az ember hogyan kezd
hozzá a tizenötös játék kира-
kásához! A legtöbb ember elő-
ször megkeresi az egyes számú
mezőt, és azt próbálja a helyre
vinni. Utána megkeresi a
kettes mezőt, és azt is a helyére
továbbítja. Eközben a többi
mező helyzetével keveset törőd-
dik. 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, ak-
kor a következő szám helyre
rakásának menetét egyértel-
műen meghatározza a szám
helyzete. Így egy szám helyre
rakásához a gépnek maximum
tizenöt helyrerakási algorit-
musra van szüksége. Ez össze-
sen legfeljebb 125, ami nem
olyan sok; a számítógép a DA-
TA sorokban leírt mozgatósi
sorrendek segítségével köny-
nyen „megtanulhatja” őket.

Feltétel volt, hogy az üres
mező helyének rögzítettnek
kell lennie. Ez legegyszerűb-
ben 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 helyre rakása után
ugyanoda visszük, függetlenül
attól, hogy az adott számot ho-
gyan vittük a helyére. Ily mó-
don a következő szám helyre
rakása előtt az üres mező jól
meghatározott helyre kerül.

Az eddig elmondottak alap-
ján a program már megírható.
A továbbiakban már csak a fi-
nomítással kell törődnünk.

Az így megírt program
ugyan megoldás, de sok oda-
vissza mozgatást tartalmaz.
Sok esetben lehetőség van két
egymást követő szám helyre
rakása között akár több mozdi-
tás összevonására is. Ehhez
szükséges, hogy a DATA sor-
ból kiolvasott mozgatósi sor-
rendet ne azonnal, hanem csak

a következő DATA sor adatai-
nak megvizsgálása után rajzol-
juk ki a képernyőre. Ekkor vi-
szont a következő szám helyé-
nek meghatározása sem lesz
olyan egyszerű, hiszen ezek a
„késleltetett” mozgatósi sorok
lehet, hogy megváltoztatnák a
következő szám helyzetét is.
Ebből látszik, hogy a hatéko-
nyabb program megírásának
ára van.

A program másik finomítá-
sának 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, megbízha-
tóbb eljárást generálni?

Természetesen lehet, s egy
példa a mellékelt programban
is látható. Ez a <t>-re meg-
határozza, hogy melyik DATA
sört kell létrehozni, és ebbe
a sorba a <t> után következő
mozgásokat megfelelően kó-
dolva beleteszi. A „tanulás”
végét <v>-vel kell jelezni.
Bár ezzel a módszerrel a hibák
nem lehet teljes mértékben
kiküszöbölni, az, hogy a bevitt
mozgások a képernyőn követ-
hetők, a megbízhatóságot lé-
nyegesen javítja.

Vizsgáljuk meg most részle-
tesen a kirak rutint (5000). Az
5040—5160-as sorok között ta-
lálható a számok kirakását
végző ciklus. Az n=0 esetén az
üres mezőt rakjuk helyre.
A ciklus csak 13-ig tart, mert a
tizenhármas mező helyre raká-
sa után a tábla már biztosan ki
van rakva.

Két szám helyre rakása kö-
zött az e\$ teszi lehetővé az
oda-vissza mozgatás kiküszö-
bölését. Az 5050-es sorban a
keres rutint hívjuk meg, amely
az m\$-ban az n. mező helyre
rakásához szükséges mozga-
tást tartalmazza. Amíg a moz-
gatósi az e\$-ban tárolt pont
ellentétes, semmi nem történik
(5090). Ha nem így van, akkor
az e\$ fennmaradó elemeinek
megfelelő mozgatósi történik
(5100—5120). Az m\$-ban meg-
maradt elemek átkerülnek
e\$-ba, és a következő ciklus-

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

TIZENÖTÖS
JATEK

5-BALRA
6-FEL
7-LE
8-JOBBRA

0-KEVERD
OSSZE

SPACE-
RAKD KI

ban ezek összevonhatóságát vizsgáljuk a következő számjegy mozgásával.

Mivel így a mozgás mindig egy ciklussal lemarad, szükségesek 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\$-ban létrehozza a tényleges állást — ami, ha ha nem törekednék egymást követő lépések összevonására, 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 rajzóprogram.

9. feladat:

Rekurzív rajz

Írjon programot, amely beolvas egy számot és kirajzolja az ennek megfelelő fokszámú rekurzív görbét.

Pintér Gábor

```

10 REM Tizenötös játék
1988.06.29.
Pintér Gábor

20 RUN 6000
30 CLEAR
DEF KEY ERASE
LET RT=PEEK 23730+256*PEEK 23731
SAVE "15 V1.0" LINE 40
POKE PEEK 23631+256*PEEK 23632+2.181
PAUSE 30
SAVE "cBeta 3.1"CODE RT+1,65636
-RT
STOP
BORDER 7
PAPER 7
INK 0
CLEAR RT
LOAD "cBeta 3.1"CODE
CLS
RANDOMIZE USR 58419
50 LIST FORMAT 1
CSIZE 7,8
60

1000 DEF PROC kép
1010 REM A kép nem változó
részét rajzolja ki.

1020 LOCAL i
1030 CLS
1040 PRINT CSIZE 8,11;AT 1,22;"TIZE
NOTOS" TAB 24;"JATEK" TAB 23
"5-BALRA" TAB 23;"6-KEFEL" TAB
23;"7-LE" TAB 23;"8-JOBBRA" TAB
23;"0-KEVERO" TAB 25;"OSSZE
" TAB 23;"SPACE-" TAB 25;"RAK
D-KT"
1050 CSIZE 8
INK 5
PRINT AT 0,0
FOR i=1 TO 21
NEXT i
1070 INK 0
1080 END PROC

1200 DEF PROC mezó x,y
1210 REM AZ (X,Y) MEZŐT
RAJZOLJA MEG.

1220 LOCAL s$
1230 LET s$=STR$ CODE t$(x,y)
1240 CSIZE 32
1250 IF CODE t$(x,y) THEN
PLOT PAPER 6;40*x-32,199-40*y
"
CSIZE 8,16
PLOT PAPER 6;40*x-16-4*LEN s$
191-40*y,s$
ELSE
PLOT PAPER 5;40*x-32,199-40*y
"
1260 CSIZE 8
PAPER 7
1270 END PROC

1400 DEF PROC alap
1410 REM A kezdőállapot
beállítása.

1420 LOCAL i
1430 DIM t$(4,4)
1440 REM A tábla.
1450 LET t$=""
1460 REM AZ űres mező helye.
RANDOMIZE
1470 FOR i=1 TO 4
FOR j=1 TO 4
IF i#j AND j#4 THEN
LET t$(j,i)=CHR$(0)
ELSE
LET t$(j,i)=CHR$(4+i+j-4)
mezó j,i
NEXT j
1510 END PROC

2000 DEF PROC le
2010 REM Mozdítás lefelé.

2020 BEEP .02,30
2030 IF ty>1 THEN
LET t$(tx,ty)=t$(tx,ty-1)
LET ty=ty-1
LET t$(tx,ty)=CHR$(0)
mezó tx,ty
mezó tx,ty+1
2040 END PROC

2100 DEF PROC fel
2110 REM Mozdítás felfelé.

2120 BEEP .02,27
2130 IF ty<4 THEN
LET t$(tx,ty)=t$(tx,ty+1)
LET ty=ty+1
LET t$(tx,ty)=CHR$(0)
mezó tx,ty
mezó tx,ty-1
2140 END PROC

2200 DEF PROC jobbra
2210 REM Mozdítás jobbra.

```

```

2220 BEEP .02,17
2230 IF tx>1 THEN
LET t$(tx,ty)=t$(tx-1,ty)
LET tx=tx-1
LET t$(tx,ty)=CHR$(0)
mezó tx,ty
mezó tx+1,ty
2240 END PROC

2300 DEF PROC balra
2310 REM Mozdítás balra.

2320 BEEP .02,23
2330 IF tx<4 THEN
LET t$(tx,ty)=t$(tx+1,ty)
LET tx=tx+1
LET t$(tx,ty)=CHR$(0)
mezó tx,ty
mezó tx-1,ty
2340 END PROC

3000 DEF PROC kever
3010 REM A játék összekeverése.

3020 LOCAL i,r,h$,l$
3030 LET r=1
3040 DO
LET h$="1234"
3050 LET h$(5-r)=" "
3060 IF tx=4 THEN
LET h$(1)=" "
3070 IF tx=1 THEN
LET h$(4)=" "
3080 IF tx=2 THEN
LET h$(3)=" "
3090 IF ty=4 THEN
LET h$(3)=" "
3100 IF ty=1 THEN
LET h$(2)=" "
3110 LET h$(1)=" "
3120 FOR i=1 TO 4
IF h$(i)<" " THEN
LET t$(i,h$(i))
NEXT i
3130 NEXT i
3140 LET r=VAL t$(INT (LEN t$*RAND)
+1)
3150 ON r
balra
fel
jobbra
3160 LOOP WHILE INKEY$="" AND (IN 3
1)=32 OR IN 31=0
3170 DO
LOOP UNTIL INKEY$="" AND (IN 3
1)=32 OR IN 31=0
3180 END PROC

4000 DEF PROC keres n, REF m$
4010 REM A táblázatban megkeresi
n-et; (x,y) helyen van.

4020 LOCAL x,y,i,j,n$
4030 DIM n$(4,4)
4040 FOR i=1 TO 4
LET n$(i)=t$(i)
NEXT i
4050 LET x=tx
LET y=ty
4060 FOR i=1 TO LEN e$
ON VAL e$(i)
balra
fel
kjobbra
4080 NEXT i
4090 REM n$-ben a tényleges
állás van.

4100 LET x=x-1
4110 LET y=y-1
4120 LET y=y+1
4130 IF n$(x,y)=CHR$(n) THEN
GO TO 4170
4140 IF y<4 THEN
GO TO 4120
4150 LET x=x+1
4160 GO TO 4130
4170 RESTORE @000+100*n+10*x+y
READ m$
REM Hogyan kell ezt a helyére
vinni?
4180 END PROC

4200 DEF PROC kle
4210 REM Mozdítás lefelé.
REM Kijelzés nélkül.

4220 IF y>1 THEN
LET n$(x,y)=n$(x,y-1)
LET n$(x,y)=CHR$(0)
4230 END PROC

4300 DEF PROC kfel
4310 REM Mozdítás felfelé.
REM Kijelzés nélkül.

4320 IF y<4 THEN
LET n$(x,y)=n$(x,y+1)
LET n$(x,y)=CHR$(0)
4330 END PROC

4400 DEF PROC kjobbra
4410 REM Mozdítás jobbra.
REM Kijelzés nélkül.

4420 IF x>1 THEN
LET n$(x,y)=n$(x-1,y)
LET n$(x,y)=CHR$(0)
4430 END PROC

```

```

4500 DEF PROC kbalra
4510 REM Mozdítás balra.
REM Kijelzés nélkül.

4520 IF x<4 THEN
LET n$(x,y)=n$(x+1,y)
LET n$(x,y)=CHR$(0)
4530 END PROC

5000 DEF PROC kirak
5010 REM Kirakja a játékot.

5020 LOCAL n,i,j,m$,e$
5030 LET e$=""
5040 REM Átvitel
FOR n=0 TO 13
REM 13 szám
+ kezdeti beállítás.
5050 keres n,m$
5060 LET i=1
REM Mozgató.
5070 IF i=LEN m$ THEN
GO TO 5160
5080 IF e$="" THEN
5090 IF VAL m$(i)+VAL e$(LEN e$)=5
THEN
LET e$=e$+(i TO LEN e$-1)
GO TO 5140
REM Odá-vissza mozgató kiku-
szobólása
5100 FOR j=1 TO LEN e$
ON VAL e$(j)
balra
fel
jobbra
5120 NEXT j
5130 LET e$=m$(i TO )
GO TO 5160
5140 LET i=i+1
5150 GO TO 5070
5160 NEXT
5170 REM Vege.
5180 FOR j=1 TO LEN e$
ON VAL e$(j)
balra
fel
jobbra
5200 NEXT j
5210 END PROC

6000 REM A főprogram

6010 alap
LET tn=0
6020 DO
LET k$=INKEY$
6030 IF k$ IN 31
6040 IF (k$ OR k$)=32 AND k$="" TH
EN
GO TO 6030
6050 IF k$ AND k$=2 OR k$="5" OR k
$=CHR$(8) THEN
balra
IF tn THEN
LET d$=d$+"1"
6060 IF k$ AND k$=1 OR k$="8" OR k
$=CHR$(9) THEN
jobbra
IF tn THEN
LET d$=d$+"4"
6070 IF k$ AND k$=4 OR k$="6" OR k
$=CHR$(10) THEN
le
IF tn THEN
LET d$=d$+"2"
6080 IF k$ AND k$=0 OR k$="7" OR
k$=CHR$(11) THEN
fel
IF tn THEN
LET d$=d$+"3"
6090 IF k$ AND k$=15 OR k$="0" TH
EN
kever
6100 IF k$="" THEN
kirak
6110 IF k$="1" OR k$="2" THEN
INPUT "Melyiket? ";tans
keres2 tans,d$
LET tn=1
BEEP .1,24
6120 IF k$="U" OR k$="U" THEN
BEEP .1,16
LET d$=d$+"*****"
KEYIN d$
LET tn=0
6130 LOOP
6140

7500 DEF PROC keres2 n, REF m$
7510 REM A táblázatban
megkeresi n-et;
(x,y) helyen van.
És létrehozza a DATA sor
elejét.

7520 LOCAL x,y
7530 LET x=1
7540 LET y=1
7550 LET y=y+1
7560 IF t$(x,y)=CHR$(n) THEN
GO TO 7600
7570 IF y<4 THEN
GO TO 7550
7580 LET x=x+1
7590 GO TO 7560
7600 LET m$=STR$(8000+100*n+10*x+y)
+ " DATA "
REM A DATA sor eleje
7610 END PROC

```

```

8011 DATA "111333"
8012 DATA "1133"
8013 DATA "1113"
8014 DATA "1113"
8015 DATA "111"
8016 DATA "11333"
8017 DATA "1333"
8018 DATA "113"
8019 DATA "113"
8020 DATA "1333"
8021 DATA "133"
8022 DATA "113"
8023 DATA "113"
8024 DATA "1333"
8025 DATA "133"
8026 DATA "113"
8027 DATA "113"
8028 DATA "1333"
8029 DATA "133"
8030 DATA "113"
8031 DATA "113"
8032 DATA "113"
8033 DATA "133"
8034 DATA "133"
8035 DATA "133"
8036 DATA "133"
8037 DATA "133"
8038 DATA "133"
8039 DATA "133"
8040 DATA "133"
8041 DATA "133"
8042 DATA "133"
8043 DATA "133"
8044 DATA "133"
8045 DATA "133"
8046 DATA "133"
8047 DATA "133"
8048 DATA "133"
8049 DATA "133"
8050 DATA "133"
8051 DATA "133"
8052 DATA "133"
8053 DATA "133"
8054 DATA "133"
8055 DATA "133"
8056 DATA "133"
8057 DATA "133"
8058 DATA "133"
8059 DATA "133"
8060 DATA "133"
8061 DATA "133"
8062 DATA "133"
8063 DATA "133"
8064 DATA "133"
8065 DATA "133"
8066 DATA "133"
8067 DATA "133"
8068 DATA "133"
8069 DATA "133"
8070 DATA "133"
8071 DATA "133"
8072 DATA "133"
8073 DATA "133"
8074 DATA "133"
8075 DATA "133"
8076 DATA "133"
8077 DATA "133"
8078 DATA "133"
8079 DATA "133"
8080 DATA "133"
8081 DATA "133"
8082 DATA "133"
8083 DATA "133"
8084 DATA "133"
8085 DATA "133"
8086 DATA "133"
8087 DATA "133"
8088 DATA "133"
8089 DATA "133"
8090 DATA "133"
8091 DATA "133"
8092 DATA "133"
8093 DATA "133"
8094 DATA "133"
8095 DATA "133"
8096 DATA "133"
8097 DATA "133"
8098 DATA "133"
8099 DATA "133"

```

Írásomat a Sprite-ok bonckés alatt című cikk (1987/12. sz.) kiegészítéséül 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-rutinnal a képernyőt az \$F9-es rasztorsortól 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 megtévesztő. Olyan, mintha történetesen PRINT-tel kiírhatnánk a képernyőre, a 28—29-es sorba. Sokan úgy hiszik, hogy a sprite-térbe csak spritel lehetünk ki. Most, bizonyítékkal, hogy nem spritel irunk ki, kirakhatunk néhányat ugyanoda. Az itt közölt program csak szemlélteti a módszert, és egy csík-srollt valósít meg alul-fölül, amely azután tetzés szerint variálható. Íme a program:

```

C000 SEI
    LDA #00
    STA DC0E    CIA I IRQ kikapcsolása
    LDA #01
    STA D01A    VIC raszter IRQ bekapcsolása
    LDA D011
    AND #7F
    STA D011    raszterregiszter átvitelének törlése
    LDA #2B
    STA 0314
    LDA #C0
    STA 0315    IRQ átirányítás $C02B-re
    LDA #F9
    STA D012    a sor, ahol a VIC IRQ-t vált ki
    LDA #00
    STA 9E      a táblázat mutatójának nullázása
                IRQ engedélyezve
    CLI
    RTS
C02B LDA D019
    STA D019    ez VIC IRQ-nál kötelező!
    LDA D011
    STA 02      az eredeti regisztertartalom mentése a nullás lapra

    LDA #80
    STA D011    a képernyő kikapcsolása
    LDY 9E      mutató betöltése
    LDA C055,Y  a bitkép táblázatból 1 bájt
    STA 3FFF    ez alapállapotban ($D000 = 151)
    LDA C05D,Y  a táblázat következő mutatójának betöltése
    STA 9E      és tárolása
    LDX #30
C04A DEX
    BPL C04A    várakozó ciklus
    LDA 02
    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

02 helyett A3,
9E helyett BD.

Ezenkívül a programot a 0000—3FFF tartományba kell vinni, hogy a VIC regisztereit 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. Könyvnyitendő 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 elemeit sorra összehasonlítjuk a 17. bájton 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. bájton kezdődő, tehát a keresendő sorozatot 1 bájttal 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 bájttal, 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 kigyűjtése. A megtalált sorozatot bemásoljuk egy táblázat következő üres helyére, ennek a helynek a sorszáma lesz az adott sorozat sorszáma. Vi-

Néhány tanács azoknak, akik most ismerkednek az Enterprise-zal és van EXOS 2.1 műszaki leírásuk.

AZ ESCAPE SZEKVENCIÁKRÓL

A szekvenciák rendkívüli módon egyszerűsítik a gép programozását. Az EXOS ellipszist rajzol, egyenest húz, ADSR-görbét definiál stb. Az escape szekvenciák használatakor lényegében egy sztringet készítünk, 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, grafikus 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 EXOS 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.

Nézzünk 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 szegmensben van, a 32 k-tól 44076—32768=11308 bájt 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 van.)

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 elérhetjük így: SPOKE 254,44076, érték). A végcímet hasonlóan határozhatjuk meg. Példánknál maradvány 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. (Erről az EXOS könyvben a 284. oldaltól olvashatunk.) Hardver szövegmódban (TEXT 40) a képernyősorok összekeveredhetnek, így ha valaki POKE-kal ír, meglepetések érhetik. A 3. listán látható kis rutint olyan nagyobb programokba érdemes beépíteni, ahol BASIC-ből a gyorsaság kedvéért POKE-ot használnánk. Paramétereit tekintve hasonlít 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 rájöhettek 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,B7,CB,15,CB")
17 CODE =HEX$("14,10,FA,C1,9,1,4,0")
18 CODE =HEX$("9,5E,23,56,D5,E1,1")
19 CODE EX=HEX$("0,0,9,B7,11,1,40")
20 CODE EA=HEX$("ED,52,36,41,C9,0")
```

3. lista

```
10 PROGRAM EXOS_DEMO
11 ALLOCATE 200
12 CODE R=HEX$("06,01,0E,18,16,20,F7")
13 CODE =HEX$("10,06,01,0E,19,16,1B,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$("10,3E,01,11")&WORD$(R+41)
17 CODE =HEX$("F7,01,18,07,06,56,49,44")
18 CODE =HEX$("45,4F,3A,3E,01,06,01,0E")
19 CODE =HEX$("01,16,1B,1E,01,F7,0B,3E")
20 CODE =HEX$("01,01,0A,00,11")
21 CODE =WORD$(R+72)&HEX$("f7,00,18,0A")
22 CODE =HEX$("1B,43,67,28,00,00,00,00")
23 CODE =HEX$("00,00,3E,01,01,06,00,11")
24 CODE =WORD$(R+94)&HEX$("F7,00,18,06")
25 CODE =HEX$("1B,41,00,02,E6,01,3E,1F")
26 CODE =HEX$("01,E6,01,11,00,00,ED,43")
27 CODE =WORD$(R+153)&HEX$("ED,53")
28 CODE =WORD$(R+155)&HEX$("F5,C5,D5")
29 CODE =HEX$("3E,01,01,06,00,11")
30 CODE =WORD$(R+151)&HEX$("F7,00,11")
31 CODE =HEX$("10,00,E1,19,E5,D1,01,10")
32 CODE =HEX$("00,E1,37,3F,ED,42,E5,C1")
33 CODE =HEX$("F1,3D,20,D6,C9,1B,45,00")
34 CALL USR(R,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 végéig 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 paramétereznünk a régi eljárásainkat, de ez még mindig kisebb munka, mint ha állandóan újra meg újra megírnánk 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.

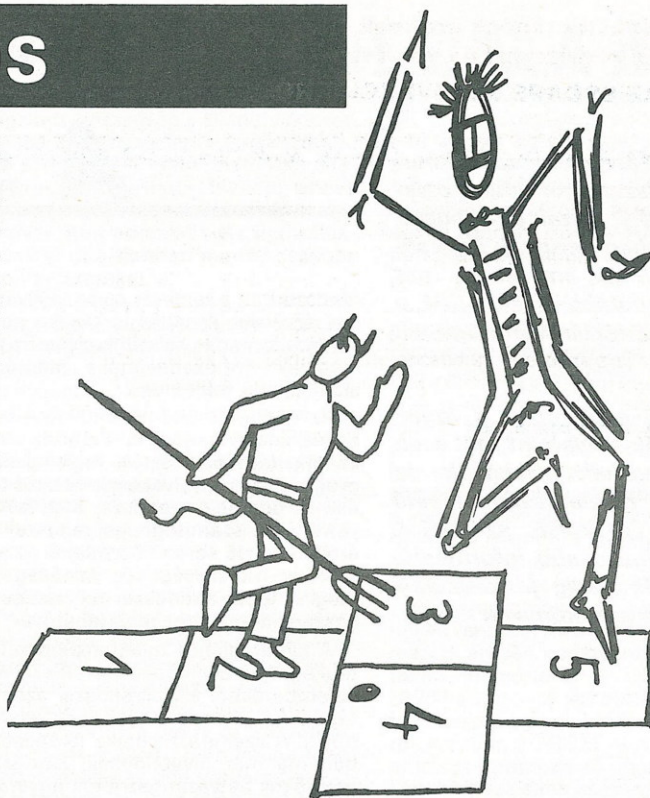
Sajnos, főleg a kezdők, akik megismerkednek egy programnyelvvvel, azt hiszik, hogy már profik, számukra nincs lehetetlen. De ha megnézi az ember egy feladatukat, bizony megrázó élményben lesz része, mivel a program(ocská) tele van fölösleges magyarázatokkal és rengeteg GOTO-val.

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

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

Az 1. lista egy IBM PC XT/AT PL/I fordító. Ez a program nagyon átláthatatlan: már ilyen kis feladathoz 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(80)), WORK CHAR(80);
ON END FILE(SZALAG) GOTO VEG;
DO WHILE('1'B);
GOTO OLV;
C1:IF SUBSTR(WORK,9,2)='EZ' THEN DO PUT SKIP
EDIT(WORK)(A(80)); END;
END;
GOTO VEG;
OLV:GET FILE(SZALAG)EDIT(WORK)(A(80));
GOTO C1;
VEG:END PRG1;
```

1. lista

```
PRG2:PROC OPTIONS(MAIN);
DCL SZALAG FILE INPUT ENV(MEDIUM(SYS001,2400)F(80)),
WORK CHAR(80);

ON END FILE(SZALAG) GOTO VEG;

DO WHILE ('1'B);
GET FILE(SZALAG) EDIT(WORK)(A(80));
CALL VIZSG;
END;

VIZSG;
IF SUBSTR(WORK,9,2)='EZ' THEN CALL PRINT;
END VIZSG;

PRINT;
PUT SKIP EDIT(WORK)(A(80));
END PRINT;

VEG;
END PRG;
```

2. lista

TOP - lista																	
Felhasználói programok	IBMS					Játék programok	IBMS										
	IBM	AMIGA	C-128	C-64	C-4 (6)		SPECTR.	ENTERP.	TVC	IBM	AMIGA	C-128	C-64	C-4 (6)	SPECTR.	ENTERP.	TVC
1 Aliga Paint						1 Maniacs M. 00.						2 LLL					
2 Geos 1.6 d.						3 Monopoly 00.						4 Skyfox 00.					
3 Rockmon v5.3						5 TV Ball						6 Elite					
4 Windows						7 Defender otc.						8 Barbarian 00.					
5 Aliga Cad +						9 Pirates!						10 Batman					
6 Printfox																	
7 Printmaster																	
8 Disk manager																	
9 Music master																	
10 Starpainter																	

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
Diákszerkesztőség

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.

A SZEREPLŐK BEMUTATÁSA

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 alrendszert 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 beviteli egységekből, vezérlőegységekből és egyéb berendezésekből áll, amelyekhez olyan „elvont” alkotórészek is járulnak, mint az utasításrendszer (gépi kód), a címkezező alrendszer stb.

A hardverréteg 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 fejlesztőinek.

Az operációs rendszer olyan kapcsolatot teremt például a fordítóprogramot íróknak, amely csökkenti a programozási igényt és egyszerűíti a gép egyes szolgáltatásainak igénybevételét. Az operációs rendszer által létrehozott kapcsolat természetesen nem képes minden esetben a hardverréteg teljes eltakarására. Egy fordítóprogram kódgenerátorának írása nem nélkülözheti a tárgynyelv 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 réteghez hasonlóan a nyelvek sem képesek az operációs rendszer teljes elfedésére. Sok programozási nyelv például lehetővé teszi az operációs rendszer egyes szolgáltatásainak igénybevételét 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 használhatóságot teszi lehetővé.

A hierarchiához tulajdonképpen legfelső rangként 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 számunkra nem az összetétele, hanem a kiszolgálása a fontos. Állandóan szem előtt kell tartani, hogy a felhasználókat elsősorban a rendszertől kapható információ érdekli. 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 határozott támogatását. Ezért a jó számítógépes rendszernek olyan felhasználói kapcsolatot és feladatmegoldó környezetet kell nyújtania, amely csak minimális mértékű számítógépes szakismertéget igényel.

Az operációs rendszer fogalma

Az operációs rendszer nem teljesen egyértelmű fogalom. Függsz a hagyományoktól, a hardverreteg 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 részeinek tekintett alrendszereket tartalmazza. Egyes konkrét rendszereknél az alrendszerek halmaza szűkülhet vagy bővíthet.

1. A rendszer működésébe gépkezelői beavatkozást megvalósító alrendszerek. Ide tartozik a rendszer átkonfigurálása, a működési statisztikák lekérdezése, programok és berendezések állapotának vizsgálata stb.

2. A rendszer és a külvilág közötti kapcsolatok legalsó szintjét megvalósító beviteli/kiviteli (B/K) alrendszerek. Ide tartozik a B/K igények sorkezelője, a B/K műveletet indítása és a külső egységekről érkező megszakítások kezelése.

3. A multiprogramozás vezérlését megvalósító alrendszerek. 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 tekinti.

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

egy nyelv — az egyszerű mikrogépeknél 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 kapcsolatban. Az első és második generációs gépeknél például vagy nem is volt operációs rendszer, vagy jóformán kizárólag hardverelemekből állt. Különléte kapcsolókból, címkiválasztó kulcsokból, indítókból, megállítókból, lépésenkénti végrehajtást kiváltó 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 természetesnek tart.

Hardver alapfogalmak

A számítógépes rendszerek legalsó rétege, a hardver, közelebbről 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 rövidítése: CPU) vagy röviden központi egység elnevezésű dobozként jelennek meg, ami más alkatrészeket is tartalmaz. Bár a központi egység növekvő teljesítménye mellett is a számítógépes rendszer összköltségének egyre csökkenő hányadát képviseli, mégis, több szempontból is, a rendszer legfontosabb összetevője. Az operációs rendszerek felépítésének szempontjából a központi egység néhány 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 memóriarekeszekre hivatkoznak. Ilyen központi egység programozása igen nehézkes lenne. A következő „fokozat” egy vagy több indexregiszter használata, amelyeknek tartalma — általában összeadással — módosíthatja az egyenes címet. Az indexeléssel is kombinálható indirekt címzés — a cím tartalma egy újabb cím, ahol az operandus van — sok lehetőséget tartogat az ügyes programozónak. Gyártottak már többszörös indirekciót megengedő központi egységeket is.

A központi memória méretének növekedésével együtt bővíteni kellett az utasítások címtartományát is. Ma már a 16 biten címezhető 64 kb-át abszolút minimumot jelent. Hogy ne kelljen az utasításokban 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 háttértárak fejlődése lehetővé tette a virtuális tárkezelés kialakulását, amikor is a felhasználó által látott, jóformán korlátlan méretű tárolónak csak egy kis része létezik a rendszerben operatív memóriaként, nagy része valami-

motorja I.

lyen háttértáron — általában mágneslemezen — van. A központi egység címzési mechanizmusa fölé az operációs rendszerben egy olyan virtuális tárkezelő alrendszer építenek, amely a felhasználó számára a címhivatkozások során igényelt információcsere után a háttértár és az operatív tár között automatikusan elvégzi.

Az utasításrendszer a hardverrétegben mindazon utasítások összessége, amelyeket a központi egység végére 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 utasításkészlet néhány tucat utasításra korlátozódott, a mai rendszereknél általában több száz különböző utasításkód használható. A kódon és a cím(ek)en kívül gyakran valamilyen járulékos információ is szerepel az utasításban, például az indexelés, indíktás stb.

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

Az utasítás megvalósító programokat szokták mikrokódnak, mikroprogramnak is nevezni. Az így létrehozott virtuális központi egység sok egyéb lehetőséget is tartalmazhat. Igen általános például több működési állapot megvalósítása. Ezek közül gyakoriak a kétállapotú architektúrák. Normál állapotban a bővített utasításkészletnek csak egy részalmeza használható, míg privilegizált állapotban a teljes utasításrendszer rendelkezésre áll. A privilegizált állapotot az IBM 370-en például „supervisor állapot”-nak hívják. A privilegizált állapot legtöbbször a megszakításokkal kapcsolatos, sok rendszerben a megszakítás fellépése kiváltja a privilegizált állapotba való átmenetet is. Természetesen szükség van egy olyan utasításra is, amely normál állapotban mesterséges megszakítást 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 megszakítási rendszerre azért van szükség, mert a számítógéprendszer olyan alkatrészeket is tartalmaznak, amelyek párhuzamosan is működtethetők. Gondoljunk például a B/K műveletekre: ezek egyrészt időigényesek, másrészt általában csak egy indítást igényelnek a központi egységtől, végrehajtásukról külön processzorok — csatornák — gondoskodnak. A megszakítási rendszer teszi lehetővé a párhuzamosan üzemeltethető egységek közötti koordinációt, és a központi egységen belül előálló speciális feltételekre — például a processzorhibára, amit akár a nullával való osztás, akár egy nem létező bájtra való hivatkozás okozhat — való megfelelő reakciót.

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

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

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

A megszakítási rendszert főleg az teszi bonyolulttá, hogy egy megszakítás feldolgozása közben újabb megszakítás érkezik, amit szintén kezelni kell, és ez a jelenség jóformán tetszőleges mélységben, egymásba skatulyázva is felléphet. A problémát a gyakorlatban úgy egyszerűsítik, hogy bizonyos megszakítások kezelésének (például processzorhiba) idejére letiltják minden más jellegű megszakítás fogadását. Ez azt jelenti, hogy ha ilyen megszakítás bekövetkezik, azt a központi egység várakoztatja az éppen feldolgozás alatt álló processzorhiba kezelésének befejezéséig. Ugyanakkor természetes, hogy valamilyen megszakítás kezelése közben a központi egység más megszakításokat elfogad. Egy nyomtató működése közben például egy lemezel való információcsere kényelmesen lefolytatható. Azt, hogy a megszakítórutinok is általában megszakíthatók legyenek, úgy oldják meg, hogy az állapotvektort 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 veremmutató pedig automatikusan átmegy az eggyel mélyebben elhelyezett elemre.

A közös cél minden esetben az, hogy a rendszer minél kevesebb ideig legyen elvágyva a külvilágtól megszakítás fogadására alkalmatlan állapotban.

Védelmi mechanizmusra azért van szükség, mert a megszakítások kezeléséből már egyenesen következik 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 megszakítás jön létre, és az operációs rendszer be tud avatkozni. Azt, hogy ilyenfajta védelemre szükség van, nem kell különösebben indokolni, hiszen például a felhasználói programok átirhatnák az operációs rendszer területének tartalmát, ami teljes anarchiához vezetne.

A központi egység védelme szintén a felhasználói programok ésszerűtlen viselkedése által okozható veszélyek elhárítását célozza. Ha például a program végtelen ciklusba kerül, az operációs rendszer nem kapja meg a vezérlést bizonyos, periodikusan ismétlődő feladatainak (például a képernyő tartalmának frissítése) ellátásához, ami nyilván nem megengedhető. A szokásos megoldás az ún. óramegszakítás, amely szabályos időközönként generálódik. Az óramegszakítás lehet állandó (például 1/50 másodperc), vagy szabályozható időközű. Maga a megszakítás egy regiszterben való számlálással valószínűsíthető 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 utasításkészlet számára elérhetők — hívjuk memóriának. Ha mágneslemezekről, mágnesszalagokról és egyéb hasonló memóriákról van szó, inkább a háttértár vagy külső tároló kifejezést használjuk.

A memória a hardverréteg nélkülözhetetlenül fontos része; nem véletlen, hogy egy számítógéppel kapcsolatban általában ez az első adat, amit megkérdeznek 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. Általánosságá váltak a csak olvasható (angolul ROM=Read Only Memory) memóriák, amelyekben a rendszer állandóan használt szoftverösszetevőit — például az operációs rendszer fontosabb rutinjait, fordítóprogramokat — tárolják folyamatosan. Egyes rendszerekben már a ROM előtt is alkalmaztak erősen korlátozott módon 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 törölhető, 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 valamivel kisebb, de még mindig 100 megabájtkoban mérhető kapacitással elfogadhatóan gyors véletlen elérést tesz lehetővé. A cserélhetőség feladásával az elérési idő tovább csökkent, a fix és cserélhető lemezek kombinációja pedig minden rendszer számára megvalósíthatóvá tette az alkalmazási környezethez legjobban illeszkedő memóriakonfiguráció kialakítását. Az ún. winchester-rendszer, valamint a hajlékony cserélhető lemezek bevezetésével a nagy gépek mellett a mini- és a mikrogépek is megkapták az olcsó, gyors és elég megbízható nagy kapacitású háttértárakat.

Egyéb perifériák

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

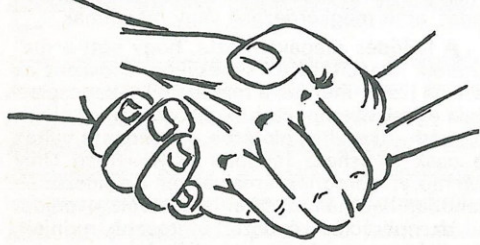
A nyomtatók az írógéptől indulva a sornyomtatókon keresztül a mátrix-, majd a lézernyomtatókig sokat fejlődtek s nagy változatosságot mutatnak, minden számítógépes rendszer nélkülözhetetlen részei.

A képernyő és a billentyűzet a manuális adatcsere 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 terminálok elterjedése, mivel így a nehézkes kötegelt rendszereket a legtöbb alkalmazási területen a felhasználót segítők, interaktív rendszerek váltották fel. Ez legalább egy hardvergenerációval felérő fejlődés volt.

Az adatátviteli vonalak a távközlési vonallal összekapcsolt számítógépekkel játszanak fontos szerepet. Az információátvitel itt bájtonkénti, a vonalra az adat egy modem közbeiktatásával jut. Az adatátvitelhez természetesen meglehetősen összetett szoftver tartozik, hiszen a kétoldalú kapcsolattartás bonyolult feltételeket teremt.

Bakos Tamás

Programozási fogások és



melléfogások

Cikksorozatomban a nyomtatásban közölt BASIC programokban sűrűn előforduló programozási hibákat mutatom be, ismertetve 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 ismertem 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 utasításpárral 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 négyzetszám 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 eredetihez 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
```

```
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"
```

4. lista

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

A TUDOMÁNSZERVEZÉSI ÉS INFORMATIKAI INTÉZET

előzetes megbeszélés szerint díjmentes programbemutatót tart (vidéken is) az általa forgalmazott oktatóprogramokból.
Horváth Zsuzsa 865-011/2663 mellék
vagy 813-197
Budapest, Pf. 454, 1372

Mintakötés nemcsak hölgyeknek

Egy kis kötéstechika

A kötött holmikát gépen is kötőtűvel készítik, bár az 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 készítetik. 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 tű megkötik a szemeket. Az ilyen típusú kötött kelmében a szemek a csavarvonal mentén helyezkednek el (3. ábra).

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

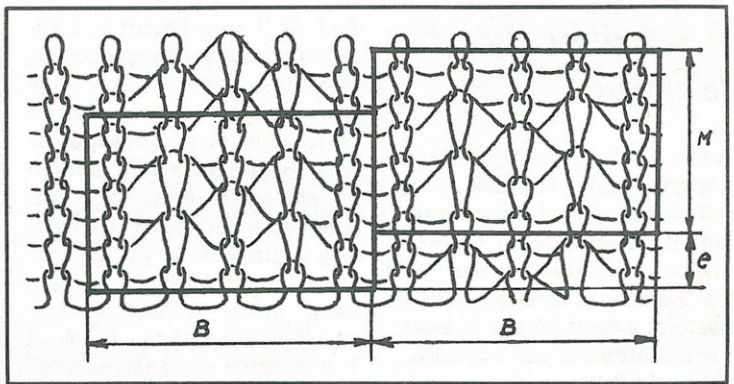
korlatban alkalmazottak három lehetőséget kínálnak:

- 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. esetet 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:

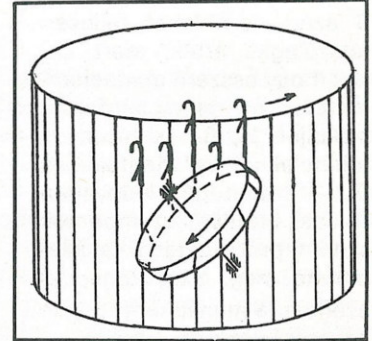
$$N = xK + yB,$$

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

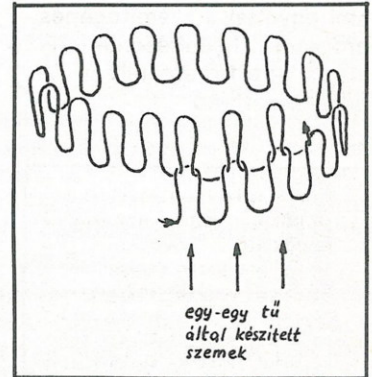
$$e = \frac{cH + 1}{y}$$

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

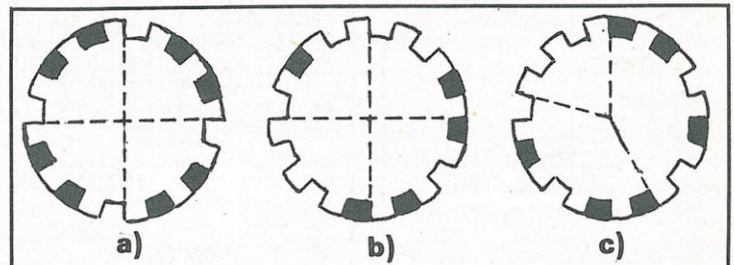
A III. esetben az alkalmazhatóság feltétele, hogy N-nek



2. ábra



3. ábra



4. ábra

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

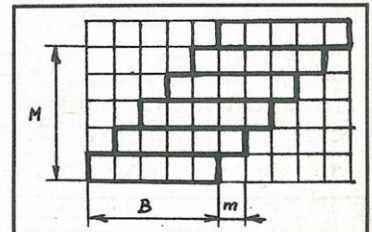
$$N = xK + B_n$$

és emellett fennálljon a

$$K = M_1 B_n + B_m$$

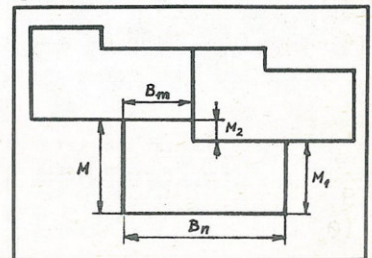
összefüggés is (6. ábra). M_2 ilyenkor 1-gyel egyenlő.

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



5. ábra

6. ábra



24 vonalas be-, kimeneti bővítő

COMMODORE 64

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 — elkövetett 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 amelyek 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.

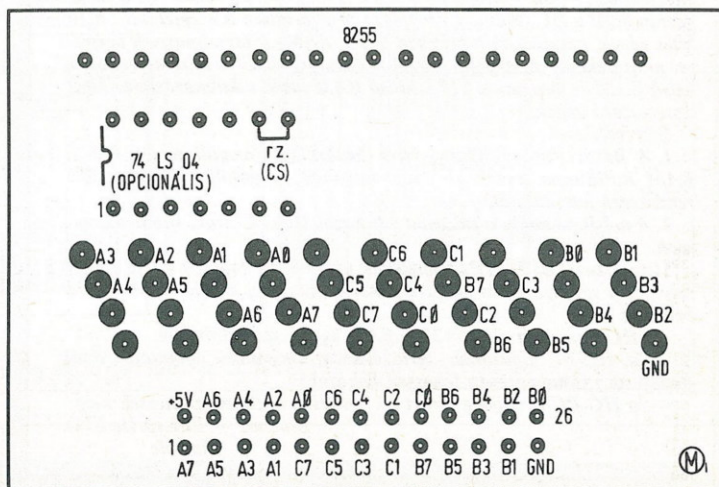
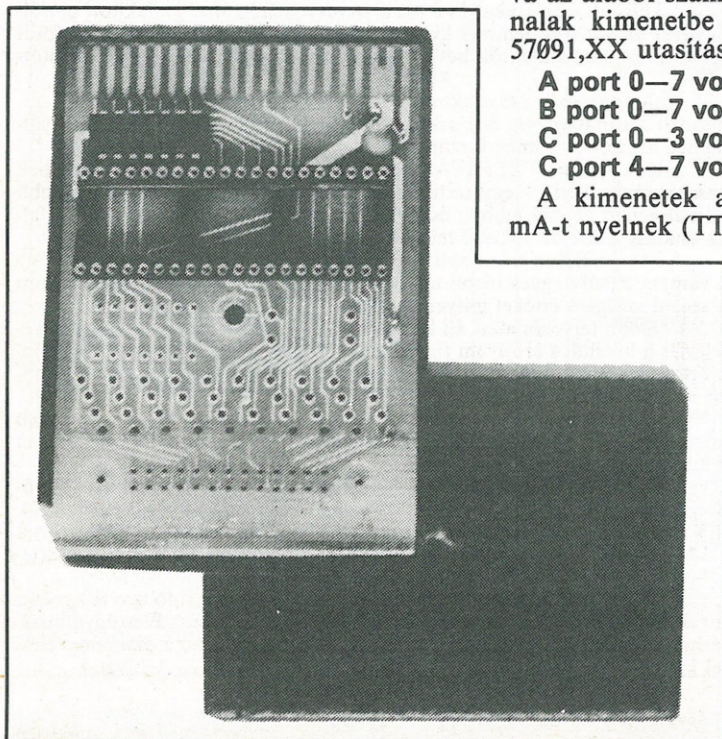
A 8255 a leirtakon kívül számos más üzemmódban is működtethető. Alkalmask többek között gyors, párhuzamos adatátvitelre, intelligens eszközökkel való kapcsolattartásra, hand-shake üzemmódra. Ezek a profi felhasználási módok a 8255 gyári adatlapjain találhatóak.

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

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

A C64-es +5 voltos tápját külső eszközzel legfeljebb 100 mA-rel szabad terhelni. Ebből a bővítőkártja elhasználnál 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 szervet. A különös levélváltásban szívesen működünk közre, mert a papírra vetett gondolatok, javaslatok, vélemények — pro és kontra — az ország számítástechnikai kultúrájára világítanak rá.

Íme az első levél és a hozzá mellékelte felhívás, tiltakozás, melyet harmincan írtak alá:

Tisztelt Kovács Győző!

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 Vámárak 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.

A hazai illetékeseknek (ha ugyan törődött is ezzel illetékes, annál inkább a vámértékkel!) olyan típusok behozatalát kell megkönnyíteni, amik hosszabb távon is alkalmasak lesznek egyénnek, iskolának 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 iradalom van, valamint sokrétű programok léteznek, amik az egységesség (kompatibilitás) miatt is indokolttá teszik e típusok országos elterjedésének támogatását! Ezeket a rendszereket ne sújtsák semmilyen büntetővámossal, mivel arra nincs elfogadható érv!

A miskolci Sinclair—Enterprise Klub tagsága úgy döntött, hogy tiltakozik a HC-PC kategóriájú számítógépeket és tartozékait 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 egyetértésüket fejezzék ki. A Magazin olvasóitól pedig várjuk egyéb észrevételeiket, javaslataikat 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 megjelentek a 16 bites gépek, amelyek minimálisan 512 k RAM-ot tartalmaznak beépítve, mint például az Amiga 500. A 8 bites gépeknél is minimum 256 k belső RAM és maximum 512 k külső RAM kerül alkalmazásra (Z80/B processzorú gépek).

Észrevételeink:

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, vámmok.

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

— HC típusokra előírt 128 k RAM korlátozás eltörlését,
— a HC-PC típusokat, tartozékait (monitor, nyomtató, duál floppy stb.) vámmentesen lehessen behozni,
— a HC-PC típusokra előírt 25 000-es rendeletet szüntessék meg.

Sinclair — Enterprise Klub
Miskolc

És a válasz:

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 áttanulmányoztuk, és az azokkal kapcsolatos álláspontunkról az alábbiakban tájékoztatjuk.

Főként 1988 elejétől — a külföldi utazás szabályainak megváltoztatásával — lehetővé vált a többszöri kiutazás, a családi 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 — valljuk be, külföldön olcsóbb — műszaki cikkekhez. Nem kívánunk foglalkozni a hazai számítógépgyártással, illetve azzal, hogy miért nem gyártanak 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 vámkedvezményt azonban feltételhez kellett kötni, ugyanis tapasztalataink azt mutatják, hogy a külföldről behozott műszaki cikkek és közöttük főként a számítógépek és tartozékai nem mindig annak a személynek a szükségleteit elégítik ki, aki azokat behozta, hanem a felvásárló vállalatok, kisközvetkezők 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 gépeiket, az „iskola” szó csak egyszer szerepel, amiből csak következtetni lehet arra, hogy oktatási célra történő felhasználásra is gondolnak.

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

A külföldi vételárakat és a hazai kereskedelem által kialakított belföldi piaci árakat figyelemmel kísérjük, és a vám alapjául szolgáló belföldi 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 hasznosságát illeti — egyetértünk, az természetes, hogy jobb és nagyobb teljesítményű géppel jobban és többet lehet dolgozni, termelni, illetve felhasználni 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 alapgépekre, illetve a tartozékokra vonatkozó, valamint az egyéb készülékek stb. behozatalakor alkalmazásra kerülő vámkedvezmények mértéke is vizsgálat tárgyát képezi.

Levelükben, illetve a FELHÍVÁS utolsó mondatában, sajnálatos módon, szerepel a „25 000-es” rendelet — ami valójában a 17/1987. (XII. 27.) ÁH 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 haszonzerésre irányuló tevékenységet hivatott a fenti rendelet meggátolni. 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.

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

A Z80 mikroprocesszor utasításelhözéskor mintegy 3, adat olvasáskor 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 hozzávetőlegesen 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 angolra varázsolhatják a gépüket. Azért félig, mert így csak a hibaüzenetek nyelvét és a billentyűzetet alakítják át. A valódi megoldás csak az, ha a német EPROM-ot működésen kívül helyezzük.

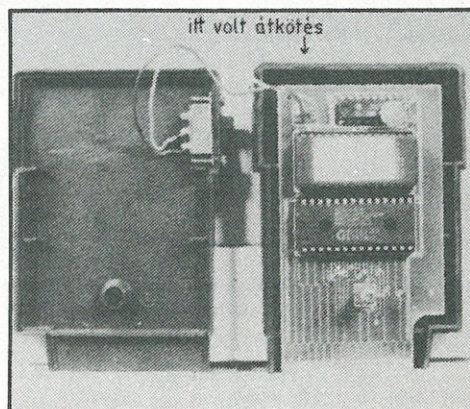
Ha valaki szedett már szét kétnyelvű ROM-kártyát — tényleg csak egy csavar tartja, és a gép garanciáját sem rontja el —, a kártya egyik sarkában (lásd a *fényképen*) egy kis drótdarabot figyelhet meg. Ez a kis drótdarab a megoldás: megszakítva, gépünk angolra válik! A drótvágásnál elegánsabb megoldás 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

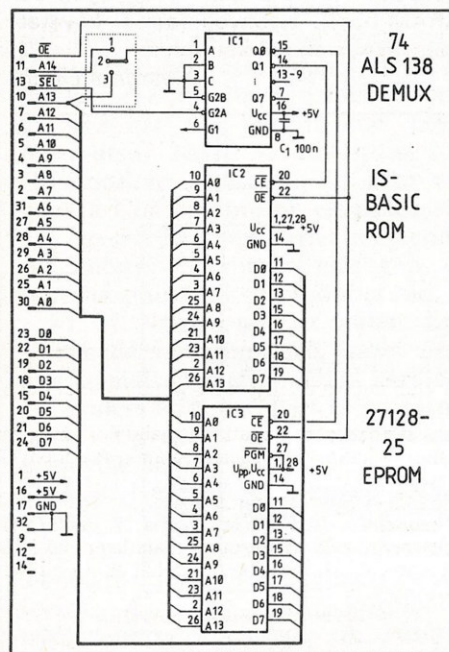
programjainkat, segédprogramjainkat tarthatjuk. Erről részletesebben majd később írok.

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

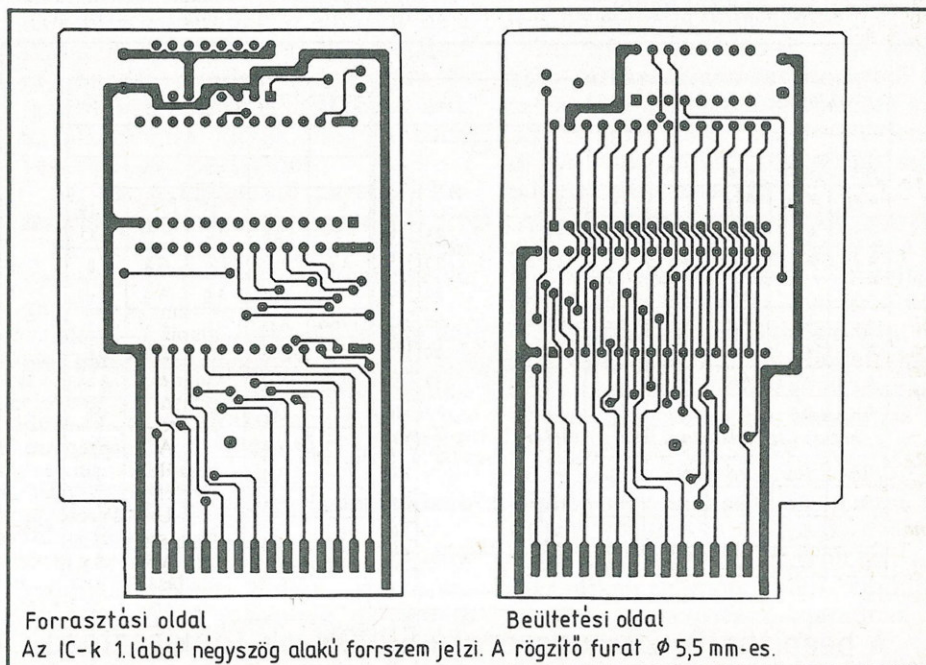
Az eredeti, „gyári” kártyák furatfémzett technológiával készültek. Ez igen bonyolult, anyag- és eszközigenyes, ezenkívül az egészségre káros technoló-



1. ábra



2. ábra



gia (a palládiumvegyületektől sok embernek kihullottak már a fogai). Házilag a furatfémmezést 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 panelelen 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 ónozóban bevonni, így elkerülhetjük a káros oxidációt. Aki nem tud a 32 pólusú csatlakozóra házilag aranyat galvanizálni, próbálkozzon meg néhány hónaponként az újraónozással.

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 érkezett. Ha most az A14 címvezetékén a dekóder „A” bemenetére magas (H=high) szint kerül, akkor az igazságtáblázat szerint a Q1 kimenet kerül az aktív L szintre, így engedélyezi a ROM működését. Ha eközben az OE (output enable=kimenet engedélyezés) jel is aktív L szintű, a ROM az adatbuszra kapuzza a kívánt bájtot.

A	B	A	B	D	C	D	C	-
---	---	---	---	---	---	---	---	---

A kapcsoló 1. helyzete
(kétnyelvű gép)

A	B	A	B	C	C	C	C	-
---	---	---	---	---	---	---	---	---

A kapcsoló 2. helyzete
(angol gép)

A	B	A	B	d	c	d	c	d	c	-
---	---	---	---	---	---	---	---	---	---	---

A kapcsoló 3. helyzete

0. 1. 2. 3. 4. 5. 6. 7. 8.

szegmenscímek

- 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

Ha az A 14-en L szint van, akkor $\overline{Q0}$ kimenet lesz alacsony szintű, így az EPROM kapja meg a \overline{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	L	H	H	H	H	H	H
H	L	L	L	H	L	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	L	L	H	H	H	L	H	H	H	H
H	L	L	H	L	H	H	H	H	L	H	H	H	H
H	L	L	H	H	L	H	H	H	H	L	H	H	H
H	L	L	H	H	H	H	H	H	H	L	H	H	L

* G2A + G2B = H

A drót átvágásával — illetve a kapcsoló nyitásával — a dekóder IC A bemenete a levegőben 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 bemenetére az A13 címvezetékét kapcsoljuk. Ezzel azt érhetjük el, hogy a két memória engedélyezése közötti címkülönbség 18 kb-ot helyett 8 kb-ot lesz. A 27xx sorozatú EPROM-ok és ROM-ok, sőt, az ugyanilyen kapacitású statikus RAM-ok lábkiosztása nagyon hasonló. A két foglalatba két 8 kb-ajtos EPROM-ot dugva, ezek egy 16 kb-ajtos memórialap alsó, illetve felső 8 kb-ajtos 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

27250	27128	2754	2732	2716			2716	2732	2764	27128	27256
Vpp	Vpp	Vpp			1	28			+5V	+5V	+5V
A12	A12	A12			2	27			PGM	PGM	A14
A7	A7	A7	A7	A7	3	26	+5V	+5V	NC	A13	A13
A5	A5	A5	A5	A5	4	25	A8	A8	A8	A8	A8
A5	A5	A5	A5	A5	5	24	A9	A9	A9	A9	A9
A4	A4	A4	A4	A4	6	23	Vpp	A11	A11	A11	A11
A3	A3	A3	A3	A3	7	22	OE	OE/Vpp	OE	OE	OE
A2	A2	A2	A2	A2	8	21	A10	A10	A10	A10	A10
A1	A1	A1	A1	A1	9	20	CE	CE	CE	CE	CE
A0	A0	A0	A0	A0	10	19	Q7	Q7	Q7	Q7	Q7
Q0	Q0	Q0	Q0	Q0	11	18	Q6	Q6	Q6	Q6	Q6
Q1	Q1	Q1	Q1	Q1	12	17	Q5	Q5	Q5	Q5	Q5
Q2	Q2	Q2	Q2	Q2	13	16	Q4	Q4	Q4	Q4	Q4
					14	15	Q3	Q3	Q3	Q3	Q3

A beépíthető 27xx sorozatú EPROM-ok lábkiosztása

↑
billentyűzet

1: +Ucc	32: GND
2: A7	31: A6
3: A8	30: A0
4: A9	29: A3
5: A10	28: A4
6: A11	27: A5
7: A12	26: A2
8: OE	25: A1
9:	24: D7
10: A13	23: D0
11: A14	22: D1
12:	21: D6
13: SEL	20: D5
14:	19: D2
15: D4	18: D3
16: +Ucc	17: GND

A ROM-BAY csatlakozó lábkiosztása kintől a számítógépbe „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 bízunk 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 szétduzzannak a méregetől, 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 embereknek kevesebb pénzük van szórakoztató elektronika vásárlására. Az OTP-hitelakció 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 kb-átra formázható, nem beszélve a gépbe beépített RAM diszkról, ami további 4 x 16 kb-ajt virtuális tárat jelent. Ha valaki komolyan fejleszteni akar, annak ez kitűnő 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.

zött gyakori hiba, hogy nem érzékelik a DOS-t és valami rendellenesség van a felső 64 kb-ajton. 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 reklamált „visszahatás”. Ennek az az oka, hogy ha az RF-jel valamiért visszaáramlik, azt a gép zavarként 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-

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 mentség!

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 árát 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 találnak 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-

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

M. M. Sok panasz érkezett hozzánk, hogy a programkasszettekhez 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

Fedezzük fel együtt!

Dőlnek a dominók

Az előző két részben elsősorban az Enterprise színeivel foglalkoztunk. A színek számát a GRAPHICS utasítással adtuk meg, a színeket a SET PALETTE utasítással jelöltük ki, akkor, ha kettő, négy és 16 színnel dolgoztunk. A keret színét a SET BORDER, a „papír” színét a SET PAPER, a „ceruza” színét a SET INK utasítással állíthatjuk be. A SET PALETTE utasítással megadott színek közül egy a SET COLOUR utasítással változtatható.

A színezést ezek után az olvasóra bizzuk. Most a grafikával foglalkozunk egy kicsit bővebben, miközben megismerkedhetünk a legalapvetőbb BASIC utasításokkal és programozási alapfogalmakkal, valamint a videolapokkal.

Rajzoljunk egy téglalapot! Először numerikus konstansokkal — számokkal — adjuk meg a csúcspontok koordinátáit (11. lista). Jelöljük a téglalap bal felső pontjának koordinátáit X,Y-nal, és az oldalak hosszát legyen A és B (12. lista). Az X,Y,A és B változók értékeit az INPUT PROMPT utasítással kérjük, amivel szöveg is kiíratható.

A 150-es sorban változtatjuk a X és Y változók értékeit, és a 160-as sorról a GOTO utasítással visszaugrunk a 140-es sorra. Ebből a végtelen ciklusból hibajelzéssel lép ki a programunk.

Az IF utasítással vizsgálhatjuk meg, hogy mikor kerül le a téglalap jobb felső pontja a képernyőről az X és Y változók értékének növelésekor.

```
160 IF X + A > 1279 THEN STOP
170 IF Y + B < 720 THEN 140
Fessük be a téglalapot!
```

Akkor, ha a téglalapot ferdén szeretnénk a képernyőre rajzolni, egyszerűbb a teknőcutasításokat használni (F—81. old.)

A programban (13. lista) az OPTION ANGLE DEGREES utasítás hatására az ANGLE, LEFT, illetve a RIGHT utasításokban szögekben adhatjuk meg az elfordulás mértékét. Az ANGLE utasítással a képernyő jobb oldala felé mutató irányhoz képest forgathatjuk a téglalapot, a LEFT és RIGHT utasításokkal pedig a „ceruza” pillanatnyi irányához adódik a szög. A 150-es sorban alaphelyzetbe állítottuk a „ceruzát”.

Egy kis matematikai ismerettel most is megoldható, hogy a „ceruza” a téglalap középpontjába kerüljön (170-es sor). A 12. lista szerinti programban a festés a

```
PLOT X + A/2, Y + B/2, PAINT
utasítással oldható meg.
```

Szimuláljuk most egy dominó eldőlését. A téglalapot, azaz a „dominót” többször — csökkenő dőlési szögekkel — kirajzoljuk, majd „eltüntetjük”. Az eltüntetés úgy történik, hogy a téglalapot a „papír” színnel rajzoljuk meg (14. lista). Láthatóan a szimuláció nem a legtekélyesebb.

Nézzünk egy másik megoldást (15. lista). A 16 színű képernyő hat színét a SET PALETTE utasítással a „papír” színére, azaz feketére állítjuk. A dőlés fázisait ezekkel a színekkel megrajzoljuk (160—190-es sorok), majd a SET COLOUR utasítással egy kis időre láthatóvá tesszük az eddig nem látott fázisokat (210—260-as sorok). A 230—240-es sorokban várakozó ciklust írtunk, amivel lassítjuk a „dominó” dőlését.

Ez sem az igaz! A láthatatlanul megrajzolt téglalapok törlik az első téglalap vonalait. Ezen úgy segítünk, hogy a fázisrajzokat egy másik lapon készítjük el, és ezeket jelenítjük meg (16. lista).

11. lista

```
100 REM ---11. program ---
110 GRAPHICS HIRES 2
120 PLOT 200,200;500,200;500,400;
130 PLOT 200,400;200,200
140 PLOT 350,300,PAINT
```

12. lista

```
100 REM ---12. program ---
110 GRAPHICS HIRES 2
120 INPUT PROMPT "csucspont=":X,Y
130 INPUT PROMPT "oldalak=":A,B
140 PLOT X,Y;X+A,Y;X+A,Y+B;X,Y;X,Y
150 PLOT X+A/2,Y+B/2,PAINT
160 IF X+A>1279 THEN STOP:GOTO 140
170 IF Y+B<720 THEN 140
```

13. lista

```
100 REM ---13. program ---
110 OPTION ANGLE DEGREES
120 GRAPHICS HIRES 2
130 LET X=400:LET Y=300:LET A=300:LET B=
100
140 INPUT PROMPT "elfordogas szoge=":FI
150 PLOT X,Y:ANGLE FI:FORWARD A:ANGLE 90
+FI:FORWARD B:ANGLE 180+FI:FORWARD A:ANGLE
270+FI:FORWARD B
160 LET X=X+400:PLOT ANGLE 0
170 PLOT X,Y:LEFT FI:FORWARD A:LEFT 90:F
ORWARD B:LEFT 90:FORWARD A:LEFT 90:FORWARD
B
180 PLOT X,Y:ANGLE FI:ATN(B/A)/2:FORWARD
SOR((A+B)/2),PAINT
```

16. lista

```
100 REM ---16. program ---
110 OPTION ANGLE DEGREES
120 GRAPHICS HIRES 16
130 SET PALETTE 0,2,0,0,0,0,0,0
140 LET X=600:LET Y=100:LET A=40:LET B=3
00:LET T=1
145 PLOT X,Y:ANGLE 90:FORWARD B:ANGLE 18
0:FORWARD A:ANGLE 270:FORWARD B:ANGLE 360:
FORWARD A
150 PRINT "Varj egy kicsit!"
152 SET VIDEO MODE 1
154 SET COLOUR 2
156 OPEN #1:"video:"
158 SET #1:PALETTE 0,2,0,0,0,0,0,0
160 FOR FI=90 TO 0 STEP-15
170 SET #1:INK T:LET T=T+1
180 PLOT #1:X,Y:ANGLE FI:FORWARD B:ANG
LE 90+FI:FORWARD A:ANGLE 180+FI:FORWARD B:
ANGLE 270+FI:FORWARD A
190 NEXT
200 CLEAR TEXT
205 DISPLAY #1:AT 1 FROM 1 TO 20
210 FOR T=1 TO 6
220 SET #1:COLOUR T,2
230 FOR I=1 TO 20
240 NEXT
250 SET #1:COLOUR T,0
260 NEXT
270 SET #1:COLOUR 7,2
```

14. lista

```
100 REM ---14. program ---
110 OPTION ANGLE DEGREES
120 GRAPHICS HIRES 2
130 LET X=600:LET Y=100:LET A=40:LET B=3
00
140 FOR FI=90 TO 0 STEP-15
150 SET INK 1
160 PLOT X,Y:ANGLE FI:FORWARD B:ANGLE
90+FI:FORWARD A:ANGLE 180+FI:FORWARD B:ANG
LE 270+FI:FORWARD A
170 IF FI=0 THEN EXIT FOR
180 SET INK 0
190 PLOT X,Y:ANGLE FI:FORWARD B:ANGLE
90+FI:FORWARD A:ANGLE 180+FI:FORWARD B:ANG
LE 270+FI:FORWARD A
200 NEXT
```

17. lista

```
100 REM ---17. program ---
110 OPTION ANGLE DEGREES
120 GRAPHICS HIRES 16
140 LET X=200:LET Y=0:LET A=40:LET B=3
00:LET T=1
143 SET INK 7
145 PLOT X,Y;+736:ANGLE 90:FORWARD B:ANG
LE 180:FORWARD A:ANGLE 270:FORWARD B:ANG
LE 0:FORWARD A
146 PLOT ANGLE 90+ATN(A/B),FORWARD SOR(A
+736)/2,PAINT
150 PRINT "Varj egy kicsit!"
152 SET VIDEO MODE 1
154 SET VIDEO COLOUR 2
156 SET VIDEO X 24
158 SET VIDEO Y 9
160 FOR FI=90 TO 0 STEP-15
162 I OPEN #1:"video:"
164 SET #1:INK 7
180 PLOT #1:X-732,Y:ANGLE FI:FORWARD
B:ANGLE 90+FI:FORWARD A:ANGLE 180+FI:FORWA
RD B:ANGLE 270+FI:FORWARD A
182 PLOT #1:ANGLE FI+ATN(A/B),FORWARD
SOR(A+736)/2,PAINT
185 LET T=T+1
190 NEXT
200 CLEAR TEXT
210 FOR T=1 TO 7
220 DISPLAY #1:AT 5 FROM 1 TO 9
230 FOR I=1 TO 10
240 NEXT
260 NEXT
```

15. lista

```
100 REM ---15. program ---
110 OPTION ANGLE DEGREES
120 GRAPHICS HIRES 16
130 SET PALETTE 0,2,0,0,0,0,0,0
140 LET X=600:LET Y=100:LET A=40:LET B=3
00:LET T=1
145 PRINT "Varj egy kicsit!"
160 FOR FI=90 TO 0 STEP-15
170 SET INK T:LET T=T+1
180 PLOT X,Y:ANGLE FI:FORWARD B:ANGLE
90+FI:FORWARD A:ANGLE 180+FI:FORWARD B:ANG
LE 270+FI:FORWARD A
190 NEXT
200 CLEAR TEXT
210 FOR T=1 TO 6
220 SET COLOUR T,2
230 FOR I=1 TO 20
240 NEXT
250 SET COLOUR T,0
260 NEXT
270 SET COLOUR 7,2
```

18. lista

```
10 REM ---18. program ---
100 REM --- szinek keverese ---
110 OPTION ANGLE DEGREES
120 TEXT
130 SET VIDEO MODE 1
140 SET VIDEO COLOUR 2
150 SET VIDEO X 10
160 SET VIDEO Y 5
170 FOR J=1 TO 36
180 OPEN #J:"video:"
190 NEXT
200 FOR J=1 TO 36
210 SET #J:PALETTE BLACK,RED,GREEN,BLU
E
220 PLOT #J:160,90
230 PLOT #J:ELLIPSE 60,60
240 FOR I=1 TO 3
250 PLOT #J:160,90
260 PLOT #J:ANGLE I+120+J*250-360+IN
T(J*190/360);
270 PLOT #J:FORWARD 60
280 NEXT
290 FOR I=1 TO 3
300 SET #J:INK I
310 PLOT #J:160,90
320 PLOT #J:ANGLE I+120+J*250-360+IN
T(J*190/360)+60
330 PLOT #J:FORWARD 40,
340 PLOT #J:PAINT
350 NEXT
355 TEXT #PRINT J ".lap"
360 DISPLAY #J:AT 3 FROM 1 TO 5
370 NEXT
380 REM --- megjelenites ---
385 CLEAR TEXT
390 FOR J=1 TO 36
400 DISPLAY #J:AT 12 FROM 1 TO 5
410 NEXT
420 GET Q#
430 IF Q#="" THEN 390
```

20. lista

```
1 REM --- FORGÓ KOCKAK ---
2 REM Nagy Akos Miskolc FFB 20/N Pr
3 !
100 REM ---VIDEO LAPOK NYITASA---
110 TEXT
120 SET VIDEO MODE 1
130 SET VIDEO COLOUR 1
140 SET VIDEO X 19
150 SET VIDEO Y 11
160 FOR B=1 TO 14
170 OPEN #B:"VIDEO:"
180 NEXT
190 REM ---NAGY KOCKAK RAJZOLASA---
200 LET W=2560:LET L=R=100
210 LET D=2500:LET K=160:LET A=0
220 CALL RAJZOLAS
230 REM ---KIS KOCKAK RAJZOLASA---
240 LET W=4900:LET R=150
250 LET D=1200:LET K=70:LET A=1
260 CALL RAJZOLAS
270 REM ---FORGATAS---
280 FOR T=1 TO 14
290 DISPLAY #T:AT 1 FROM 1 TO 11
300 DISPLAY #T:(15-T):AT 12 FROM 1 TO 11
310 NEXT
320 SET Q#
330 IF Q#="" THEN 280
340 !
350 REM --- RAJZOLAS ---
360 DEF RAJZOLAS
370 DIM I(4),C(4),Z(4)
380 FOR L=1 TO 14
390 IF A=1 THEN
400 LET M=15-L
410 SET #M:INK YELLOW
420 ELSE
430 LET M=L
440 SET #M:INK CYAN
450 END IF
460 DISPLAY #M:AT 12 FROM 1 TO 11
470 LET I(2)=(L-1)*(K/13.5)
480 LET I(1)=1-(K-I(2))
490 LET I(4)=1-I(2)
500 LET I(3)=K-I(2)
510 FOR P=1 TO 4
520 LET Z(P)=300-I(P)
530 LET C(P)=R+(1+2*(P-2))*SOR(I
*(1-(1-(P-2)/4)))
540 PLOT #M:Z(P),C(P);Z(P),C(P)+K
550 NEXT
560 PLOT #M:Z(1),C(1);Z(2),C(2);LIST
1-600
570 PLOT #M:Z(3),C(3);Z(4),C(4);Z(1)
,C(1)
580 PLOT #M:Z(1),C(1)+K;Z(2),LIST1-3
0,C(2)+K;Z(3),C(3)+K;Z(4),C(4)+K;Z(1),C(1)
+K
590 DISPLAY #M:AT 1 FROM 1 TO 11
600 NEXT
610 END DEF
```

19. lista

```
1 REM ---INTEGETO EMBER---
2 REM NAGY AKOS - Miskolc 19/Nagy
3 REM Foldes Ferenc Gimaizium
4 REM
110 SET VIDEO COLOUR 1
120 SET VIDEO X 19
130 SET VIDEO Y 11
140 FOR B=1 TO 2
150 OPEN #B:"VIDEO:"
160 SET #B:INK YELLOW
170 PLOT #B:300,320,ELLIPSE 30,30,PAINT
180 PLOT #B:300,290;300,140
190 PLOT #B:300,140;300,40
200 PLOT #B:300,140;220,40
210 NEXT
220 PLOT #1:200,300;300,240;400,300
230 PLOT #2:400,180;300,240;200,180
240 NEXT
240 TEXT
250 FOR T=1 TO 2
260 DISPLAY #T:AT 1 FROM 1 TO 11
270 WAIT DELAY 1
280 NEXT
290 GET Q#
300 IF Q#="" THEN 250
```

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.

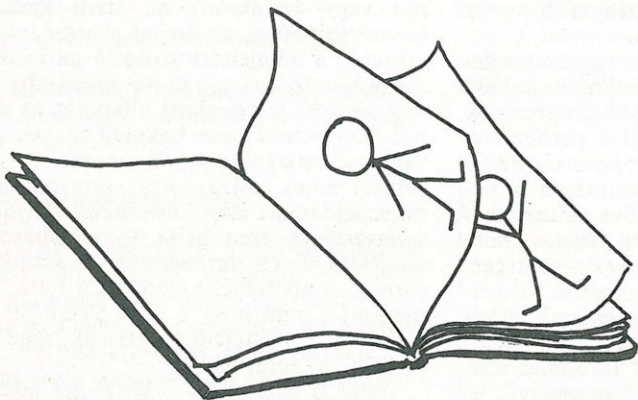
205-ös sor: megjelenítjük az 1. lapot.

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

Tovább javítjuk a programot. Ezúttal minden fázist 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 lapméret, ha kiadjuk a SET BORDER 2 parancsot.

A „dominó” elhelyezéséhez tudnunk kell, hogy egy karakter 32×36 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 körcírcikre osztunk, és a körcírciket befestjük a három alapszínnel. Ezután a körlapot megforgatjuk. A körlap fehér színűnek látszik. Futtassuk le a programot, majd változtassuk a színek arányát és összetételét! 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 leírt játékot úgy valósíthatjuk meg a számítógéppel, hogy a papírlapok helyett videolapokat használunk, és arra rajzoljuk a mozgás fázisait. A legegyszerűbben például (19. lista) integető emberkét 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 tessenek.

Dusza Árpád

Mi a manó?

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

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

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

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

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

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

— Szükség szerint fel-le szereli a csatlakozókártyát, várva, hogy a csatlakozó meddig bírja.

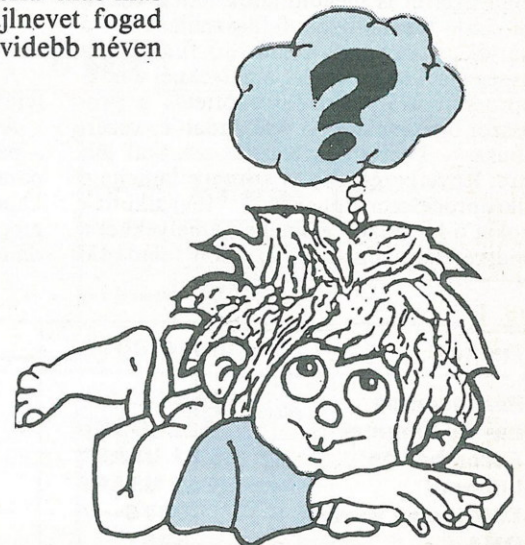
— Addig megy, amíg a lemezre átírt programokat nem tudja lemásolni.

— Megfogadja a Novotrade Rt. eddig ki nem mondott tanácsát: aki annyira nem ért a számítógéphez, hogy egy nyavalyás játékprogramot 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 panaszok. Wágner Miklós budapesti olvasónk többek között a következőket panaszolja. A Spectrum-emulátort és a lemez meghajtót nem lehet egyszerre használni, így választás elé kerül, hogy melyiket csatlakoztassa. Hasonló választás 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 lemezről akar tovább tölteni, és mivel ez nem sikerül neki, leáll. Ha a programot átmásolja lemezre, a betöltésnél az első fájl után szintén leáll a gép, mivel a periféria más-más hosszúságú fájlnevet fogad el, és így a rövidebb néven

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



Gaetsch Günterné rajza

Hardver

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

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 végrehajtó 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 egyes billentyűk 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-időzítési problémák is felmerültek.

Igen sok, külvilággal kapcsolatos feladat hasonló volt, ezek azonban önállóan, kis céhardverrel 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 rendszerelem közötti és az adott feladattal

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

A fejlődés során megjelenő újabb perifériavezérlők már programozhatók voltak. Ezáltal a processzorban lévő programmal olyan kódokat adhatunk ki a perifériavezérlő 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 kismértékben eltérő feladatra is felhasználhatóvá vált. Ezeknél a processzor saját cím-, adat- és vezérlőjelei segítségével információcsere-t folytat a perifériavezérlő 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 perifériavezérlő működésének módját és paramétereit szabja meg a processzortól kapott kódoknak — parancsbájtoknak — 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 átvitt 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 záró 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, bájtokban 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 perifériavezérlőket az úgynevezett egytokos 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
Lemez meghajtó vezérlő	8271	—	MC6843

EGYTOKOS MIKROSZÁMÍTÓGÉPEK

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

Általánosan fogalmazva: az egytokos mikroszámítógép a számítógépes részek egy tokba való integrálását jelenti. Ezek a részek a következők.

Csak olvasható memória (ROM vagy PROM, EPROM)

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

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 alkalmazott központi egységgel. A CPU legtöbbször képes be/kimeneti vonalak és a memória egyes biteinek tesztelésére, beállítására és törlésére, mivel az alkalmazások során sokszor szükség van egy-egy kimeneti vonal állapotának megváltoztatására vagy egy-egy bemeneti vonal beolvasására. Ezek általában a külső, kétállapotú készülékekhez, kapcsolókhoz, termosztátokhoz, szilárdtest-relékhez, 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 alkalmasak 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 vezérlőregiszterébe írásával. A megfelelő formátumú soros adat előállítását ezek után a hardver áramkör végzi és kezeli.

Időzítő/számláló egységek

Az egytokos mikroszámítógépek alkalmazása sok esetben valós idejű (real time) végrehajtást kíván. Ez megoldható a végrehajtási idők 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 jelzõtbitet 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 eseményszámlálóként használják, amihez egy külön be-

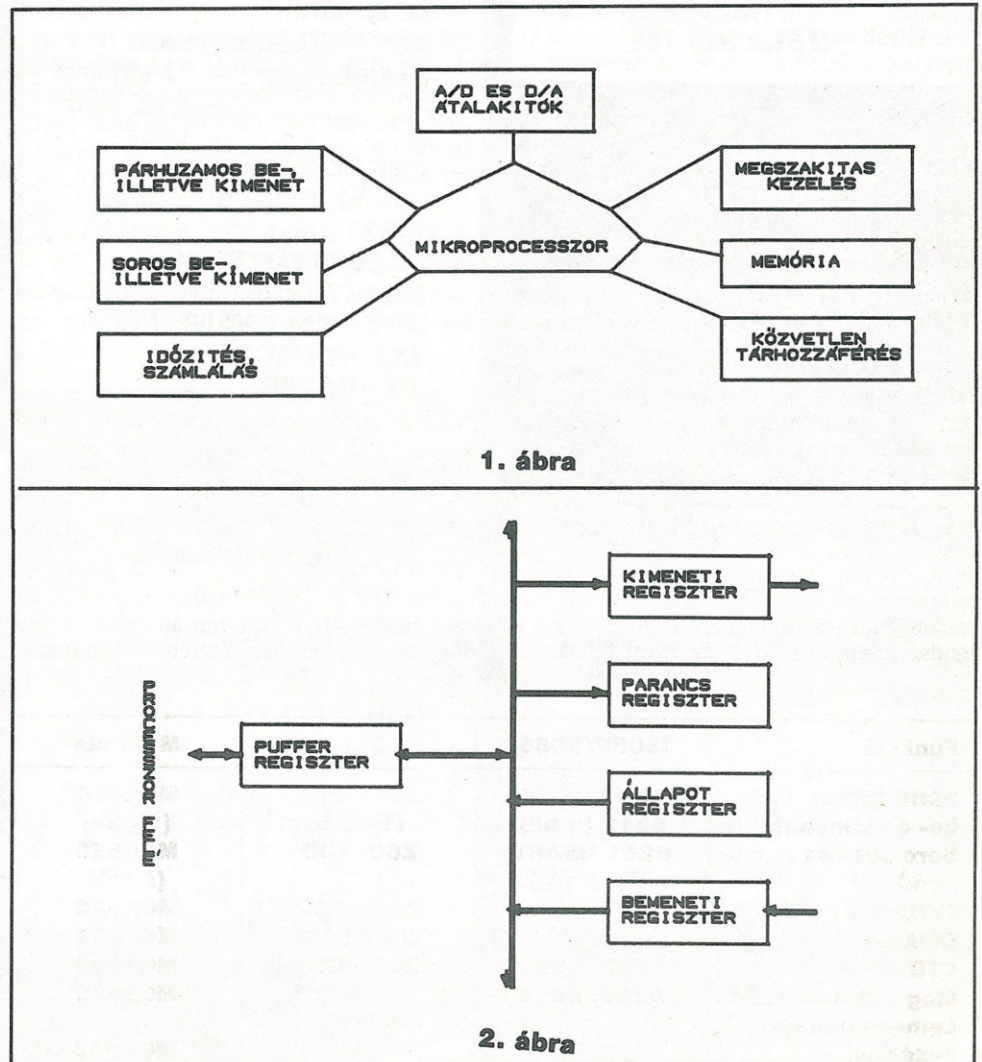
menet kell, és a használat a külső esemény létrejötte lépteti.

Az időzítést beállító elemek

A legtöbb egytokos 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 generációnak tekinthetők. Segítségükkel az új, modern felépítésű mikroprocesszoros rendszerekben megjelenő feladatok oldhatók meg, amelyek közül most csupán kettőről, mintegy illusztrációként teszünk említést.

Az új típusú, nagy — millió bájtt — címtartományú mikroprocesszorok és a korszerű operációs rendszerek alkalmazásához elengedhetetlen a nagy számítógépeknél 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 sínrendszer osztott idejű felhasználásából adódó konfliktushelyzeteket oldja meg a BAU, a Bus Arbiter Unit, vagyis a sínvezérlő egység.

Dr. Kónya László



BÖRZE



COBRA
ELEKTRONIKAI ÉS SZOLGÁLTATÓ KISSZÖVETKEZET
1097 Budapest, IX. Ilatos út 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
	114 600,— 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 lízingelhetők 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

- **BILLENTYŰZETEK**
- **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

SZEGED
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
ÍRÓGÉP**

Bp. VI.,

Népköztársaság útja 2.



1146 Bp., AJTÓSI DÜRER
SOR 10.

Levél cím:
1393 Pf.: 319.
Telefon: 421-974
Telex: 22-6544

február havi kínálata

a 80 386-os
mikroproceszorra
alapozott, multiuser
üzemmódú

ACER SYS 32

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

Merre tart

a világ?

10th Annual
Volume 6 Number 9

COMDEX

SHOW DAILY

4

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, olvasó, 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 tárgyalt — 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 Distributi-on EXposition — Számítógépterjesztők Kiállítása) sorozat 1979-ben egyetlen szállodában kezdődött összesen

A tizedik Las Vegas-i COMDEX mindenben felülmúlta az eddigieket. (1700 kiállító, 110 000 látogató, 8 kiállítási csarnok és szálloda, 85 ország, 52 szekció, 1000 kiállítási dolgozó, 1600 szállítóeszköz, 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. — Érdekeséggé tegyem 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 szállodát, amelynek telkén fel akarja

1. kép





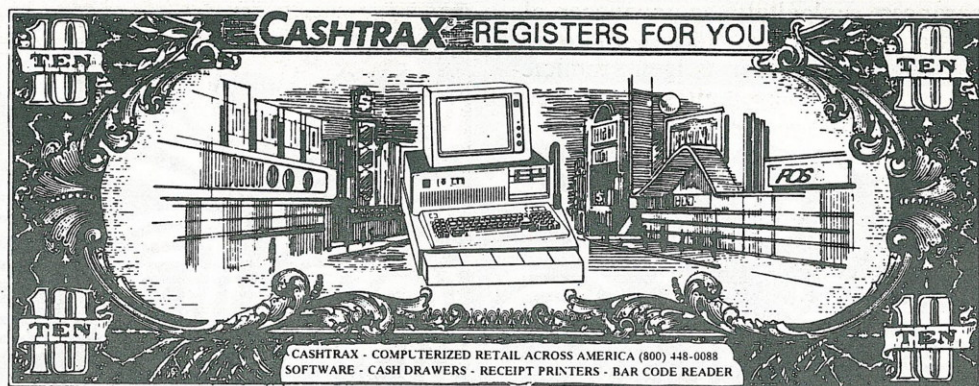
2. kép

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. ábrán 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? Előszörban 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 például rendkívül alacsonyok. 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.

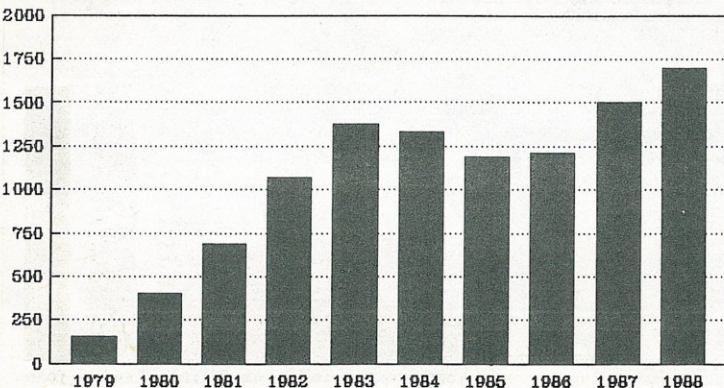
S a kiállítás? — A hátsó borító utolsó képén és e cikk 1-2. képén — melyek a kiállítás két fő helyszínén készültek — látható, hogy minden nagy cég megjelent. (A két fekete-fehér képen az is, hogy a felkapottabb helyeken nagy volt a tömeg.) A belső borító

első két képén látható, hogy milyen minőségű képeket lehet előállítani a NEC Multisync 2A és MACSync monitorain. (Az elsőt egyébként a Las Vegas környéki Spring Mountain — Tavasz Hegység látható.) Ugyanezen oldal alsó két képén 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-plotternyelveit. 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.

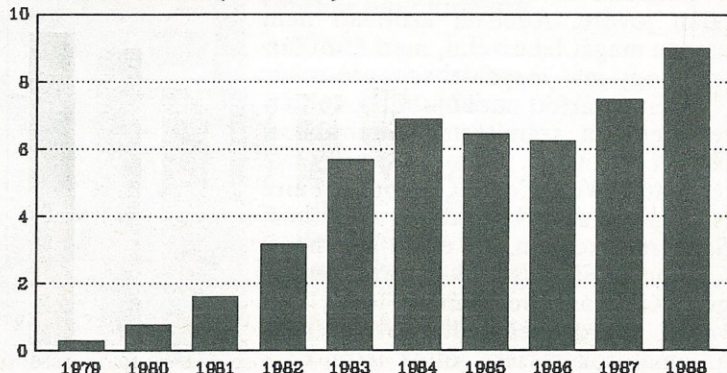
A kiállítás és a konferencia öt napig tartott, de ez az idő csak a kiállítás futó áttekintésére és egy-egy érdekesebb rész alaposabb megnézésére volt elég. A konferenciára, bármilyen csábító külsejű robot csalogatott (hátsó borító belső oldal legfelső kép), alig maradt idő.

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)



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

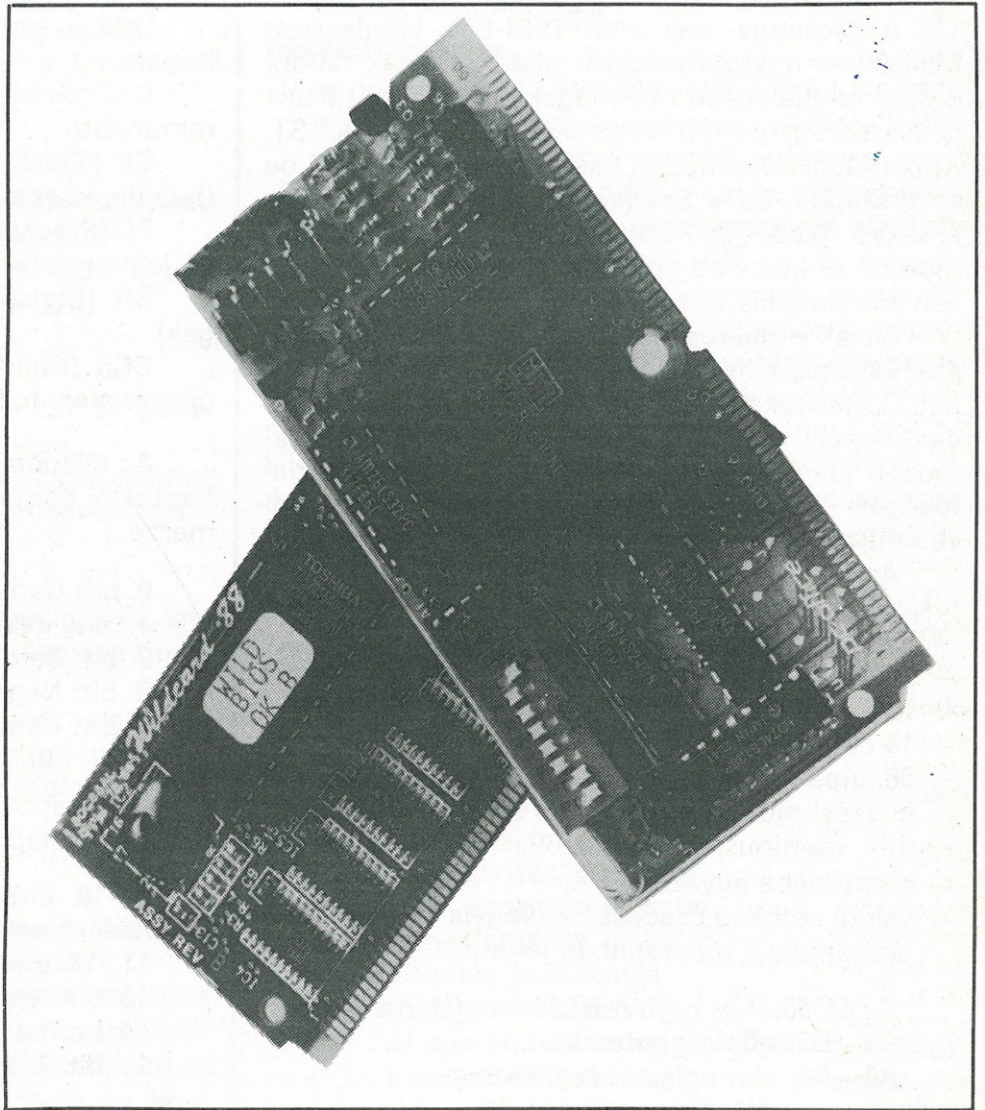
Az IBM 625, más cég által a PS/2-höz tervezett eszközzel tett említést. Legjelentősebb ezek közül a Hewlett-Packard és az AT&T által kiállított OS/2 LAN Manager. Ez azt jelenti, hogy az IBM engedélyezte ezt a tevékenységet. A COMDEX Show Daily erre a következő kérdést tette fel: Ez a gláznoszty nemsokára peresztrójkát 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.

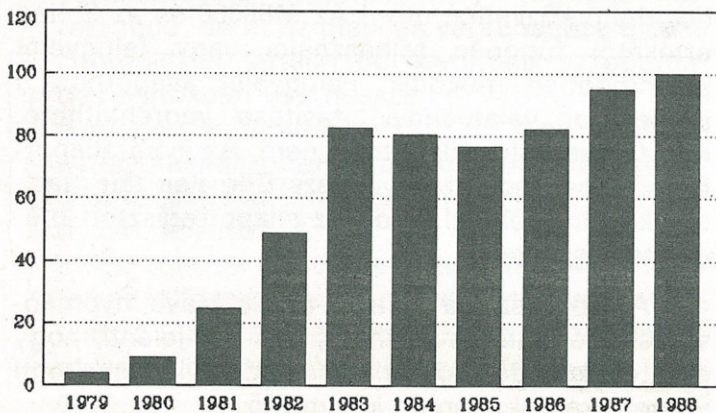
Simonyi Endre és Simonyi Márton

4. kép

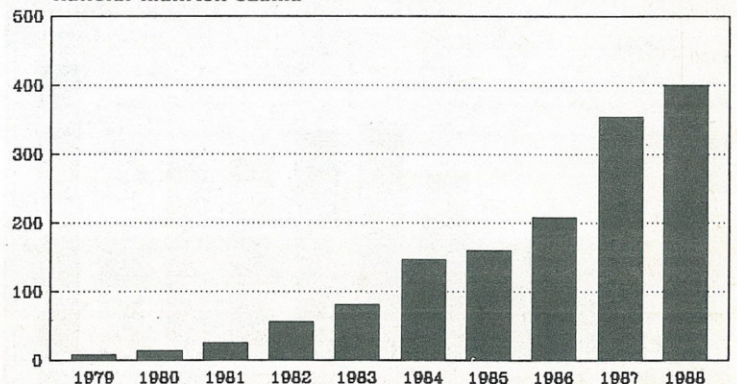


5. kép

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 M68008-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: utasításszámláló)

SR (Status Register): állapotregiszterek (flaggek)

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

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

0. bit: Carry (C) — átvitelkapcsoló

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

2. bit: Zero (Z) — zérókapcsoló

3. bit: Negative (N) — negatívkapcsoló

4. bit: extension (X) — bővítkapcsoló

5—7. bitek: nem használt (értékük 0)

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

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

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

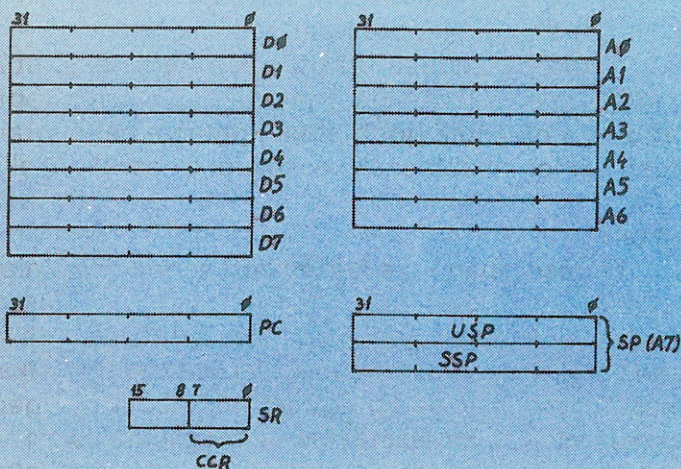
13. bit: Supervisor (S) — felügyelői állapot

14. bit: nem használt (értéke 0)

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

A rendszerbajt 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 befejezőként 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 programszámláló 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.

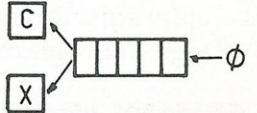
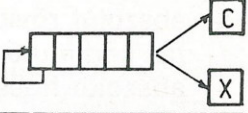
A táblázat jelöléseinek értelmezése:

"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

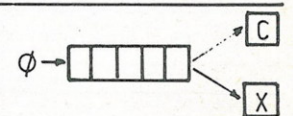
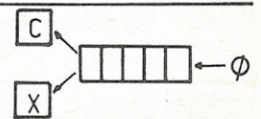
A kétoperandusú utasítások mnemonikja után mindig az első (bal oldali) operandus a forrás operandus, a második (jobb oldali) pedig a cél operandus! Ezt leginkább azok vessék jól az eszükbe, akik eddig Intel vagy Zilog mikroprocesszorokat használtak.

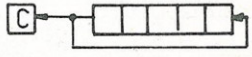
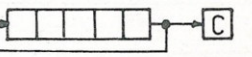
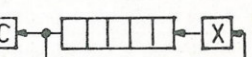
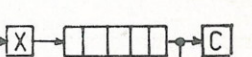
Leírásomban — a korlátozott terjedelem miatt — nem adhattam részletes információkat. Céлом csupán az volt, hogy a kezdőknek felvázoljam a 16 bites mikroprocesszor igen jó képességeit, az alkalmazók munkáját pedig egy jól használható segéd-eszközzel — a táblázatokkal — megkönnyítsem.


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) $NV + \bar{N}\bar{V}$ LT (less than) $N\bar{V} + \bar{N}V$ GT (greater than) $N\bar{V}\bar{Z} + \bar{N}VZ$ LE (less or equal) $Z + N\bar{V} + \bar{N}V$
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 törlendő 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	-	\updownarrow	\updownarrow	\updownarrow	\updownarrow	Kivonást végez, de nem írja felül az operandusokat.
CMPA.X "ea", Ax	CoMPare Addressreg.	W, L	-	\updownarrow	\updownarrow	\updownarrow	\updownarrow	CMP az egyik címregiszterrel.
CMPI.X # "adat", "ea"	CoMPare Immediate	B, W, L	-	\updownarrow	\updownarrow	\updownarrow	\updownarrow	CMP egy közvetlen adattal.
CMPM.X (Ax)+, (Ay)+	CoMPare in Memory	B, W, L	-	\updownarrow	\updownarrow	\updownarrow	\updownarrow	CMP a memóriában lévő operandusok között.
cpGEN "paraméter"	CoProc. GENeral Comm.	?	?	?	?	?	?	Vezérlésátadás társprocesszornak.
DBcc Dx, "címke"	Decrement and Branch if Condition Codes are true	W	-	-	-	-	-	Relatív ugrás, ha a feltétel teljesül és Dx a dekrementálás után $\neq -1$ (Lásd még: Bcc)
DIVS "ea", Dx	DIVide Signed	W	-	\updownarrow	\updownarrow	\updownarrow	ϕ	$Dx = \frac{Dx (L)}{"ea" (W)}$ Előjeles osztás. Ha "ea" = ϕ , akkor \rightarrow TRAP# ϕ S Ha "ea" > Dx, akkor \rightarrow V = 1
DIVU "ea", Dx	DIVide Unsigned	W	-	\updownarrow	\updownarrow	\updownarrow	ϕ	Ugyanaz, mint DIVS-nél, de ez előjel nélküli osztás.
EOR.X Dx, "ea"	logical Exclusive OR	B, W, L	-	\updownarrow	\updownarrow	ϕ	ϕ	Logikai KIZÁRÓ-VAGY. (EOR, XOR)
EORI.X # "adat", "ea"	logical EOR with Immed	B, W, L	-	\updownarrow	\updownarrow	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	-	\updownarrow	\updownarrow	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 \rightarrow (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 \rightarrow -(SP) SP \rightarrow Ax SP + Localstack \rightarrow SP
LSL.X Dx, Dy LSL.X # "adat", Dx LSL.X "ea"	Logical Shift Left	B, W, L	\updownarrow	\updownarrow	\updownarrow	ϕ	\updownarrow	Dy-t Dx helyel Dx-et "adat" helyel "ea"-t 1 helyel
LSR.X Dx, Dy LSR.X # "adat", Dx LSR.X "ea"	Logical Shift Right	B, W, L	\updownarrow	\updownarrow	\updownarrow	ϕ	\updownarrow	Dy-t Dx helyel Dx-et "adat" helyel "ea"-t 1 helyel
MOVE.X "ea", "ea"	MOVE source to dest.	B, W, L	-	\updownarrow	\updownarrow	ϕ	ϕ	Adatmozgatás egyik címről a másikra.
MOVE "ea", CCR	MOVE source to CCR	W	\updownarrow	\updownarrow	\updownarrow	\updownarrow	\updownarrow	SR-nek csak az alsó byte-ját (CCR) állítja (a felső byte nem változik).
MOVE "ea", SR	MOVE source to SR	W	\updownarrow	\updownarrow	\updownarrow	\updownarrow	\updownarrow	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.lista", "ea" MOVEM.X "ea", "reg.lista"	MOVE Multiple registers	W, L	-	-	-	-	-	W esetén minden regiszter előjelhelyesen kibővül 32 bitre (L formátumra).
MOVEP Dx, ofs. (Ax) MOVEP ofs. (Ax), Dx	MOVE to or from Peripherals	W, L	-	-	-	-	-	
MOVEQ # "adat", Dx	MOVE immediate Quick	L	-	↑	↑	∅	∅	Az "adat" max. 8 bites lehet. Közvetlen adat gyors töltése.
MULS "ea", Dx	MULTiply with Sign	W	-	↑	↑	∅	∅	Előjeles szorzás.
MULU "ea", Dx	MULTiply without Sign	W	-	↑	↑	∅	∅	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 komplement képzése.
NEG.X "ea"	NEGate with eXtend	B, W, L	↑	↑	↑	↑	↑	Elsőként elvégez egy nullára vonatkozó vizsgálatot is!
NOP	No OPERATION	-	-	-	-	-	-	PC + 2 → PC
NOT.X "ea"	logical NOT	B, W, L	-	↑	↑	∅	∅	1-es komplement képzése.
OR.X Dx, "ea" OR.X "ea", Dx	logical OR	B, W, L	-	↑	↑	∅	∅	Logikai VAGY.
ORI.X # "adat", "ea"	logical OR with Immed.	B, W, L	-	↑	↑	∅	∅	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	-	↑	↑	∅	↑	Dy-t Dx helyel Dx-et "data" helyel "ea"-t 1 helyel 
ROR.X Dx, Dy ROR.X # "adat", Dx ROR.X "ea"	ROtate Right	B, W, L	-	↑	↑	∅	↑	Dy-t Dx helyel Dx-et "adat" helyel "ea"-t 1 helyel 
ROXL.X Dx, Dy ROXL.X # "adat", Dx ROXL.X "ea"	ROtate through eXtend Left	B, W, L	↑	↑	↑	∅	↑	Dy-t Dx helyel Dx-et "adat" helyel "ea"-t 1 helyel 
ROXR.X Dx, Dy ROXR.X # "adat", Dx ROXR.X "ea"	ROtate through eXtend Right	B, W, L	↑	↑	↑	∅	↑	Dy-t Dx helyel Dx-et "adat" helyel "ea"-t 1 helyel 
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: ∅ → "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.
SUBI.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

E X C E P T I O N S

Szám	Cím	Funkció
\$ ∅	\$∅∅∅	RESET esetén SSP (Supervisor Stock - 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 intervallumon 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, útközben mindenféléről medítál: az emberek és a technika viszonyáról, a tudományos megismerés nehézségeiről, a jó gépkönyvirásról, az analitikus gondolkodásról, a kantai 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 hivatalos indoklás szerint „éretlenség és a tanulmányok elhanyagolása” miatt. Fokozott érzékenysége, tehetségére túl korán és roppant erővel nehezettek rá a tudomány nagy kérdései. Szembesülésében kételyek merültek fel benne a megismerés általános elfogadott módszereivel szemben. (Mi is írtunk ilyesmiről rovatunk 1987/11. számában.) Idegrendszere túlterhelődhetett, konfliktusba került környezetével. Végül már kötelező sokkos kezelést írtak elő számára. Ettől újra „normális” lett, de emlékezetéből kiesett múltjának legnagyobb része. „Hősünk” az alaptörténet, az utazás idején már mint műszaki író gépkönyvek, műszaki leírások készítésével-szerkesztésével keresi kenyerét. Egyéni törettetéseit tetézi, hogy az orvosok fiánál is a kezdődő elmebetegség tüneteit vélik felfedezni. A tájleírásokkal váltakozó meditációk súlyát a korábbi énjét, saját azonosságát 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 eszmefuttatásaiból ízelítőt 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 *meditációt* 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 benaresi egyetemen keleti filozófiát tanulva, de *nem* lett a buddhista tanok híve. Mint az empiria (a tapasztalatokon alapuló ismeretek) elsődlegességét valló tudós ment Indiába és így is tért vissza — némi-képpen csalódottan és kissé ingerülten is, meg egy fontos tanulsággal gazdagabban: úgy találta, a hinduizmus, a buddhizmus és a tao közötti doktrinális különbségek kisebbek, mint a „nyugatibb” vallások közötti különbségek, mert ott, a Távol-Keleten a verbalizált, szavakban megfogalmazott állításokat soha nem tekintették azonosaknak az azokban és a velük leírt realitásokkal.

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

A meditációk tehát a technika és az ember viszonyára, egyes emberek technikaellenességének 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 egyetenteni, 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 geometriákról* találunk érdekes eszmefuttatásokat.

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 ízelítőül két rövid részlet felvillantására és közbeszúrva egy „meditációra” arról, miért nincsen nálunk elegendő mérnök-író.

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 keresem meg a kenyeremet és tudom, hogy ezek is mennyire tele vannak hibákkal, kétértelműségekkel, követhetetlen ugrásokkal és annyira összekutyult információval, hogy gyakran hatszor is el kell olvasnom ezeket, amíg megtalálom bennük az értelmet. De ami már az első alkalommal melbe vágott bennük, az a javítóműhelyekben is tapasztalt, szinte megegyezélesen *kivülálló*, nézgelődői magatartás (alapállás), amivel ezeket a gépkönyveket írták. Mert szinte sugallják: „ez itt a gép, időben és térben *elszigetelve* a mindenség összes többi dologától”.

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

Nem tudatosul az olvasóban az, hogy itt az író kezében egy *láthatatlan kés dolgozott*. Egy meglehetősen halálos kés, egy intellektuális 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 vagdosni.

Mondjuk a motorkerékpár *leírásában* mint önálló egység szerepel egy ún. *viszacsatoló mechanizmus*, amelyik magába foglalja a büttykös tengelyt, a vezérlő láncot, a gyújtásszabályozót, a megszakítókat stb. Ez a „része” a motorkerékpárnak kizárólag azért „létezik”, mert az előbbi — a gépkönyvkészítésnél használt — analitikus késnek *éppen így tetszett* szétvágni a teljes rendszert. Ha például bemennél egy motorkerékpár-szaküzletbe és megpróbálnál egy „viszacsatoló 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* vágná szét a motorkerékpárt. Minden szerelő jól ismeri a problémát: 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 kése *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 gépeik, programjaik 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 most 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-ÍROKBAN”

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 lásd a vevő, hogy a tervező milyen okos, milyen *bonyolult* szerkezetek 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 hivatásos 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 születnek igazán jó, piacképes termékek sem. (A hiánygazdaságok „piacán” persze sok minden eladható még, ami a valódi piacon nem!)

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

Felmerül még az a kézenfekvő kérdés is, hogy *dokumentálhatatlan-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* sebességváltó szerelvény részeinek tekintem a benne lévő tengelyeket, a fogaskerekeket, és nem fordítva.

A számítógépeknél — elsősorban a programoknál — ez az egyszerű „*bennfoglalási*”, strukturálási elv már nem ennyire egyszerűen követhető: szoftvert lehet úgy is tervezni, hogy tulajdonképpen „*semmi sincs igazán semmiben*”. Ez a hagyományos gépeknél szinte lehetetlen volt. Tipikusan nem „*haraphatnak egymásba*” a gépek alkotóelemei, mert akkor nem lehetne rendszeren szétszerelni és összeszerelni őket. — 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 melletti 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 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.

Kezdetben még szórakoztatónak is találta ezt a „felfedezését”. Parkinson nyomán meg is fogalmazott egy vicces törvényt: „*egy-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 iránt, 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 elékezett volna minden tudományos módszer általános érvényűségének katasztrofális cáfolatához!

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 pillanatban az összes elképzelhető konstrukcióból *egy* mindig magasán a többi felett állónak bizonyult.” De főhősünk számára ez nagyon gyenge válasz volt. Az „*egy adott pillanatban*” kitétel valósággal megrázta 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 elfogadottnak 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 olyasmi, ami az idők végtelenségéig elfogadott marad, hanem *időleges*, amelyet ugyanúgy lehet *tanulmányozni*, további vizsgálatok tárgyává tenni, mint bármi mást. Az 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 nézel, annál többet látsz!) Ahelyett, hogy *szelektálnád* a helyes magyarázatot a hipotézisek sokaságából, inkább az ellenőrizendő *hipotézisek sokaságát növeled*. Ahogy 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 egyetérteni a könyv minden gondolatmenetével, a hipotézisekről szólóval sem. A szerző végül is nem „tanokat hirdetett”, hanem (jó) regényt írt.

— KE —

CENTRUMKONTROLL

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 felhasználik az állásértékelésnél. Igaz, a különböző programok más-más értékelést alkalmaznak, mindegyik más súlyal 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 térfelén lévő 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 különválasztjuk 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:

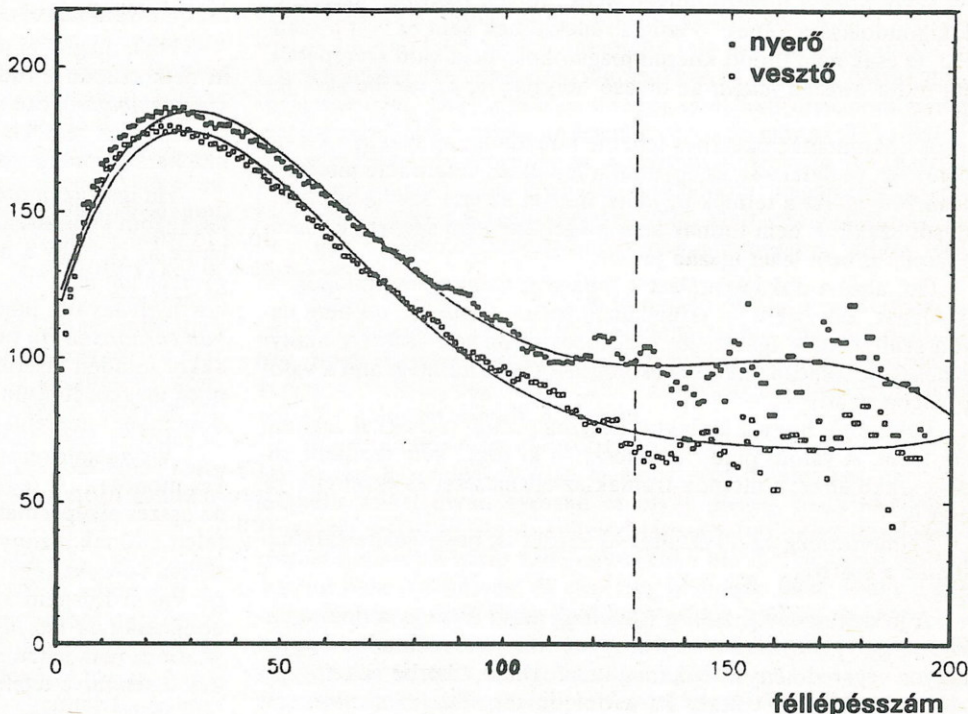
$$F_{cen} = B \times (\text{támadók száma} + 2 \times \text{védők száma}),$$

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

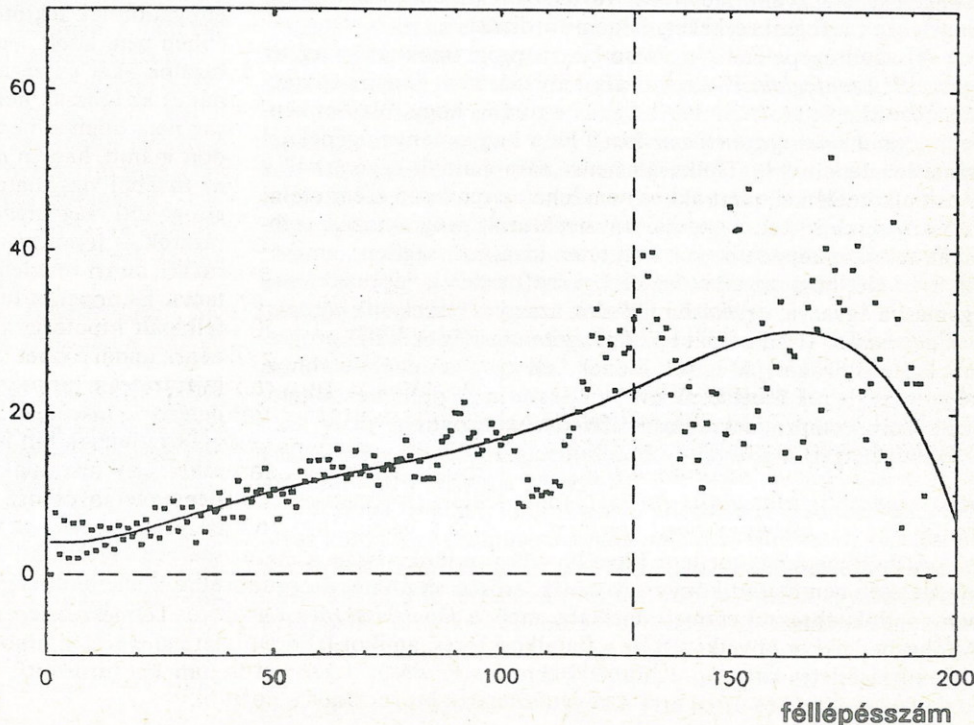
Centrumérték



• 2. ábra

A nyerő és a vesztes centrumértékének különbsége

3. ábra



A centrumérték mezők szerinti elosztása 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 rovatba. Remélem, a kedves Olvasók megírják ezzel kapcsolatos véleményüket is.

Dudás Gyula, Göd

Örülök, hogy kedvenc lapom profilként 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 gyerekújságot, 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. Ezt lehet kismértékben hajlítani (amíg meg nem törik). 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 helyezve 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 százezres 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 hanglemezzel 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. Én egyetértek és csatlakozom Boros Péterhez abban, hogy különböző játékokhoz és felhasználói programokhoz lehetne részletes leírást, térképet, ötleteket közölni. Jómagamnak 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 játékon 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 Magazinnak kellenie felkarolnia a „játék-leírást”, vállalkozást. Természetesen nagyban függene 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 olvasókhoz. 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. Mélységesen 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 köztünk nézeteltérés. Mi mindig és mindenhol a programlopást, az illegális programmásolást és terjesztést ítéltük és ítéljük el. Nem tartjuk etikusnak, ha valaki más szellemi produktumából csinál üzletet, annak a másiknak 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 kockáztató szerzőn 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 beültetem az IC-keket 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 kimenetére kell csatlakoztatni minimális belső módosítás után. Ennek megépítése esetén szükséges-e az IC-keket beültetni?

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

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 feltétlenül jó ügyet szolgálna. Az évi 12 megjelenés persze a piacon található nem csekély számú programhoz mérten nem túl sok lehetőség volna, de ha megvalósítunk valamelyes előszelekcíót, 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 módszereként a szakfolyóiratok tesztelési szisztémáit alkalmaztuk, hozzáigazítva azokat sajátosságainkhoz. Az év folyamán igyekeztünk megtartani ugyanazt a rendszert. A szisztematikusan vizsgált adatokon kívül igyekeztünk feltárni egy sereg heurisztikusan meghatározható 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 tételnek 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 programvágya. Ez a TII korábbi oktatásiprogram-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.

SZUMMA

Terméknév

Az eddig boncolgatott terméknevek felsorolása haszontalan volna. Inkább az érintett témaköröket tekintjük át. Voltak olyan termékek, amelyek kifejezetten csak a számítástechnikai háttérrel foglalkoznak (Assembler-oktatás). Mások szerzőszámokat 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 szellemben 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 betöltőket feltalálták, a sebességsökkenésből 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 dokumentációk kivitelezése olykor eléggé kritikán 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, sőt 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 termékeket 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 nyúlású vádossággal 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 iskolákban 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

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

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

A félvezetőgyártók széles választékban állítanak elő egytokos 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 perifériáiban.

A könyv alapvető célja a különféle típusok bemutatása, tulajdonságaik 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 egytokos mikroszámítógépek sok potenciális felhasználója viszonylag kevés elektronikai ismerettel rendelkezik. Várhatóan a könyvben leírtak az ilyen felhasználók érdeklődését is fel fogják kelteni.

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 tervezőmunkán túl a hozzáértés és kreativitás területéről származó programozási gyöngyszemeket veszik nagytitok alá. Az igazgyöngyöt a kagylók növesztik a testüket irritáló homokszemek köré, ezek a programozási gyöngyszemek ugyanígy a programozókat irritáló valóság 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 programhatékonytárgyalja, 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 STÚDIO 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 fognak 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 ismereteik, akik kezdetben idegenkedéssel fogadják az új technika bevezetését. A programmal megszerezhetők 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 eszköztárában a rejtelmeibe az információközlő berendezésektől a személyi számítógépekig. Az Ad-oktat 12 fejezetben tárgyalja az anyagot. Ennek el-sajátítása 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 alaposabban 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 miniszótár, amellyel a tanult fogalmak könnyen átismételhetők. Az Adatrend Kisszövetkezet által kifejlesztett Ad-oktat elsősorban azoknak ajánlható, akik a hagyományos irodát 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 helyesírási 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-ot 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 tárból van, a winchester-táron pedig további 800 ezer szó található.

A szótárt 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 eredményeképpen immár négy éve jelennek meg grafológiai célú szoftverek, melyek ára napjainkra elérhetően alacsonyra mérséklődött: például a Prologicom 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 személyzetes 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 alkalmazásuk 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) használatkor a gépi utasítások 20 százalékát a felhasználók aktiválják az idő 80 százalékában. Így az utasításkezelő folyamatok csökkenthetők a szoftvermunka növelésével. Hagyományos mikroprocesszorokat 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 Integraph 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 termékeit nagyobb RISC-hasonlósággal hirdetik. Várhatóan a jövőben a két termék összeillesztéséből olyan integrált áramkörök születnek, 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 tűnhetnek el a RISC megjelenése miatt, de használhatják azokat a módszereket, amelyeket a RISC-gépektől „tanultak”

ADOK — VESZEK — CSERÉLEK

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

ADOK

Amiga 500 számítógép eladó. 64'er, Happy Computer, CHIP, 68000'er, Amiga, Macintosh, PC Plus és egyéb NSZK magazinok féláron vásárolhatók, vagy más lapokra cserélhetők. Amiga felhasználókkal kapcsolatot keresek. Raiz Tamás, Budapest, Győri u. 8. 1123. Tel.: 566-941

Atari 800XL + XF 551 lemezegység kb. 300 programmal, szakirodalommal, joystickkal eladó. Sinclair QL csere is érdekel. Érdeklődni: 06-42-12539 egész nap.

Atari 65XE (amerikai gyártmány) típusú számítógép áron alul eladó. Érdeklődni: Nagy Jánosné, Várpalota, Pf.: 78 vagy a 71-950-es telefonon.

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

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

C16, Plus/4 programokat eladok (8 Ft/db) magnókazettára. Cím: Pauló Tamás, Békéscsaba V., Máriássy u. 51. 5600

Plus/4-re mindentudó személyi jövedelemadó-számító program a '88 őszi változásokkal eladó. Ára 400 Ft + kazetta, utánvéttel is. Bogdács Lajos, Derecske, Dózsa Gy. u. 15. 4130

C Plus/4, C16 tulajdonosoknak új 1551-es floppy eladó! Irányár: 15000 Ft. Somodi Géza, Győr, Rákos F. u. 10. 9024

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

C64 (régii) + SHOOT'EM UP CONSTRUCTION KIT kazettán (500) eladó. Keresem Amiga 500 tulajdonosok ismeretségét. Kótai Balázs, Sopron, Margitbányai u. 37/A. 9400 "D7MEGA".

C64 + 1541 + 1531 + joystick + Lightpen + 50 lemez + 1500 program + irodalom olcsón eladó! Sürgős! Michailov Mihály, Csorna, Vörös-H. ú. 71. 9300 Tel.: 333

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

Comodore 64 magnóval, több mint 400 programmal, joystickkal, jelentős mennyiségű dokumentációval + cartridge (gyorstöltő, monitor, reset, másoló) sürgősen eladó. Ár: 20000 Ft. Cím: Péntes Balázs, Balassagyarmat, Rákóczi u. 79. 2660

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

Enterprise programokat adok-veszek-cserélek (kazettán). Racskó Zsolt, Nyíregyháza, Hámán Kató u. 1. 4400

Enterprise programok eladók! (10-től 30 Ft-ig.) Válaszborítékért listát küldök. Zemen László, Budapest, Kada u. 141. fsz. 9. 1104

Enterprise programok olcsón eladók! Válaszborítékért elküldöm listám. Sala István, Budapest, Hársfa sétány 17. 1203

JOYSTICK SZERVÍZ a Flórián Áruházhoz közel! Javitás, magnófej-beállítás. játékpogramok kazettán és floppy. Budapest, III. Kerék u. 36. IV/24. Szerdán és pénteken 17-19-ig.

Junoszy televíziót videomonitorral alakítok (kép+hang) az eredeti funkciók megtartása mellett. Minden videokimenettel rendelkező számítógéphez alkalmazható, C128 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 eladó az alábbi programokhoz: AMICA PAINT, EGA V3.2 stb. Kérésre listát küldök. Honti József, Csákvár, Május 1. u. 11. 8083

MDM S/1/4" (Videoton TVC) és Olivetti AT 090 8" lemez meghajító eladó. Nagy Jenő, Győr, Tákó u. 2/A. 9025

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

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

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

Spectrum (48/128 k) és **C64** programok eladók. Lemezen és kazettán is megrendelhetők. Régi és a legújabb szoftverek egyaránt beszerezhetők. Kiss Gábor, Győr, Venyige u. 31. 9028

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

ZX-Spectrum (48 k) gépemet több száz programmal, kézikönyvekkel, esetleg magnóval eladom. Irányár 16000 Ft. Illés Tamás, Nyíregyháza, Fészek u. 200. 4400

TVC programokat adok-veszek-cserélek (550 prg.). Listát kérek. Molnár János, Szolnok, Jászi F. u. 10. VI/25. 5000. Tel.: 56/31-085

Tudja Ön, hogy milyen egyszerű a FASTLOAD CARTRIDGE-el dolgozni? Vállalom bővített karakterkészlet, Reset gomb, IRQ (pause) kapcsoló beépítését. Seikosha SP1200VC - 180VC nyomtatók magyarosítását, FASTLOAD, SPEEDTAPE programsegítő modulok készítését. Szívesen küldök tájékoztatót! Bártfai Barnabás, Budapest, Móricz Zs. u. 22. 1193. Tel.: 668-411/64 (délelőtt).

VESZEK

C16-hoz fényceruzát keresek. Ajánlatot a következő címre: Szabó Péter, Mátészalka, Alkotmány u. 26. 4700

C64-re Pascal fordítóprogramot keresek megvételre, új kazettán. Machlik János, Debrecen, Sárvári 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 választokat az alábbi címre kérem: Papp Roland, Kapuvár, Gimnázium Fő-tér 25. 9330

CSERÉLEK

Atari 800XL játék- és felhasználói programokat cserélek lemezen. Listát adok és kérek. Uhrin János, Békéscsaba, Madách 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 nevű programokat. Emmer Zsolt, Budaörs, Lévai u. 21. III/12. 2040

C16, Plus/4 és **C116** játék- és felhasználói programokat cserélnék kazettán. Listát kérek. Griecs György, Gábortelep, 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.

Pontvadászat

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 legjobbak névsorát is, akik a megérdemelt dicsőségen túlmenően könyvutalványt 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 tekintjük 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 ugyanekkor

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

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

$$a_1 = \frac{6}{5} \sqrt[4]{\frac{2}{3}} a_0.$$

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 megfontolás 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 ●●●

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

$a(t)$ 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

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

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

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

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 mindjárt 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)

Érdemes megjegyezni, hogy olyan folyamatok is létrejöhetnek bizonyos adatai paramétereiknél, 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

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

és így

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

is jelenti. Az adott függvény így 10^{-9} pontossággal $\ln x$ -et adja meg.

(5 pont)

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

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

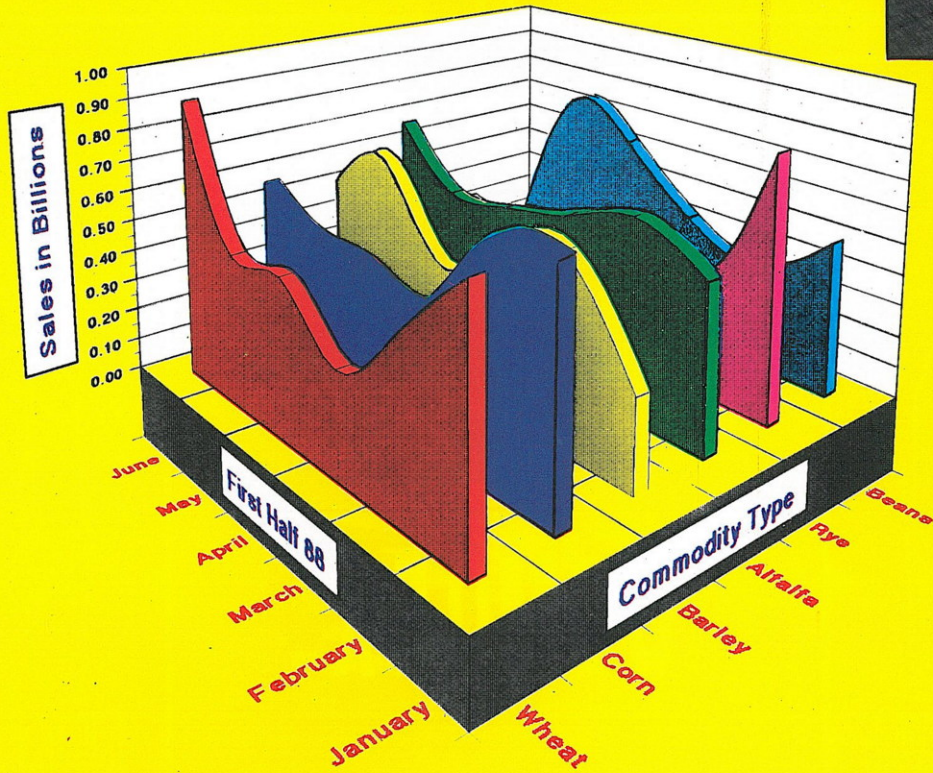
A feladat megnevezése	Maximális pontszám	Megfejtések száma	Megoldási százalék
1988/ 8. 1.	3	6	88,9
1988/ 8. 2.	18	3	77,8
1988/ 9. 1	4	5	85,0
1988/ 9. 2.	8	2	31,3
1988/10. 1.	5	2	60,0
1988/10. 2.	8	3	100,0
1988/11. 1.	6	2	75,0
1988/11. 2.	10	2	80,0
1988/12. 1.	12	2	83,3
1988/12. 2.	5	2	90,0

Nagy D. István, Csíkszereda, Románia	105,5 pont
Sterczér 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

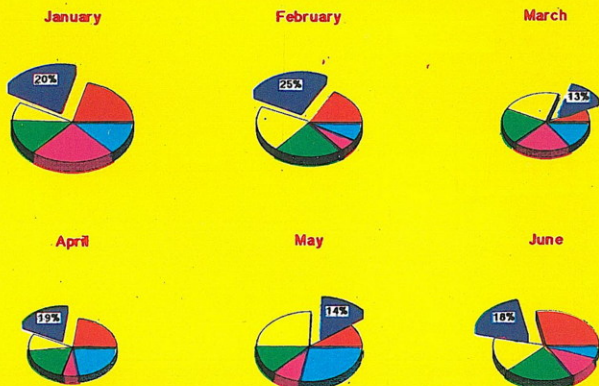
Ezeket a legjobb megfejtőket 200 forint értékű könyvtalvánnyal jutalmaztuk, amelyet postán küldünk el nekik.



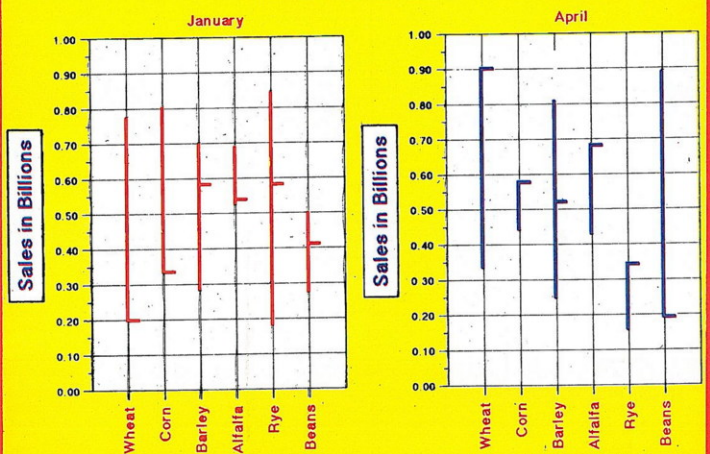
Commodities Sales



Commodities Sales

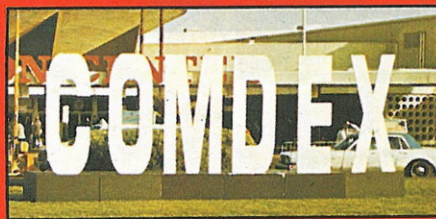
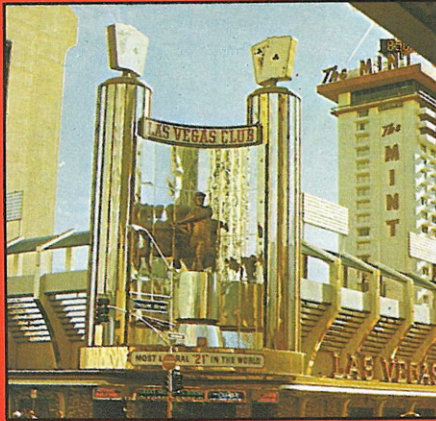


Commodities Sales



First Half 88

First Half 88



Amerikából jöttem...

