

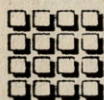
ENTERPRESS

KÉTHAVILAP AZ ENTERPRISE SZÁMÍTÓGÉPEK FELHASZNÁLÓINAK

II. évfolyam

4. szám

1991. július — augusztus



MÁTRIX Kft.



Vegyes gondolatok

Tombol a nyár, a hobbi számítógépesek igazi nagy idenye. Ebben az évszakban az összes felhasználó - legyen az diák, vagy akár dolgozó családapa - arra törekszik, hogy lehetőleg minél több ideje jusson a GÉP-re. Az igazi megszállottak képesek akár az egész nyarukat lesötétített szobában, kellemetlen hőségben, kilencven százalékos relatív páratartalommal kedvenc masinájuk mellett tölteni.

Ha valaki egyszer készítené egy felmérést arról, hogy mely időszakokban készülnek nagyobb számban jó minőségű programok a géphez, akkor biztosan a július és az augusztus hónapoknál lennének a maximális értékek. Sokan vágnak bele ilyenféle nagyobb programok készítésébe, kisebb hányaduk be is fejezi. Nem szégyen a kudarc, hisz nem feltétlenül a nemtudás az oka. A lényeg pedig egészen más: semmilyen tanfolyam, fej-tágitó nem adhat annyi tudást, tapasztalatot és élményt, mint amennyit az ilyen nagy nekibuzdulások adnak. Milyen jó dolog napokig csak egy-egy kis rutinocskával foglalkozni, minél ügyesebbé, gyorsabbá, rövidebbé csiszolgatni, hogy aztán később egy nagyobb program darabjaként barátokat ámulatba ejtő módon sziporkázzon.

Szerencsénkre van az országban jónéhány olyan ember, baráti társaság, akik csupán lelkesedésüktől fűtötten nagyszerű - profinak titulált cégeket megszegyenítő színvonalú - programokat, kiegészítőket készítenek a géphez. Egyre többen jelentkeznek közülük szerkesztőségünkkel, és mi is - a szerkesztés munkájába bevonandó - keressük őket.

Tesszük ezt azért, mert lapunk hatodik megjelenése vízválasztónak tekinthető: ha nem is jó, de legalábbis elfogadható eredménnyel zártuk az első évet. A lap következő száma kissé módosított tematikával fog megjelenni, egyes rovatok rövidülnek, ugyanakkor más rovatok - például a sokak által hiányolt hardveres - indulnak hódító útjukra. E sorok írásakor még úgy néz ki, hogy a megjelenést nem tudjuk gyakrabba tenni, és a terjedelmet sem tudjuk növelni. Van azonban két lehetőségünk, amelyekkel élni fogunk: csökkentjük a betűmértéket, és a címlapot „hasznos anyaggal” töltjük ki.

Olvasóink már bizonyára észrevették, hogy folyamatosan és őszintén tájékoztatjuk őket a lap helyzetéről, hiszen mindannyiunk szűgye, hogy még sokáig létezzon ez a fontos, felhasználókat összefogó kapocs. El kell tehát árulnunk: ha az Enterpress a következő bő félévben nem tud gazdaságilag jobb eredményeket produkálni, akkor a Mátrix Kft. felhagyja a kiadásával. A szükséges ellenintézkedéseket - reklámok, önálló terjesztés, további kapcsolatok kiépítése - már megtettük, mert mi még sokáig szeretnénk a lapot szerkeszteni!

Van azonban egy jelenség, amit egyáltalán nem értünk: nagyon kevés az előfizetőnk (a Postai és a kiadói együttesen hatszáz körüli), lapunkat keresve inkább mindenki napokig az újságosstandokat járja, legtöbbször eredménytelenül. (Hogy hová lehet ennyi Enterpress-t elrejtteni...?) A lap megnevezése érdekében ezért Olvasóinkat arra kérjük, hogy a kiadónál (Mátrix Kft., 8000 Székesfehérvár, Honvéd utca 8.) fizessenek elő az Enterpress-re! Az előfizetések számától - így Öntől is, kedves Olvasóink - döntő mértékben függ az Enterpress jövője. Garantáljuk a lap mindenkor megjelenését követő azonnali és biztos elküldését. Tesszük ezt azért is, mert valószínűleg megszűnik a lap utcai árusítása (nem mintha eddig igazából lett volna ilyen), hiszen a Posta egyszerűen képtelen ígéreteinek betartására, a pontos fizetésre, tájékoztatásra; mi pedig belefáradtunk az örökös és teljességgel felesleges csatározásba.

Térjünk kellemesebb témára! Jó hírnök van a programfejlesztők számára: szeptemberben indul a programküldő szolgálat! Bizonyára a sok felajzott pályázónak már nem kell részletezni, hogy ez pontosan mit is takar. Számukra írásban külön tájékoztatást készítettünk, melyet a szerkesztőségünk (1399 Budapest, Pf. 701/334) külön kérhetnek. Jó lenne, ha a nyár folyamán sok lelkes programíróval tudnánk szerződést kötni, hiszen így már következő számunkban nagyobb adag programot kínálhatnánk az érdeklődőknek.

A szerkesztők

TARTALOM

KURZUS

Assemly 6.	3-4
A Pascal 6.	5-6
Lehetőségek Páratlan Tárháza (LPT) 5.	7

PROGRAMOZÁSTECHNIKA

Basic programok titkosítása	8
Karakterek tervezése	8
Mandelbrot-halmaz	9-10

TIPPEK-TRÜKKÖK

Köztük: torz karakterek, villogó sorok	11
--	----

KÖNNYED MŰFAJ

Leírás és térkép: NAVY, ROCKSTAR MANAGER, SAVAGES	12-14
---	-------

MINDENFÉLE

Postafiók 334	15-16
Hirdetések, felhívások	16

ENTERPRESS

Kéthavilap
az ENTERPRISE számítógépek
felhasználóinak

II. évfolyam 4. szám
1991. július — augusztus

Kiadja a
**MÁTRIX Számítástechnikai,
Kereskedelmi és Szolgáltató Kft.**

Székesfehérvár

Felelős kiadó:

Annus István

ügyvezető

Felelős szerkesztő:

Ujlaki László (UL)

A lap szerkesztői:

Hajnal Csaba (HCs)

DevilSoft (Devil)

Címlap:

Németh Ferenc

Technikai szerkesztő:

Szapper László

A szerkesztőség és a kiadó címe:

8000 Székesfehérvár

Honvéd utca 8.

Telefon: (22) 12 - 619

Telefax: (22) 11 - 585

Levélcím:

ENTERPRESS

1399 Budapest

Pf. 701/334

Lapunk az ENTERPRISE Computers
GmbH kelet-európai képviselőjének, a
VTGe Electronics Ltd.-nek szakmai
támogatását élvezi.

Nyomja az

IMPEX MARKETING Kft. Nyomda

Székesfehérvár

Felelős vezető:

Szommer István

ISSN 0866-1820

Terjeszti a

Magyar Posta

Előfizethető a HELIR Bp. 1900 címen,

vagy a kiadónál

Előfizetési díj

egy évre 294 Ft, fél évre 147 Ft

Ára: 49,- Ft

Következő számunk

szepember 20-án jelenik meg

Az ENTERPRESS-ben közreadott információk célja az, hogy segítsék, tudnivalókat lássák el az ENTERPRISE számítógépek felhasználóit. A közölt programokat, kapcsolási rajzokat és leírásokat mindenki szabadon felhasználhatja, de tilos azokat a kiadó írásbeli engedélye nélkül másolni, terjeszteni.

A szerkesztőség köziratokat nem őriz meg, és nem küld vissza.

Assembly

6. rész

Az utasítások és a programok elkészítéséhez nélkülözhetetlen ASMON megismerése után nekiállhatunk elkészíteni első rutinjainkat. Ezeknek egyelőre még semmi közük nincs az EXOS-hoz, céljuk inkább az, hogy az utasítások működését és az ASMON kezelését bemutassák.

Az első rutin (1. lista) megváltoztatja a képernyő keretszínét.

```

;      1. program
border org 01000h ;forditasi cim
white equ 0bfe0h ;keretszin cime
      equ 255 ;a fehér kodja
      ld a,white ;feher lesz
      ld (border),a ;feher lett
      ret ;visszateres
      end ;forditas vege

```

1. lista

Az ASMON betöltése után nyomjuk meg az [E]-t, így az editorba jutunk. Gépeljük be a programot, ügyelve a következőkre:

- a szimbólum- és a címkeneveket mindig a sor elejére írjuk,
 - az utasításokat (praktikusan) egy tabulátornyival helyet beljebb helyezzük el,
 - az esetleges megjegyzéseket pontosvessző (;) után tesszük.
- Látható, hogy az assembly program egy-egy sora az alábbi részekre tagolódik:

[szimbólum-/címkenev] utasítás (az operandusokkal) [:megjegyzés]

A két szélső rész azért áll szögletes zárójel között, mert alkalmazásuk csak esetleges, nincs rájuk mindig szükség.

Nos, a program első sora egy megjegyzést tartalmaz, a sokat mondó "1. program" szöveget. A megjegyzést jelző pontosvessző a sor lefele kerül, ilyet is szabad csinálni. Célszerű, ha minden rutint - legyen az akár milyen parányi - ellátunk a szükséges megjegyzésekkel. Az assembly programok önmagukban nehezen olvashatók, ezért fontos a röviden, szabatosan megfogalmazott megjegyzések használata. Könnyen megtörténhet, hogy egy nem dokumentált, betekkel (esetleg napokkal) ezelőtt saját magunk által készített rutinról csak hosszas fejtegetés után tudjuk kideríteni működését. Ne sajnáljuk tehát az időt a megjegyzések írásától, ezzel önmagunkon segítünk!

A második sorban szereplő "ORG 01000h" jelzi az ASMON-nak, hogy a hexadecimális 1000-es címtől kezdve kell majd a kódot elhelyeznie, minden programot ORG-gal kell kezdeni. Az ASMON-ról tudni kell, hogy maga a rendszer a 3. lapon, azaz a 49152-es címtől helyezkedik el, a mi forrásszintű programunk pedig - alapértelmezésben - a 800h címnél kezdődik. Az ORG érték megadásánál tehát előre meg kell becsülnünk a forráslista méretét azért, nehogy a tárgy kód felülírja azt. Assembly-s körökben egyébként mindenki az ORG 01000h használja, merthogy az ASMON alapján ezt kínálja.

A harmadik sor elején a "border" szimbólumnév áll, melynek értéke 0BFE0h lesz. Erről az címről most csak annyit kell tudni, hogy ha a rendszerszám a 2. lapon van, akkor ennek a címnek tartalma határozza meg a keret színét (itt helyezkednek el az EXOS változói).

A fehér szín kódja a 255, ezzel tesszük egyenlővé a "white" szimbólumot. Érdemes ennél a pontnál néhány szót ejteni a programok nyelvezetéről: Sokan és sokat vitatkoznak arról, hogy angol vagy magyar kifejezéseket használjanak-e a programírók munkáikban. Nincs szándékomban eldönteni a vitát, én mindenképpen az angol kifejezésekhez ragaszkodom. A főindok: a számítástechnika nyelve (is) az angol. Igen zavarónak tartom a "magyarul írt" programokat, hiányoznak az ékezetek (így az magyarosnak már aligha nevezhető). Ráadásul a magyar címke- és szimbólumnevek legtöbbször - elnézést a kifejezésért - idétlennek tűnnek, nehéz rájönni, hogy jelentésük mit is takar, az angol sokkal kifejezőbb. A sorozat kedvéért kissé eltértem elveimtől, a megjegyzéseket magyarul (ékezetek nélkül...) fogom írni, így talán a többség számára jobban érthető. Egyébként döntse el mindenki, hogy melyik nyelvet használja.

Igazából maga a program az ötödik sornál kezdődik: az akkumulátorba töltjük a fehér szín kódját. Ebben a sorban látszik már, hogy milyen jó dolog a szimbólumhasználat. Az "LD A,white" sokkal többen mond, mint az itt vele teljesen ekvivalens "LD A,255". Azonnal kiderül, hogy a fehér színnel fog valami csalfintáság történni. Persze, azért a szimbólumok használatát sem kell tűzásba vinni, olyan helyen kell csak őket használni, ahol ez az áttekinthetőséget, a program olvashatóságát, a hordozhatóságot stb. szolgálja.

A forráslista hatodik sorában töltjük le a fehér szín kódját tartalmazó akkumulátortartalmat a keret színét meghatározó címre. Ennek hatására a keret színe ténylegesen felveszi majd az új színt, tehát ez a sor a rutin lényege.

A hetedik sorban a "RET" hatására (a tényleges kód lefutását követően) az ASMON-hoz kerül majd vissza a főszerep. A fordítás végét az "END" direktíva zárja.

A lista begépelése után az [F8] leütésével jutunk vissza az ASMON parancssorába. Következő feladatunk, hogy lefordítsuk az editorban lévő forrásszintű programot. Most csak a memóriába akarunk fordítani, üssük le a [Z]-t (német gépeknél [Y]), és az ASMON által feltett kérdések közül csak a "Memory assembly"-re válaszoljunk "YES"-t, a többi opciót hagyjuk változatlanul, ezeknél csak az [Enter]-t nyomjuk le. A fordítási feltételek megadása után az [A] billentyű leütésével indítható a fordítás. Sikeres fordítás után a képernyőn a

Pass 1.

Pass 2.

No error.

sorokat olvashatjuk. A "Pass" sorok az egyes fordítási meneteket futatják, a "No error" a fordítás hibamentességét jelzi. Előfordulhat, hogy egyeseknél mindenféle misztikus üzenetek jelentek meg a képernyőn. Ők valamit félregépeltek, ellenőrizték a forráslistát!

Vizsgáljuk meg, hogy valójában mi is történt! A fordítást a memóriába kértük, a 01000h címtől kezdve. Ennek tehát ott is kell lennie, meggyőződhetünk erről, ha megnézzük a tárgy kódját. Nyomjuk meg az [L]-t, írjuk be az 1000-t, és üssük le az [Enter]-t, majd néhány sornyi listázás után az [Esc]-et! Ekkor a következő sorokat látjuk:

```

1000 3E FF LD A,FF ;
1002 32 E0 BF LD (BFE0),A ;2?
1005 C9 RET ;

```

Ez a tárgy kód, visszafordított formában. Itt már nincsenek szimbólumok, címkék, megjegyzések, csak maga a jó kemény gépi kódú program. Az egyes sorok elején az a címérték látható, ahonnan az adott utasítás kódja letárolásra kerül. A következő, bajtjokot jelentő értékek (maximálisan négy) az illető utasítás gépi kódú megfelelőjét jelenti(k). Így az "LD A,FF" gépi kódja a "3E FF", a teljes utasítás itt két bajt hosszú. Ezután következik a kódhoz tartozó assembly kód, és végül a tárgy kód "fizikai megjelenését" bemutató pontosvessző utáni karakterek.

Az első utasítás tehát az 1000h címtől kezdve két bajtnyi helyet foglal a tárban, így a következő utasítás az 1002h címre kerül. A letöltött utasítás három bajt hosszú, érdemes megfigyelni, hogy a memóriacím (BFE0) ténylegesen alacsony rész (Low), magas rész (High), azaz "E0 BF" formában tárolódik.

Az egyetlen bajtnyi helyet elfoglaló "RET" az 1002h+3=1005h címre kerül.

Bár a visszafordított listában már értelemszerűen nem szerepelnek a szimbólumnevek, megjeleníthetjük őket, ha leütjük az [@]-t (népszerű néven a "kukac"-ot, német gépeken az [U]-t). Így megjelenik a következő sor:

```

BORDER BFE0 WHITE 00FF

```

Miután kigyönyörködtük magunkat, próbáljuk meg lefuttatni a programot! Ha a rendszer pillanatnyi állapotában indítanánk el, semmit sem látnánk, mert nincs a kettős lapra lapozva a rendszerszám (erről a [V]-vel győződhetünk meg). Először tegyük ezt a helyére az [O] leütésével, a B2 [Enter] és az FF [Enter] megadásával! Jöhet a program indítása: [G] 1000 [Enter]. S kőn: a keret színe fehér lett.

Második programunk (2. lista) már összetettebb. Nem elégünk meg azzal, hogy megváltoztatja a keret színét, az összes szín felhasználásával villogtatni akarjuk a keretet.

A fordítást az 1000h címtől kezdjük. Értéket adunk a "border", a "seg" és a "page2" szimbólumoknak. Ez utóbbi kettőt azért építjük bele a programba, mert így futtatás előtt nem kell a parancssorból beállítanunk a kettős lap helyes tartalmát; a program magától elvégzi ezt a feladatot. A "NOP" utasításnak a lépésenkénti végrehajtásánál lesz egy kis szerepe (az ASMON egy kisebbfajta hibáját küszöböli ki), a program lényegéhez semmi közük.

A tulajdonképpeni program a forráslista hetedik soránál

kezdődik. Az ENTERPRISE memóriaszervezéséről a sorozat legelső részében esett szó. Az ott elmondottakhoz most már csak annyit kell hozzátenni, hogy az egyes lapok tartalmát négy port állapota határozza meg. A nullás laphoz (P0) a 0B0h, az egyeshez (P1) a 0B1h, a ketteshez (P2) a 0B2h, a hármashoz (P3) a 0B3h portcím tartozik. Ha tehát a kettes lapon a 255-ös sorszámú rendszerszegmenst akarjuk használni, akkor a 0B2h portra 255-öt kell írunk. Pontosan ezt teszi a hetedik és a nyolcadik sor.

```

;      2. program
border org 01000h      ;forditási cím
      equ 0bfe0h      ;a keretszin cime
seg    equ 255        ;rendszerszeg. szama
page2  equ 0b2h      ;a kettes lap portcime
      nop            ;trace-hez...
      ld a,seg        ;rendszerszegmens
      out (page2),a  ;szegmenscitere
      ld b,0         ;ciklusszamlalo
      push bc        ;B mentese
      ld a,b         ;a szín kodja A-ba
      ld (border),a ;a szín letarolasa
      ld b,0         ;a másik ciklushoz
      djnz cycl2     ;idohuzas
      pop bc        ;B visszaallitasa
      djnz cycl1     ;folytatás?
      ret           ;visszateres
      end           ;forditas vege

```

2. lista

A kilencedik sorban a B regiszterbe nullát töltünk. A B regisztert - mint tudjuk - a DJNZ utasítás használja. Ha a regiszterbe induláskor a nulla érték kerül, akkor a ciklus 256-szor fog lefutni.

A következő sor elején álló "cycl1" címkét jelent, tehát azt a memóriacímet jelképezi, amelyre a "PUSH BC" utasítás fog kerülni. Az előbbi veremkezelő utasításra azért van szükség, mert egy belső DJNZ ciklust is használunk, és az elrontaná a külső ciklushoz tartozó regisztertartalmat. (A belső ciklust természetesen másképpen - ezerféle módon - is kialakíthatunk volna, de így legalább látunk egy kis veremkezelést.)

A keret színét mindig a külső ciklushoz tartozó B regiszter tartalma határozza meg. Az "LD A,B" utasítással betöltjük azt az akkumulátorba, majd az "LD (border),A"-val letöltjük a színt.

A következő sorban nyugodt lelkiismerettel elronthatjuk a B regiszter tartalmát az "LD B,0"-val, hiszen a külső ciklus számlálóértéke a stack-en van.

A belső ciklusra az "időhúzás" miatt van szükségünk. Ha ezt nem tennénk a programba, akkor az olyan gyorsan lefutna, hogy a villogásnak nem lenne "éltvezeti értéke". A "cycl2" címen lévő DJNZ önmagára ugrik vissza, így éri el a lassítást.

Amint véget ért a belső ciklus, azonnal helyreállítjuk a külső ciklushoz tartozó B regiszter tartalmát a "POP BC"-vel. A külső ciklus végét a "DJNZ cycl1" jelzi, a programot a "RET" utasítás zárja.

Ha végeztünk a begépeléssel, lépünk vissza a parancssorba! Nyomjuk le az [A]-t, így - hibátlan lista esetén - végrehajtódik a fordítás. A programot most lépésenkénti végrehajtásnál fogjuk lefuttatni, nyomjuk le ehhez a [T]-t, majd háromszor az [Enter]-t. A harmadik lenyomásra azért van szükség, hogy el-kaphassuk a végrehajtás legelejét is. Itt derül ki a program elejére becsempészett "NOP" szerepe: az ASMON a trace funkciónál nem tud azonnal megállni, csak a második utasításnál. Ha tehát ügyesek voltunk (ezt jelzi a képernyő bal alsó sarkában a "PC=1001"), akkor éppen a program elején állunk. Nyomogassuk az [Enter]-t, és figyeljük meg az A, B és az SP regiszterek és a Z flagbit változásait. Ha a belső ciklushoz értünk, nyomjuk le a [SPACE]-t, így az gyorsabban lefut. A lépésen-

kénti végrehajtásból az [Esc] leütésével juthatunk ki.

Fontos megjegyezni itt az ASMON egy igen bosszantó jellemzőjéről: Ha trace üzemmódban a végrehajtás a program végén álló "RET" utasításra fut, akkor nagy esélyünk van arra, hogy a verem tetején lévő szó tartalmától függően a teljes rendszer elszáll. Az utasítás hatására ugyanis (mint rendszeren) PC-be töltődik ez a szó, és a lépésenkénti végrehajtás innen folytatódik, tehát nem ér véget...

A program lefuttatásához nyomjuk meg a [G]-t, és írjuk be az 1000h-t, végül üssük le az [Enter]-t! A belső lassító ciklusnak köszönhetően a keretvillogás szemünk által érzékelhető "lassúsággal" megtörténik. A tisztelt Olvasóra bízom annak eldöntésével, hogy kilépéskor miért lesz vörös színű (kódja 1) a keret.

```

;      3. program
border org 01000h
      equ 0bfe0h
seg    equ 255
page2  equ 0b2h
      ld a,seg
      out (page2),a
      ld hl,0        ;a tar aljara mutat
      dec hl        ;mutato az ertekre
      ld a,(hl)     ;betoltes az A-ba
      ld (border),a ;a szín letoltese
      xor a         ;A=0
      or h          ;HL elerte-e
      or l          ;a nullat?
      jr nz,cycl1   ;ha nem, ujra
      ret
      end

```

3. lista

Harmadik programunk (3. lista) is a keretvillogtatáshoz kö-tődik, de itt a keret színét meghatározó értékeket a teljes me-mória, felülről lefelé történő olvasásával kapjuk meg. A lista első hét sorát már ismerjük.

A színek kódját tehát úgy akarjuk megkapni, hogy először kiolvassuk a tár tetején (a 65535-ös címen) lévő értéket, majd az ezalattit és így tovább, egészen a tár aljáig (a 0-ás címig) haladva. A program elején álló "LD HL,0" ezért ne tévesszen meg senkit: Mivel a következő utasítás a "DEC HL", így a HL tartalma a legelső felhasználáskor tényleg 65535 lesz. Látható, hogy a HL a programban mutatóként szolgál, mindig arra a tárrekeszre mutat, amelyik értékét keretszínnek akarjuk felhasználni. A színek az "LD A,(HL)" hatására kerül majd az akkumulátorba, a letöltés a szokásos módon történik.

A következő feladat annak eldöntése, hogy HL elérte-e a nullát, mert ebben az esetben abba kell hagyni a program (illetve a ciklus) folytatását. Mivel nincs 16 bites CMP utasítás, ezért valamilyen cselt kell kieszelnünk, ez a következő: Az akkumulátorba nullát töltünk (ezt teszi a "XOR A" művelet, amely az LD A,0-val teljesen azonos, csak rövidebb annál), és "belevagyoljuk" H és L tartalmát. Így ha HL tartalma nullával egyenlő, akkor A tartalma is nulla lesz, egyébként nem. Ha még nem értük el a memória alját, akkor az akkumulátorban az eredmény nem zéró (NZ igaz), így visszaugrunk a ciklus elejére.

A parancssorba lépve fordítsuk le a programot a memóriába! Azt javaslom, hogy lépésenkénti végrehajtásnál mindenki maga vizsgálja meg a program tényleges működését, az akkumulátor, a HL regiszter és a Z flagbit viselkedését. Még annyi tanácsod adok, hogy mindenki nyugodtan írja át a nyolcadik sorban álló "LD HL,0"-t mondjuk "LD HL,3"-ra, így senki sem fog egyenesen a gépe mellől nyugdíjba menni.

(folytatjuk)

-HCs-

Fizessen elő a

Hobby Elektronika és a Rádiótechnika

folyóiratokra! Így biztosan mindig hozzájut!

A cím: 1374 Budapest, Pf. 603. Tel.: 117 - 0262

A szerkesztőségben regisztrált HE előfizetőknek ingyenes nyák-film melléklet.

A PASCAL

6. rész

A programszerkezetek

Az adatszerkezetek - igen kimerítő - áttekintése után rátérhetünk a programszerkezetek tanulmányozására.

Míg az adatszerkezetek alkalmas megválasztásával egy átlagos alkalmazás esetén megtakaríthatjuk programsoraink mintegy 75 százalékát, a jó programszerkezetek használatával megtakaríthatjuk a programfejlesztésre és -belövésre fordított idő 75 százalékát.

Mielőtt a dolgok közepébe vágnánk, emlékeztetem az olvasókat a Pascal nyelv néhány felírástechnikai jellegzetességére.

Míg a Basic nem tesz különbséget az értékadás jele és az egyenlő relációs jel között, mindkettőre az egyenlőségelet alkalmazza, addig a Pascal két különböző jelet használ a két merőben eltérő funkció felírásához. Az egyenlőségelet meghagyja az egyenlőség vizsgálata számára, míg az értékadás jeleként bevezeti a BASIC-en felnőtt programozók számára szokatlannak tűnő := jelet. A megkülönböztetés pedig jogos; nézzük az alábbi programrészletet:

```
VAR
  L : BOOLEAN;
  A, B : INTEGER;
...
A := 1; B := 2;
...
L := A = B;
```

Az utolsó értékadásban a logikai típusúnak deklarált L változó akkor kapna TRUE, azaz igaz értéket, ha A és B értéke egyenlő volna; mivel ez nem igaz, L értéke FALSE, azaz hamis.

A másik sajátosság a zárójelek használatában van. Míg a BASIC csak a hagyományos kerek zárójelet használja, addig a Pascal igényt tart a szögletes és a kapcsos zárójelre is (sőt, még egy igen különös fajta zárójelre is, az *utasításcsárójelre*; lásd később):

```
( ) [ ] { }
```

A kerek zárójel egyrészt a kifejezésekben szerepelhet a matematikában megszokott módon, a kiértékelés sorrendjének befolyásolására, másrészt a függvények és eljárások paramétereit határolik. Emellett, mint láttuk, a felsorolási típus felírásában van szerepe a deklarációban. A tömbelemek indexelésére és a deklarációban a tömb méretének megadására viszont a Pascal a szögletes zárójelet használja. Mint láttuk korábban, a szögletes zárójel másik alkalmazása halmazgenerálásnál van. A kapcsos zárójel a programban megjegyzést jelöl. A megjegyzés a nyitó kapcsos zárójeltől a csukó kapcsos zárójelig terjed, így természetesen több sorra is kiterjedhet. Egyes régi klaviatúrákon nem szerepel a kapcsos zárójel, ezért a megjegyzés a

```
(* *)
```

összetett zárójelpárral is megadható.

Másik jellegzetesség a tól -ig határok jelölése. Ilyen a BASIC-ben nincs, mivel ott a tömbelemek mindig a 0 indexszel kezdődnek. A Pascal, mint korábban már megfigyelhetjük, megengedi tetszőleges kezdő- és záróindex használatát:

```
VAR
  A : ARRAY[ -5 .. 5 ] OF INTEGER;
  B : ARRAY[ -28 .. -13 ] OF REAL;
  C : ARRAY[ -1 .. 0, 0 .. 1 ] OF BOOLEAN;
```

A deklarációban az alsó és a felső határt választja el egymástól a két pont. (Csak megjegyzem, hogy a tetszőleges indextartomány megengedése egyáltalán nem felesleges gesztus; mint majd látni fogjuk, sokszor igen hasznos lehet.) Ugyancsak két pont választja el a résztartomány típus alsó és felső határának értékét. Ez nem véletlen: az index mindig résztartomány típusú:

```
TYPE
  SZINTIP =
  ( VOROS, NARANCS, SARGA, ZOLD, KEK,
  IBOLYA );
VAR
  INTENZITAS : ARRAY[ SZINTIP ] OF REAL;
  LUMINOZITAS : ARRAY[ VOROS .. IBOLYA ]
  OF REAL;
```

Még egy jelöléstechnikai dolog, és ezzel már bele is vágunk a programszerkezetek tárgyalásába: a Pascalban az utasításokat pontosvessző választja el egymástól. Ez kényelmes programtagolást tesz lehetővé, mivel nem kell (nem is lehet!) az utasítást egy sorba szorogatni; viszont veszélyt is rejt, hiszen egy esetleges szintaktikai hiba sem korlátozza a hatását egy fizikai sorra, hanem szinte bárhol kibukhat.

Az utasításrész

Nos, mint már korábban láthatuk, a Pascal program egy fejrészből és egy blokkból áll; a blokkot egyik összetevőjét, a deklarációs részt már megismertük; most nézzük meg részletesebben a másik összetevőt, az utasításrész.

Az utasításrész egy ún. *összetett utasítás*. Ennek jellemzője, hogy egy BEGIN és egy END kulcsszó határolja, szinte zárójelként: nem csoda, hogy e két kulcsszót *utasításcsárójelnek* nevezik. Az utasításcsárójelben tetszőleges számú utasítás állhat. Ezek lehetnek egyszerű utasítások, vagy újabb összetett utasítások, bár ez utóbbinak önmagában nincs sok értelme. De az utasításcsárójel-pár lehet üres is; ennek látványosan szintén nincs értelme, a valóság persze bonyolultabb (ha egy készülő programban a még el nem készült eljárásokat és függvényeket üres utasításrészrel felírjuk, akkor a fordítóprogram hajlandó elfogadni a programot, és ellenőrizhető a már elkészült részek helyessége).

Az értékadó utasítás

Az egyik egyszerű utasítás az értékadó utasítás. Ennek szintaktikája nem mutat semmi meglepőt: az utasítás *bal oldalán* egy *változó* áll, ez kap majd értéket; a *jobb oldalán* pedig egy *kifejezés*, ez határozza meg a változónak adandó értéket. A két oldalt az *értékadás operátora*, a már ismert := jelkombináció választja el egymástól.

Mivel a Pascal számos (pontosabban: számtalan) típust ismer, nagyon fontos, hogy a baloldal és a jobboldal típusának egy kifejezésen belül azonosnak vagy egymással kompatibilisnek kell lennie. A *változó* típusát mi határozzuk meg a deklaráláskor, a *kifejezés* típusát az őt alkotó elemek típusa és a rajtuk végrehajtott műveletek együttesen határozzák meg. Egy valós változó mindig megkaphatja egy egész kifejezés értékét, ezért a

```
VALOS_VALTOZO := EGESZ_KIFEJEZES
```

alakú értékadás helyes; egy egész változó viszont definíciója szerint szűkebb értéktartományt fed le, mint egy valós kifejezés, ezért az ellenkező kombináció nem megengedett, a fordítóprogram visszadobja. Ez a fajta korlátozás csak első pillantásra ellenszenves: sokkal jobb, ha azonnal kiderül a típuskeveredés, mintha futás közben jelentkeznének a megmagyarázhatatlan hibák.

A típusösszeférhetetlenség okozta konfliktusok feloldására szerencsére léteznek a Pascalban ún. *típuskonverziós függvények*, amelyek az egyik típusból a másikba alakítanak át egy értéket, ha ez lehetséges. Ezeket a függvényeket majd később nézzük át.

A Pascal értékadása sokkal hatékonyabb, mint a Basic hasonló funkciója. Most látjuk csak a sok-sok típus előnyét: sokszáz elemű rekordok és sokdimenziós, sokezer elemű tömbök kaphatnak értéket *egyetlen* értékadó utasítással!

Anélkül, hogy a kifejezés pontos szintaktikájának taglalásába belemennék (az érdeklődők megtalálják valamelyik Pascal könyvben), úgy lehetne összefoglalni: a kifejezést *változók*, *konstansok* és *függvényhívások*, illetve *operátorok* alkotják, a kiértékelés sorrendjét szükség esetén zárójelekkel adva meg.

A *változó* meghatározását ebben a pillanatban az olvasó intuíciójára bízjuk, induláshoz ez elegendő. A *konstans* lehet *alaki konstans*, amikor is a kifejezésben a konstans alaki értéke szerepel, és lehet *deklarált konstans*, ha a kifejezésben a korábban konstansnak deklarált azonosító szerepel.

A *kifejezés* kiértékelése a program által balról jobbra haladva történik, de befolyásolva az egyes operátorok ún. precedenciájától. Ez a Basic-ben is igen hasonló módon történik, csak a lehetséges operátorok száma kisebb.

A legmagasabb precedenciájú - tehát elsőként végrehajtható - operátor a *logikai negálás*, a jele NOT. Az utána álló (logikai típusú) clem értékét az ellenkezőjére változtatja.

A második lépést az ún. *multiplikatív* operátorok alkotják. Ezek a szorzás (*), az osztás (/), az egész osztás (DIV), a maradékképzés

(MOD) és a logikai ÉS művelet (AND). Az egész osztás során csak a hányados egész része keletkezik, míg a maradékképzés az egész osztás során keletkező maradékot adja eredményül.

A következők szint az *additív* operátoroké; ezek az összeadás (+), a kivonás (-) és a logikai VAGY művelet (OR). Emlékeztetem az olvasókat, hogy a + és a - jel a halmazokra és elemekre más jellegű műveletet határoz meg.

A legkisebb precedenciával a relációs operátorok bírnak; ezek az egyenlő (=), nem egyenlő (<>), kisebb (<), nagyobb (>), kisebb vagy egyenlő (<=) és a nagyobb vagy egyenlő (>=). Ide tartozik még a halmazoknál megismert IN operátor is.

Ha az operátorok precedenciája által meghatározott kiértékelési sorrend az adott esetben nem megfelelő, a sorrendet a matematikában megszokott módon zárójel alkalmazásával megváltoztathatjuk.

Az eljárásutasítás

A másik egyszerű utasítás az *eljárásutasítás*. Ez egy általunk definiált, vagy a Pascal rendszer által készen adott eljárásra való hivatkozás. Az eljárásutasítás végrehajtásakor a program végrehajtja az eljárás törzsében foglalt utasításokat. Ez nagy vonalakban a Basic GO-SUB utasítására emlékeztet, de egy sokkal korrektebb formában. Az eljárásutasítás a meghívott eljárásnak *paramétereit* adhat át. Az egész kérdéskört majd részletesen áttanulmányozzuk az eljárások és a függvények tárgyalásánál.

A ciklusszervező utasítások

A hagyományos Basic csak egyetlen ciklusszervező szerkezetet ismer, ezt a FOR ... NEXT utasításpárral valósítja meg. A Pascalban is létezik hasonló szerkezet, ugyanakkor egyéb ciklusszervező szerkezetek is léteznek.

A FOR ... DO utasítás

A FOR ... DO utasítást akkor használjuk ciklusszervezésre, ha előre tudjuk, hány alkalommal kell a ciklusmagot végrehajtanunk. Az utasítás szintaktikája a következő:

```
FOR változó := alsó_határ TO felső_határ DO
  utasítás
illette
```

```
FOR változó := felső_határ DOWNTO alsó_határ DO
  utasítás
```

Mint látjuk, a szintaktika hasonló a Basic-ben megszokotthoz. A *változó* a ciklusváltozó, ez veszi majd fel sorban az összes értéket az *alsó_határ* és a *felső_határ* között a TO kulcsszavú változatnál növekvő, a DOWNTO kulcsszavú változatnál csökkenő sorrendben. A DO kulcsszó szerepe csak az elválasztás. A ciklus minden egyes lefutásakor végrehajtódik az *utasítás*, amely természetesen lehet *összetett utasítás* is, ekkor a BEGIN ... END utasításcsoportot kell alkalmaznunk:

```
FOR változó := alsó_határ TO felső_határ DO
  BEGIN
    utasítás_1;
    utasítás_2;
    ...
    utasítás_n
  END
```

Az *utasítás* lehet újabb FOR ... DO utasítás is, pl.:

```
FOR I := 1 TO IMAX DO
  FOR J := 1 TO JMAX DO
    FOR K := 1 TO KMAX DO
      A[ I, J, K ] := B[ K, J, I ]
```

Mint látjuk, a Pascal-féle FOR ... DO utasítás nem ismeri a Basic által megengedett STEP kulcsszót, így a ciklusváltozó értéke minden lefutásakor eggyel (illetve mínusz eggyel) változik. Ez nem jelent problémát, a STEP-es alak egyszerű módon átalakítható a STEP nélküli alakba.

Itt meg kell jegyeznem néhány fontos szabályt. A ciklusváltozó csak egész típusú (vagy annak résztartománya), valamint felsorolási típusú lehet. Ez megint csak egy egyszerű korlátozás, valós típust akkor sem szabad ciklusváltozóknak használni, ha a programnyelv azt megengedi. Ennek az az oka, hogy a valós számokat a számítógép véges pontossággal ábrázolja, így egyrészt nem tudhatjuk, hogy a kezdő- és a végfeltételnél hogyan fog viselkedni a ciklus, másrészt a ciklus többszöri végrehajtása során az egyébként általában kicsiny értékű kerekítési hiba összegződhet, így egészen rossz eredményt kaphatunk.

tési hiba összegződhet, így egészen rossz eredményt kaphatunk.

A másik fontos alapszabály, hogy ciklusváltozóknak csak *lokális* változót használjunk, azaz a változót abban az eljárásban vagy függvényben deklaráljuk, amelyikben a ciklus van. Ellenkező esetben váratlan és nehezen felderíthető hibákra számíthatunk. Egyes Pascal fordítók figyelmeztetést adnak, ha a ciklusváltozó nem lokális.

A harmadik szabály - nemcsak a Pascalban - a ciklusba beugrás és a ciklusból kiugrás tilalma. Minden ciklusba csak az elején szabad belépni és csak a végén szabad kilépni. Ha ugyanis beugrunk a ciklusba, a ciklusváltozó nem kapott kezdőértéket, így a továbbiakban értéke meghatározatlan. A ciklusból kiugrás pedig áttekinthetetlenné teszi a program szerkezetét. Ha mindenképpen ki kell ugranunk a ciklusból idő előtt, inkább használjuk a WHILE ...DO vagy a REPEAT ... UNTIL szerkezetet. Végül, nem igazán elegáns, de még megengedhető megoldásként adjunk a ciklusváltozóknak a végértéknél nagyobb (A DOWNTO-s változat esetében kisebb) értéket, így a ciklusmag végén megtörténik a kilépés.

A negyedik szabály, hogy a ciklusváltozót a ciklusmagban nem szabad megváltoztatni. Ez akól hallgatólagosan kivételt teszünk az előző szabálynál megengedett esetben.

A WHILE ... DO utasítás

Az utasítást akkor használjuk, ha előre nem tudjuk, hogy a ciklusmagot hányszor akarjuk végrehajtanunk. Az utasítás alakja a következő:

```
WHILE feltétel DO
  utasítás
```

A végrehajtás során a gép megvizsgálja, hogy teljesül-e a *feltétel*. Ha igen, végrehajtja az *utasítást*, majd újra megvizsgálja a feltételt. Az *utasítást* így mindaddig végrehajtja, amíg a feltétel fennáll. Az alábbi programrészlet mindaddig olvas az F file-ból a C változóba, amíg a file végére nem érünk:

```
WHILE NOT EOF( F ) DO
  READ( F, C )
```

Az EOF(F) függvény TRUE (igaz) értéket ad vissza, ha az F file-ban már nincs több adat. A NOT EOF(F) kifejezés értéke így TRUE mindaddig, amíg *nem* értünk a file végére.

Az *utasítás* itt is lehet összetett utasítás is, pl. újabb WHILE ... DO utasítás.

A WHILE ... DO utasítás használatánál nagyon fontos, hogy a ciklusmag úgy működjön, hogy előbb-utóbb a *feltétel* értéke FALSE (hamis) legyen, különben a program soha nem lép ki a ciklusból. A

```
WHILE TRUE DO;
```

utasítás pl. garantáltan végtelen ciklust eredményez. Egyes rendszerekben ilyenkor azért a billentyűzetről megszakítható a program; vigyázzunk, nem mindenütt!

Arra is figyelniünk kell, hogy ha a *feltétel* értéke a belépéskor FALSE, a ciklusmag egyszer sem hajtódik végre.

A REPEAT ... UNTIL utasítás

Ezt az utasítást is akkor használjuk, ha nem ismeretes előre, hány alkalommal kell a ciklusmagnak lefutnia. Az utasítás alakja:

```
REPEAT
  utasítás_1;
  utasítás_2;
  ...
  utasítás_n
UNTIL feltétel
```

Belépéskor először végrehajtódik a ciklusmagot alkotó utasításcsoport, majd a gép kiértékeli a *feltételt*. Ha ez TRUE (igaz), a végrehajtás *kilép* a ciklusból, ellenkező esetben a ciklusmag ismételt végrehajtásával folytatódik.

Látjuk tehát, hogy a feltétel hatása itt pontosan az ellenkezője annak, amit a WHILE ... DO utasításnál láttunk.

Ügyeljünk arra tehát, hogy a REPEAT ... UNTIL szerkezetben a ciklusmag legalább egyszer minden esetben végrehajtódik.

Ez a szerkezet használható akkor, ha a FOR ... DO ciklusból való kiugrást el akarjuk kerülni:

```
I := 0;
REPEAT
  I := I + 1;
UNTIL MEGVAN( I ) OR ( I = IMAX );
```

Az általunk írt - BOOLEAN típusú - MEGVAN függvény vizsgálja, hogy az I-ik elem-e a keresett.

Lehetőségek Páratlan Tárháza (LPT) 5.

Befejezésül bemutatok négy látványos kép megjelenítési rutint. Természetesen mindegyik rutin működésének alapfeltétele, hogy az egyes LPB-k mindegyike egyetlen pixelsort definiáljon. A rutinok használata a következő: Először mindig a "DARK" címkejű rutint kell meghívni, majd a videomemóriát kell feltölteni a megfelelő információval, és végül meg lehet hívni a négy képbekapcsoló rutin bármelyikét.

Az egyes rutinok megértését a megjegyzések segítik. Fontos, hogy a rutinok meghívása előtt engedélyezzük a megszakítást, mert HALT utasításokkal van szinkronizálva; valamint az, hogy be legyen lapozva az LPT-t tartalmazó szegmens.

(Vége)

-DEVIL-

```

RADIX 10H
LPTADDR EQU 4000 ;LPT Z80-AS CIME
LPTLEN EQU 0F3 ;LPB-K SZAMA
VIDADDR EQU 4000 ;VIDEOMEMORIA NICK CIME
LINELEN EQU 50 ;PIXELSOR HOSSZA BAJTBAN
LEFMARG EQU 0BH ;BAL MARGO
RIGMARG EQU 33 ;JOB MARGO

DARK LD IX,LPTADDR+2 ;KEP KIKAPCSOLASA
LD DE,10 ;EGY LPB HOSSZA
LD B,(LPTLEN) ;LPB-K SZAMA
NDARK LD (IX),3FH ;BAL MARGO 3FH-RA
LD (IX+1),0 ;JOB MARGO 0-RA
ADD IX,DE ;KOVETKEZO LPB CIME
DJNZ NOARK ;VISSZA HA VAN MEG LPB
RET

BRIGHT1 LD IX,LPTADDR+2 ;KEPBEKAPCSOLAS 1
LD DE,20 ;KET LPB HOSSZA
LD C,LEFMARG ;BAL MARGO C-BE
LD A,RIGMARG ;JOB MARGO A-BA
LD B,(LPTLEN/2+(LPTLEN MOD 2))
NBRIGH11 LD (IX),C ;BAL ES JOBB MARGO
LD (IX+1),A ;BEALLITASA
ADD IX,DE ;KOVETKEZO UTANI LPB
HALT ;VARAKOZAS
DJNZ NBRIGH11
LD IX,LPTADDR+12 ;MASODIK LPB MARGOCIME
LD B,(LPTLEN/2) ;CIKLUSVALTOZO
NBRIGH12 LD (IX),C ;BAL ES JOBB MARGO
LD (IX+1),A ;BEALLITASA
ADD IX,DE ;KOVETKEZO UTANI LPB
HALT ;VARAKOZAS
DJNZ NBRIGH12
RET

BRIGHT2 LD IX,LPTADDR+4 ;KEPBEKAPCSOLAS 2
LD HL,VIDADDR+LINELEN*(LPTLEN-1)
LD DE,10 ;UTOLSÓ ELOTTI
LD B,(LPTLEN) ;PIXELSOR VIDEOCIME
NBRIGH21 LD (IX),L ;VIDEOCIMEK BEIRASA
LD (IX+1),H ;AZ LPB-KBE
ADD IX,DE ;KOVETKEZO LPB
DJNZ NBRIGH21
LD IX,LPTADDR+2 ;ELSO LPB MARGOCIME
LD DE,10
LD B,(LPTLEN)
LD A,LEFMARG
LD C,RIGMARG
NBRIGH24 LD (IX),A ;BAL ES JOBB MARGO
LD (IX+1),C ;BEIRASA
ADD IX,DE ;KOVETKEZO LPB
DJNZ NBRIGH24
LD HL,LPTADDR+LINELEN*(LPTLEN-2)
LD C,LPTLEN ;ALULRÓL A MASODIK
LD B,C ;PIXELSOR VIDEOCIME
NBRIGH23 LD IX,LPTADDR+4 ;ELSO LPB VIDEOCIME
LD DE,10 ;EGY LPB HOSSZA
NBRIGH22 LD (IX),L ;VIDEOCIM BEIRASA
LD (IX+1),H ;EGY LPB-BE
ADD IX,DE ;KOVETKEZO LPB
DJNZ NBRIGH22
LD DE,-LINELEN ;VIDEOCIM CSOKKENTESE
ADD HL,DE ;EGY PIXELSOR HOSSZAVAL
HALT ;VARAKOZAS
DEC C ;CIKLUSVALTOZO CSOKKENTESE
JR NZ,NBRIGH23 ;VISSZA HA KELL MEG
RET

BRIGHT3 LD B,(LPTLEN/2) ;KEPBEKAPCS. 3
LD C,1
LD HL,VIDADDR+LINELEN*(LPTLEN-2)
LD (BRIGVAR),HL ;ALULRÓL A MASODIK
NBRIGH33 PUSH BC ;SOR CIMENEK ELMENTESE
LD IX,LPTADDR+2 ;CIKLUSVALTOZO MENTESE
LD HL,(BRIGVAR) ;ELSO LPB MARGOCIME
LD B,C ;AKTUALIS VIDEOCIM
NBRIGH31 LD (IX+2),L ;BEOLVASASA
LD (IX+3),H ;ES BEIRASA AZ LPB-BE
LD (IX+1),RIGMARG ;LPB BAL ES JOBB
LD (IX),LEFMARG ;MARGO BEIRASA
LD DE,20 ;KOVETKEZO UTANI
ADD IX,DE ;LPB CIME
LD DE,2*LINELEN ;KOVETKEZO UTANI
ADD HL,DE ;PIXELSOR CIME
DJNZ NBRIGH31
LD DE,(BRIGVAR)
LD HL,2*VIDADDR+LINELEN*(LPTLEN-1)
OR A ;FELSO RESZBEN LEVO
SBC HL,DE ;SOR CIMENEK SZAMOLASA
LD IX,LPTADDR+10*(LPTLEN-1)+2-10*(LPTLEN MOD 2)
LD B,C

```

```

NBRIGH32 LD (IX+2),L ;VIDEOCIM
LD (IX+3),H ;BEIRASA
LD (IX+1),RIGMARG ;BAL ES JOBB MARGO
LD (IX),LEFMARG ;BEALLITASA
LD DE,-20 ;KETTOVEL FELJEBB
ADD IX,DE ;LEVO LPB CIME
LD DE,2*LINELEN ;KETTOVEL FELJEBB LEVO
ADD HL,DE ;PIXELSOR CIME
DJNZ NBRIGH32
LD HL,(BRIGVAR)
LD DE,-2*LINELEN ;VIDEOCIM VALTOZO
ADD HL,DE ;CSOKKENTESE KET
LD (BRIGVAR),HL ;PIXELSOR HOSSZAVAL
POP BC ;CIKLUSVALTOZO VISSZA
INC C
HALT ;VARAKOZAS
DJNZ NBRIGH33
RET

BRIGVAR DEFW 0

BRIGHT4 LD B,10 ;KEPBEKAPCSOLAS 4
LD C,1
LD HL,VIDADDR
LD IX,LPTADDR+2
NBRIGH42 PUSH BC ;REGISZTEREK
PUSH HL ;ELMENTESE
PUSH IX
LD A,C ;CIKLUSVALTOZO BEALL.
LD B,H ;VIDEOCIM BC-BE
LD C,L
LD HL,INTABLE ;TABLAZATCIM HL-BE
NBRIGH41 LD E,(HL) ;TABLAZAT
INC HL ;KOVETKEZO
LD D,(HL) ;ELEM DE-BE
INC HL
EX DE,HL
ADD HL,BC ;HL=VIDEOCIM+TABLAZAT-
EX DE,HL ;BOL KIVETT ERTEK
LD (IX+2),E ;LPT VIDEOCIM
LD (IX+3),D ;BEIRASA
LD (IX),LEFMARG ;MARGO
LD (IX+1),RIGMARG ;BEALLITASA
LD DE,-10 ;ELOZO LPB-CIM
ADD IX,DE ;SZAMOLASA
DEC A
JR NZ,NBRIGH41
POP IX
LD DE,10 ;KOVETKEZO LPB-CIM
ADD IX,DE ;SZAMOLASA
POP HL
LD DE,LINELEN ;KOVETKEZO SOR
ADD HL,DE ;VIDEOCIME
POP BC
INC C ;CIKLUSHOSSZ NOVELESE
HALT ;VARAKOZAS
DJNZ NBRIGH42
LD B,(LPTLEN-29)
NBRIGH44 PUSH BC
PUSH HL
PUSH IX
LD A,11
LD B,H
LD C,L
LD HL,INTABLE
NBRIGH43 LD E,(HL)
INC HL
LD D,(HL)
INC HL
EX DE,HL
ADD HL,BC
EX DE,HL
LD (IX+2),E
LD (IX+3),D
LD (IX),LEFMARG
LD (IX+1),RIGMARG
LD DE,-10
ADD IX,DE
DEC A
JR NZ,NBRIGH43
POP IX
LD DE,10
ADD IX,DE
POP HL
LD DE,LINELEN
ADD HL,DE
POP BC
INC C
HALT
DJNZ NBRIGH44
LD IX,LPTADDR+10*(LPTLEN-2AH)+2
LD L,(IX+2) ;KEP ALJANAK
LD H,(IX+3) ;HELREALLITASA
LD B,29 ;SOROKNENTI
NBRIGH46 LD DE,10 ;VIDEOCIM
ADD IX,DE ;BEALLITASAVAL
LD DE,LINELEN
ADD HL,DE
LD (IX+2),L
LD (IX+3),H
LD (IX),LEFMARG
LD (IX+1),RIGMARG
HALT
DJNZ NBRIGH46
RET

INTABLE DEFW 0,2*LINELEN,5*LINELEN,7*LINELEN
DEFW 8*LINELEN,0AH*LINELEN,0BH*LINELEN
DEFW 0CH*LINELEN,0CH*LINELEN,0CH*LINELEN
DEFW 0DH*LINELEN,0EH*LINELEN,10*LINELEN
DEFW 11*LINELEN,13*LINELEN,16*LINELEN
DEFW -10*LINELEN ;LEGORÜLESI TABLAZAT

```

Basic programok titkosítása

Sokakat érdekelhet, hogyan dugjuk el illetéktelenek elől elkészült Basic programjainkat. Ismerek erre egy eljárást, amely lehet, hogy sok felhasználónak már nem mond újat, de ilyen formában összegyűjtve tudomásom szerint még sehol sem jelent meg.

Az eljárás négy pontból áll, lássuk tehát sorra:

1., A Stop billentyű letiltása

Hatására a futó programot nem lehet a Stop billentyűvel megállítani. Megvalósítása Basic-ben:

```
SET INTERRUPT STOP OFF
```

2., A melegindítási cím átírása

A program a meleg reset-re (egyszer megnyomva a reset gomb) is "elszáll". Adjuk ki a

```
SPOKE(255,49144,0):SPOKE(255,49145,0)
```

parancsokat! Az 1., és 2., pontok közös hatása: ha a program már elindult, nem lehet megállítani. (Csak ha hibával kilép! Célszerű ezért a hibalehetőségeket a programból lekezelni HANDLER-ek használatával!)

3., Az autostartos indítás

A program a LOAD-dal való betöltés után is automatikusan elindul. A programot az ASMON-ba kell tölteni és a fejrész 5 számú bájtit - ami addig 00 volt - 05-re kell átírni. (Az 5. sorszámú bájtit sorrendben a hatodik! Az ASMON-ra vonatkozó idevágó tudnivalók az ENTERPRESS 1/1. és 1/2. számában az Örökéletkódoknál, illetve a II/3. számban az "Assembly 5." cikkben olvashatók el.) Az 1., 2., és 3., pontok együttes alkalmazása már jó programvédelmet jelent. Ezt a programot nem lehet megnézni, mert autostarttal indul, és nem lehet megállítani. Magától azonban még megállhat, ha hibára fut, vagy ha úgy írtuk meg, hogy egyszerűen a végére ér.

4., A listázás letiltása

A program listázása egy bizonyos sorszám után letiltásra kerül. Az eddigiek közül ez a legbonyolultabb és egyben a legveszélyesebb. Ha a program jól van megírva, erre a pontra már nincs is igazán szükség a fenti pontok alkalmazása után. Ha mégis alkalmazni kívánjuk, hagyjunk egy változatot az eredeti programból, nehogy hibázás esetén a teljes program elveszzen.

Ha a Basic program egyik sorszámát 9999-nél nagyobb számra írjuk át, a listázás ezen sorszámától elkezdve nem lehetséges. A sorszám felülírását szintén az ASMON-ban lehet megoldani, ehhez azonban ismernünk kell a Basic program tárbeli szerkezetét.

A Basic program a tárban tokenizált formában, soronként egymás után helyezkedik el. Egy sor négy fő részre bontható, ezek a következők:

- a sor hossza (1 bájtit)
- a sor Basic sorszáma hexadecimálisan (2 bájtit)
- a sor "tartalma" tokenizált formában
- a sor végét jelző bájtit (0).

Ha próbaképpen egy Basic programot az ASMON-ba töltünk, számoljuk le az első 16 bájtitot (ez a programfejrész), a következőtől kezdődik az első programor. Ezen bájtit tartalmával előre haladva (mivel ez jelzi a sor hosszát) a második sor hosszjelző bájtitjához érünk. Ez addig tart, amíg a sor hosszaként 0-t kapunk, ez már a program végét jelzi. A hosszjelző bájtitok tehát egy láncot alkotnak, melyen végiglépve bármelyik sort megtalálhatjuk, és átírhatjuk annak sorszámát.

Praktikus az a megoldás, amikor a program első sorának ezt választjuk:

```
100 PROGRAM "név.kit"
```

Ekkor a 110 sor sorszámát átírva listázáskor csupán annyi derül ki, hogy milyen program van a tárban. Az átírásához töltsük a programot az ASMON-ba, számoljuk le az első 16 bájtitot, a következő bájtit jelzi az első sor hosszát. Ezután a hexadecimális "64 00" következik, ami nem más, mint a decimális 100 (64h=100d), azaz a program első sora. A lánc következő tagja utáni két bájtit: "6E 00", ez a 110. Ha itt a "00"-t átírjuk "FF"-re, máris 9999-nél nagyobb sorszámot kaptunk. Ezután már csak ki kell menteni a programot! (Vigyázzunk a "Last address"-re!)

A fenti négy pont alkalmazásával készült program tehát így kezdődik:

```
100 PROGRAM "név.kit"
110 SET INTERRUPT STOP OFF
120 SPOKE(255,49144,0)
130 SPOKE(255,49145,0)
140 ...
```

Ezenkívül át van írva az autostart bájtit és a 110-es sor sorszáma. Így a program autostarttal indul, nem lehet megállítani, ha hiba miatt mégis kilép, listázáskor csak az első sorát mutatja magából.

Ez a programvédelem szerintem bőven elég ahhoz, hogy illetéke-

len személy ne tudjon belekontárkodni Basic programunkba. Természetesen aki ismeri az eljárást, az könnyedén feltörheti, de ez időt vesz igénybe, és egy Basic program titkosítása végülis nem egy egyszerűen fontos dolog, inkább csak az érdekesség miatt érdemelhet egy-két szót.

Hámori György

Karakterek tervezése

Az IS-Basic SET CHARACTER paranccsal a gép karakterképei könnyedén megváltoztathatók. A kezdők számára azonban bonyolultnak tűnik a szükséges jellemzők meghatározása, a karakterek tervezése. Nekik szánjuk az alábbi írást.

Mint a számítógépben minden, így a karakterképek szerkesztése is a kettes számrendszerhez kapcsolódik. Az egyes karakterek 8 bit szélesek, és 9 bit magasak, a teljes karaktert tehát 72 bit írja le. Tudjuk, hogy a bit tulajdonképpen egy kétállapotú kapcsoló, amely ha be van kapcsolva, akkor értéke egy, egyébként pedig nulla. Ha a kapcsolóállapotot levettjük a karakterekre, akkor azt mondhatjuk, hogy a bekapcsolt bit látszik, a kikapcsolt nem.

A karakter - tervezés szempontjából - 8 egység széles, 9 egység magas mátrixként fogható fel. A karaktert ebben a mátrixban kell megrajzolni (célszerű kockás papírt használni). Nem véletlen, hogy 8 egységből, azaz 8 bitből áll össze a karakter egy-egy sora: ezt bájtként lehet értelmezni, a gép ekként is kezeli. A bájtként tehát 8 bit foglal helyet, minden bitnek helyiértéke van.

	7	6	5	4	3	2	1	0	
1. sor									24
2. sor									36
3. sor									66
4. sor									126
5. sor									66
6. sor									66
7. sor									0
8. sor									0
9. sor									0

1
2 6 3 1
8 4 2 6 8 4 2 1

felső sorának értéke így $16+8=24$, hiszen a negyedik és a harmadik bit van egybe állva. Ezt az eljárást követve mind a kilenc sornál elvégezzük a számítást.

Ha elkészültünk ezzel, akkor a kapott számokat közölnünk kell a géppel, ezt a SET CHARACTER paranccsal tehetjük meg. A parancs első paramétere az az ASCII kód, amelyhez tartozó karakter képét át kívánjuk definiálni. Az ezt követő kilenc paraméter a karaktert alkotó kilenc értéket jelenti, elsőnek a karakter tetején állót kell megadni. Az utasítás formája tehát:

```
SET CHARACTER asll 164,sor1,sor2,sor3,sor4,sor5,sor6,sor7,sor8,sor9
```

Az ábrázolt karaktert tegyük az "A" betű helyére, ennek ASCII kódja a 65.

```
SET CHARACTER 65,24,36,66,126,66,66,0,0,0
```

Bizonyára sokan furcsálják, hogy miért nekünk kell ezeket az értékeket kiszámítani, miért nem bízzuk ezt a gépre? Nekik teljesen igazuk van, hiszen: "Dolgozzanak a gépek, gondolkozzanak az emberek". A Basic rendelkezik a BIN() függvénnyel, amellyel az egyes sorok értékei kiszámíthatók. A függvény felhasználására láthatunk példát a listán. Az egyes sorokhoz tartozó bájtitértékeket egy tömbben helyezük el. Ilyen "ábrázolásban" a karakterképek könnyedén tervezhetünk. Kár, hogy a Basic interpreter a BIN() elején álló nullákat eltüntet.

```
100 PROGRAM "CHARDEF.BAS"
110 NUMERIC C(1 TO 9)
120 LET C(1)=BIN(00011000)
130 LET C(2)=BIN(00100100)
140 LET C(3)=BIN(01000010)
150 LET C(4)=BIN(01111110)
160 LET C(5)=BIN(01000010)
170 LET C(6)=BIN(01000010)
180 LET C(7)=BIN(00000000)
190 LET C(8)=BIN(00000000)
200 LET C(9)=BIN(00000000)
210 SET CHARACTER ORD("A"),
C(1),C(2),C(3),C(4),C(5),
C(6),C(7),C(8),C(9)
```

A TANDEM Kereskedelmi és Szolgáltató Kft.-nél előnyös áron vásárolhat IBM kompatibilis számítógép-részegységeket, háttértárolókat, lemezeket és egyéb kiegészítőket. Cím: 1132 Budapest, Visegrádi utca 6. Telefon: 112-8604

Mandelbrot-halmaz

Valószínűleg minden olvasó tudja, hogy mi is az a Mandelbrot-ábra. A Mikroszámítógép Magazin egyik régi számában található egy ilyen témájú Basic program, amellyel ENTERPRISE-on is elővarázsolhatjuk ezeket a csodálatos képeket. Az említett program egyetlen hátránya, hogy egy kép kiszámítása akár 24 óráig is eltarthat. A sebesség jelentős mértékű növelése csak gépi kódú programmal lehetséges. Most egy ilyen program forráslistáját mutatom be, amely ASMÓN-nal fordítható le.

A működése a megjegyzések alapján követhető. A leglényegesebb része a 32 bites előjeles számokat összeadó és szorzó rutinok, amelyek más pontos számolást igénylő programokban is felhasználhatóak.

A program használata egyszerű: Miután begépeljük, lépünk ki az editorból, majd a [Z] (német gépeknél az [Y]) billentyű megnyomásával úgy állítjuk be a fordítási opciókat, hogy a gép a memóriába fordítson (Memory assembly: YES). Ezután az [A] billentyűvel fordítjuk le a programot, és végül a [G]-vel elindíthatjuk. A "START:" kérdésre 3000-et kell beírni. Elindítás után a program elkezd a kép megrajzolását, ami maximálisan egy órányi ideig tart. Ha befejezte a számolást, a keretszín átvált fehérre, jelezve, hogy egy billentyű lenyomására vár a program. Ha az [ESC]-et nyomjuk meg, akkor visszatér ASMÓN-ba, ha az [S] billentyűt, akkor pedig kimentti a képet "MANDEL.PBX" néven. A kimentett kép a Paintbox rajzolóprogramba tölthető be, ahol azt átábrázolhatjuk vagy akár ki is nyomtathatjuk. A programból a kép elkészülte előtt is kiléphetünk az [ESC] megnyomásával. A kép adatait és a nagytípus mértékét a program elején lévő EQU-oknál állíthatjuk be, ahol az előjel +1 vagy -1, az egész és tört részek 0 és 0FFFFh közötti hexadecimális számok lehetnek. A felbontás 1 és 1000h között változtatható.

A program működésének megértéséhez szükséges némi matematikai ismeret a Mandelbrot-halmaz felépítéséről. Minden halmazbeli elemnek a képernyőn egy pont felel meg, amelynek színe attól függ, hogy hány számolási ciklus alatt ér el egy megszabott határértéket. A Mandelbrot-halmazt az (X,Y)^(A,B) összefüggés adja meg, ahol az (X,Y) és az (A,B) egy-egy komplex számot jelent. Az (A,B) értéke a kiszámítandó pont helyétől függ, az (X,Y) pedig mindig az előző kiszámolt érték, első számolásakor nulla. Az így folyamatosan számolt szám vagy a végtelenhez tart, vagy a nulla körül ingadozva egy spirált ír le a komplex térben. Az utóbbi esetben a pont eleme a halmaznak, és a képernyőn fekete színnel jelenik meg. Ha azonban a végtelenbe "szalad" akkor attól függően, hogy hány ciklus alatt ér el egy bizonyos határértéket, egy színes pontot kapunk eredményül.

-DEVIL-

.RADIX	10H	
ORG	3000	
VALOSEL	EQU -1	:X KOORDINATA ELOJEL
VALOSEG	EQU 2	:X KOORD. EGESZ RESZ
VALOSTR	EQU 0	:X KOORD. TORT RESZ
KEPZEL	EQU -1	:Y KOORD. ELOJEL
KEPZEG	EQU 1	:Y KOORD. EGESZ RESZ
KEPZTR	EQU 3000	:Y KOORD. TORT RESZ
NAGY	EQU 280	:FELBONTAS MERTEKE
SOROZ	EQU 20	:HATARERTEK
START	DI	
	LD A,0CH	:VARAKOZAS
	OUT (0BFH),A	:TILTASA
	LD (STACK),SP	:STACK MENTESE
	LD SP,4000	:STACK BEALL.
	IN A,(0B1)	:1-ES ES 2-ES
	LD (SEG1),A	:LAPON LEVO
	IN A,(0B2)	:SZEGMENSEK
	LD (SEG2),A	:ELTAROLASA
	LD A,0FCH	:0FCH, 0FDH
	OUT (0B1),A	:SZEGMENSEK
	INC A	:BELAPOZASA
	OUT (0B2),A	
	LD HL,LPT	:LPT
	LD DE,4000	:ELGALLITASA
	LD BC,LENLPT	
	LDIR	
	LD HL,5000	:VIDEOMENORIA
	LD DE,5001	:TORLESE
	LD BC,4BF0	
	LD (HL),L	
	LDIR	
	XOR A	:KERETSZIN
	OUT (81),A	:FEKETE
	OUT (82),A	:LPT
	OUT (83),A	:BEKAPCSOLASA
	OR 40	
	OUT (83),A	
	OR 80	
	OUT (83),A	
	LD A,13	:BIAS SZINEK
	OUT (80),A	:BEALLITASA
	LD A,3	:ESC' BILL.
	OUT (0B5),A	:SORANAK BEALL.
	LD HL,5000+4BF0	:KEZDETI KEPCIN
	LD (KEPCIN+1),HL	:BEALLITASA
	LD A,1	
	LD (KEPCIN+1),A	
	CALL MANDEL	:RUTIN HIVASA
	LD A,3	:NA AZ 'ESC'
	OUT (0B5),A	:LE VAN NYORVA
	IN A,(0B5)	:AKKOR VISSZATERES
	RLA	:ASMONBA
	JR NC,STOP	
	LD A,7	:KERETSZIN
	OUT (81),A	:FEHER
	LD A,1	:VARAKOZAS
	OUT (0B5),A	:AZ 'S' BILL.

	IN BIT	A,(0B5)	:LENYOMASARA
	BIT	S,A	:NA LENYORVA
	JR NZ,VAR		:AKKOR
SAVE	CALL	VISSZA	:KEP KIMENTESE
	LD A,2		
	LD DE,NAME		
	EXOS 2		
	LD A,2		
	LD DE,PBFORM		
	EXOS 8		
	LD A,2		
	LD DE,5000		
	LD BC,4BF0		
	EXOS 8		
	LD A,2		
	EXOS 4		
	LD A,(SEG1)	:EREDETI	
	OUT (0B1),A	:SZEGMENSEK	
	LD A,(SEG2)	:VISSZALAPOZASA	
	OUT (0B2),A		
	LD SP,(STACK)	:VISSZATERES	
	RET	:ASMONBA	
STOP	CALL	VISSZA	
	LD A,(SEG1)		
	OUT (0B1),A		
	LD A,(SEG2)		
	OUT (0B2),A		
	LD SP,(STACK)		
	RET		
VISSZA	LD A,90	:LPT	
	OUT (82),A	:VISSZAALL.	
	LD A,0CFH		
	OUT (83),A		
	XOR A		
	OUT (80),A	:KERETSZIN	
	OUT (0BFH),A	:FEKETE	
	LD A,30	:VARAKOZAS BE	
	OUT (0B4),A	:MEGSZAKITAS	
	EI	:ENGEDELYEZESE	
MANDEL	RET		
CIKLY	LD C,0F3	:FUGG. SOROK	
	PUSH BC	:SZAMA	
	LD A,C	:A SORSZAMMAK	
	LD DE,NAGY	:ES A	
	CALL SZORZ1	:FELBONTASHAK	
	LD DE,KEPZTR	:MEGFELELO	
	EXX	:KEZDETI	
	LD DE,KEPZEG	:B' ERTEK	
	EXX	:KISZAMOLASA	
	EX AF,AF'		
	LD A,KEPZEL		
	EX AF,AF'		
	CALL PLUSZ		
	LD (KEPZ1LO),HL	:ES ELTAROLASA	
	EXX		
	LD (KEPZ1NI),HL		
	EXX		
	LD (KEPZ1EL),A		
	POP BC		
	LD B,0A0	:SORONKENTI	
	PUSH BC	:PONTOK SZAMA	
CIKLY	LD A,B	:AZ AKTUALIS	
	LD DE,NAGY	:X KOORD-NAK	
	CALL SZORZ1	:ES A	
	ADD HL,HL	:FELBONTASHAK	
	LD DE,VALOSTR	:MEGFELELO	
	EXX	:KEZDETI	
	ADC HL,HL	:A' ERTEK	
	LD DE,VALOSEG	:KISZAMOLASA	
	EXX		
	EX AF,AF'		
	LD A,VALOSEL		
	EX AF,AF'		
	CALL PLUSZ		
	LD (VALOS1LO),HL	:ES ELTAROLASA	
	EXX		
	LD (VALOS1NI),HL		
	EXX		
	LD (VALOS1EL),A		
	XOR A		
	LD (ELOJELX),A		
	LD (ELOJELY),A		
	LD DE,0	:KEZDETI	
	LD BC,0	:X' ES 'Y'	
	EXX	:ERTEKEK	
	LD DE,0	:MULLAZASA	
	LD BC,0		
	LD A,SOROZ	:HATARERTEK	
CIKLY	PUSH AF		
	LD A,(ELOJELX)		
	EX AF,AF'		
	LD A,(ELOJELY)		
	CALL SZORZAS	:HL,HL' = X*Y	
	PUSH AF		
	ADD HL,HL	:HL,HL' = 2*X*Y	
	PUSH HL		
	EXX		
	ADC HL,HL		
	PUSH HL	:HL,HL' ELMENTESE	
	EXX		
	PUSH BC		
	LD B,D		
	LD C,E		
	EXX		
	PUSH BC		
	LD B,D		
	LD C,E		
	EXX		
	CALL SZORZAS	:HL,HL' = X*Y	
	LD (MEGYZ1LO),HL	:ELTAROLASA	
	EXX		
	LD (MEGYZ1NI),HL		
	POP BC		
	EXX		
	POP BC		
	PUSH HL	:X HEGYZET	
	EXX	:ELMENTESE	

Egér=botkormány

Ha rendelkezünk Basic Plus kártyával, és van egy egerünk, akkor a parancs kiadása után az egeret úgy használhatjuk, mintha belső botkormány lenne.

SET 201.4

parancs kiadása után az egeret úgy használhatjuk, mintha belső botkormány lenne.

Fent és lent

Adjuk ki Basic-ben a SPOKE 255,15026,12 parancsot! Az [Enter] leütése után a képernyő tetején és az alján is a státuszszorban gyönyörködhetünk. (A két trükköt Szekeres Tamás küldte be.)

A szinusz hangjai

A szinusz-függvény egy érdekes alkalmazására mutat példát az alábbi program. A hang magasságát ugyanis a rutin közepére helyezett SIN() határozza meg.

```
100 PROGRAM "SINSOUND.BAS"
110 OPTION ANGLE DEGREES
120 DO
130   FOR I=0 TO 359
140     SOUND PITCH SIN(I)*32+32, INTERRUPT
150   NEXT
160 LOOP
```

Szimmetria

Az alábbi lista begépelése, a program elindítása után a gép véletlenszerűen szimmetrikus mintákat rajzol a képernyőre.

```
100 PROGRAM "SYMM.BAS"
110 STRING D$(0 TO 1)
120 LET D$(0)=CHR$(32)
130 LET D$(1)=CHR$(142)
140 RANDOMIZE
150 DO
160   LET X=RND(20)+1
170   LET Y=RND(12)+1
180   LET Z=RND(2)
190   PRINT #102, AT Y, X: D$(Z);
200   PRINT #102, AT Y, 41-X: D$(Z);
210   PRINT #102, AT 25-Y, X: D$(Z);
220   PRINT #102, AT 25-Y, 41-X: D$(Z);
230 LOOP
```

Nagyobb méret

Megalomániásoknak ajánljuk a listán látható rutint. A program lefutása után a szerkesztésre használt képernyő mérete a Basic-ből elérhető maximális méretű lesz. Az eredetit a TEXT paranccsal állíthatjuk vissza.

```
100 PROGRAM "BIGEDIT.BAS"
110 CLOSE #0
120 CLOSE #102
130 SET VIDEO MODE 0
140 SET VIDEO COLOR 0
150 SET VIDEO X 42
160 SET VIDEO Y 27
170 OPEN #102:"VIDEO:"
180 OPEN #0:"EDITOR:"
190 DISPLAY #102:AT 1 FROM 1 TO 27
```

Torz karakterek

Ez az ötlet tulajdonképpen az előző elfajzásának tekinthető. Az alapértelmezésű videocsatornát ugyanis négy színűnek állítjuk be, és így igen érdekes (értsd: gyakorlatilag olvashatatlan) karakterek jelennek meg a képernyőn.

```
100 PROGRAM "COLCHARS.BAS"
110 CLOSE #102
120 SET VIDEO COLOR 1
130 OPEN #102:"VIDEO:"
140 DISPLAY #102:AT 1 FROM 1 TO 24
```

Új CLICK

A CLICK hang a billentyűzet gyengességét próbálja ellensúlyozni: biztos érintkezéskor megszólal. Ha a pittyegő hang már kezd unalmassá válni, akkor adjuk ki az

OUT 191,34

parancsot. A CLICK így egy kissé öblösebbé válik.

Sztringek a Pascal-ban - újra

Csak a tavaszi fáradság miatt maradt ki előző számunkból a hasonló című ötlethez tartozó Pascal nyelvű lista. Az ott közölt szöveg továbbra is érvényes, és íme, itt van a hozzátartozó program is:

```
PROGRAM STRING.PAS;
TYPE STR6=PACKED ARRAY[1..6] OF CHAR;
VAR T:STR6;
PROCEDURE STDOUT(VAR X:STR6);
BEGIN
  RA:=CHR(121);
  RBC:=#0006;
  RDE:=ADDR(T);
  RA:=EXOS(8);
END; (of StdOut)
BEGIN
  T:='STRING';
  STDOUT(T);
END. (of program)
```

A sarkon áll

Még mielőtt bárki rosszra gondolna, gyorsan elmondjuk, hogy az egyes editorcsatornák által a státuszszorban megjelenített szám jelentéséről lesz szó. Sokan nem tudják ennek a számnak a jelentését. Nos, az editor tárolókapacitása is véges. Ha már csak 100 bájtnyi üres hely van a csatornában, akkor ezt jelzi ki a gép a képernyő jobb felső sarkában. A szöveg további írásakor ez a kijelzett érték folyamatosan csökken, hiszen fogy a szabad helyek száma. Ha az érték elérte a nullát, akkor ez azt jelenti, hogy az editor legelején álló szöveg eleje elveszett. A WP-vel dolgozóknak már a reflexeikben van a sarok figyelése, mert csak így menekülhetnek meg szövegeik elvesztésétől...

Fütyülés felvételek

Ha csak magnóval rendelkezünk, akkor gondot okozhat az egyes programok elejének megtalálása a szalagon. Készítsük ezért felvételeinket úgy, hogy a SAVE végrehajtása előtt adjuk ki a SOUND DURATION 2E4, PITCH 10

parancsot. Ennek hatására a magnó a program hangjai mellett a fütyüt is felveszi a rögzítés köztes részeibe. Ha az egyes programokat más-más PITCH értéknél rögzítjük, akkor a programba való behallgatással könnyebben tájékozhatunk a program kezdetének helyzetéről.

Villogó sorok

A WP "Line colour" paranccsával átszínezhajthatjuk az egyes sorokat. Mivel az EDITOR: eszköz funkciói mindehol élnek, így ezt a hatást Basic-ben is elérhetjük. A sorok átszíneződését az ASCII 246 kódú karakter elküldése jelzi az editornak, így mindenféle billentyűnyomogatás nélkül is elérhetjük ezt a hatást.

```
100 PROGRAM "FLASHLN.BAS"
110 PRINT AT 5,1:"Ez a sor villog.";
120 FOR I=0 TO 50
125   WAIT DELAY 1
130   PRINT AT 5,1:CHR$(246);
140 NEXT
```

Ha letelt az idő

Számtalan olyan alkalom van, amikor csak egy bizonyos ideig kell egy folyamatnak futnia (pl. várakozás billentyűre, várakozás karaktervételre soros vonalán stb.), utána egy másikra kell áttérni. Az EXOS-nak van egy ilyen időzítője, amelyet Basic-ből is remekül lehet használni. Ezt illusztrálja az alábbi program:

```
100 PROGRAM "TIME-OUT.BAS"
110 PRINT "Nyomj le egy billentyűt!"
120 SET TIMER 5
130 WHEN EXCEPTION USE TIMEOUT
140   LET AS=INKEYS
150   IF AS="" THEN 140
160   PRINT "Megnyomtad idő előtt."
170 END WHEN
180 HANDLER TIMEOUT
190   PRINT "Letelt az idő."
200 END HANDLER
```

NAVY

1988-at írunk, amikor is a DINAMIC SOFT nevű játékprogramokat forgalmazó cég kiad két játékot NAVY MOVES és NAVY MOVES 2 címmel. A spanyol *Ignacio Abril* és *Jorge Azpiri* - aki ugyancsak az Ibériai félsziget szülőtte - programja, grafikája igazán megtetszett a Spectrum tulajdonosoknak, azóta is játszanak e két játékkal, ha ugyan el nem adták a számítógépeket.

A kétezredik év felé közeledünk rohamos gyorsasággal, amikor is egy szigeten néhány tudós közreműködésével az Egyesült Államok államvédelmi hivatala egy vadonatúj pusztító fegyvert fejleszt ki. A fegyver az agy fájdalomközpontjára hat, magas hullámhosszú hang- illetve fényhullámok segítségével, ezek segítségével tudja a kormány irányítani az emberek gondolatait. Már-már a diktatórikus állambatalom kezében van a fegyver, a szigeten lévő tudósok az utolsó simításokat végzik a fegyveren. A demokraták egy csoportja egy kommandóosztást bízott meg az-zal, hogy a szigetet robbantsa fel, semmisítse meg ezt a rémisztő, még az atombombánál is borzasztóbb fegyvert.

Egy tengeralattjáróval indulunk az Egyesült Államok nyugati partvidékéről. A játék első részében el kell jutnunk a szigetre, mely nem lesz könnyű, mert úszó aknáknak, vizimotoros órkok, búvárok, cápák, majd végül egy óriási polip és egy hatalmas kígyó próbálja a tudunkra adni, hogy semmi keresnivalónk errefelé. Pár(száz) óra alatt meg is érkehetünk célunkhoz. A küldetésünk célját tudjuk, de a megvalósítás lehetőségét még nem. A sziget felrobbantásához különböző kódokra van szükségünk. Ezeket a kódokat akkor tudjuk megszerezni, ha lelövünk egy tudóst, odamegyünk és a joystick-ot lefelé húzzuk. Ekkor megjelenik, hogy a tudós miket rejtegetett a zsebeiben. Általában egy *vöröskeresztes csomag* illetve egy *kódkártya* van náluk. Válasszuk a kódkártyát! Egy négyjegyű számot kapunk, jegyezzük meg a kódkártya típusával együtt. (Ez lehet pl. *Maquinas*, *Captain*, *Transmis* stb.) Ezeket a kódokat kell felhasználnunk a termináloknál. (A fent leírt tárgyfelvételi módszer alkalmazásával vehetjük ki a lelőtt katonák zsebéből a löszereket. Ez igen fontos, mert ha elfogy a muníciónk, akkor nem tudunk több kódkártyát felvenni. (Ilyenkor csak egyetlen lehetséges megoldás van: a játék újratekésztése.) Szerezzük meg egy első tiszt ID-jét (*1ST OFFICER*), valamint egy irányító (*TRANSPORTS*) azonosító kódját. Menjünk az egyik terminálhoz (a terminálok zöld monitorok), és írjuk be a legelső parancsot: *STOP ENGINE*. A gép kéri az első tiszt ID-jét. Írjuk be, és ha mindent jól csináltunk, akkor a számítógép terminál elfogadja működésünk (*ACCEPTED*). Ezután írjuk be: *EMERGE*. Szintén az első tiszt kódja kell. Ha minden jól megy, akkor a hajó a felszínre emelkedik. Lépünk ki a terminálból (*END* vagy *FIN*), és menjünk a térképen feketével besatírozott terminálhoz! (Ez meg is van jelölve egy "A" betűvel). Itt írjuk be az *"OPEN DOOR"* parancsot, majd az irányító ID-jét is. Ekkor feltárul egy titkos ajtó. Az utolsó parancs az *"TRANSMITH"*, melyhez szintén az irányító kódja kell. Ekkor a gép megkérdezi a titkos kulcszót: *"OABERBYAMD"*. Menjünk be az ajtón, és lépünk a reaktorhoz! A bomba aktivizálódik. Most már csak meg kell menekülnünk, mielőtt a bomba robbanna!

Végezetül az irányításról annyit, hogy a játékot bármely botkormányal játszhatjuk, a billentyűzet választásakor a nekünk megfelelő billentyűket beállíthatjuk. (Sorrendben a jobbra, balra, fel, le és a tűz gombokat.) Minden irányításválasztás után a gép bekéri a következő billentyűket is: *fegyverválasztás* (a pisztoly illetve a tűzágú között ezzel választhatunk a játék során), *a játék megállítása és a játék újratekésztése*. Reménykedjünk, hogy a játék folytatása, az *ARTIC MOVES* is át lesz valaha írva *ENTERPRISE*-ra.

ROCKSTAR MANAGER

Aki szereti a pop- vagy a rockzenét, az bizonyára örül a *CODE MASTERS* játékának, a *ROCKSTAR MANAGER*-nek. Itt van a lehetőség, hogy kedvenceinket ne csak hallgassuk, epekedve nézzük, hanem menedzseljük is őket. A program a gyengébb műfajok kedvelőinek jó szórakozást nyújthat, viszont igazi, kőkemény rockerek be se töltsék a játékot, mert Lemmy-n kívül csak a nyálásbas zenét játszó énekesek, együttesek találhatók meg a *ROCKSTAR* palettáján. (Igazán készíthetnének egy *Hard Rockstar* vagy egy *Heavy Metalstar* programot, ahol meg-

található lenne pl. a *Guns N'Roses*, a *Skid Row*, a *Megadeth*, a *Mötley Crüe*, a *Queensryche* stb. *Code Masters!* Ha netán olvassátok eme sorokat, itt a lehetőség!

Betöltés után a címképernyőn egy eléggé vulgáris szöveget olvashatunk: *Rockstar ate my hamster!* (Rocksztár bekapta a f...szom! Elnézést kérünk a "bekapta" szó használatáért. *A szerk.*) Miután elborzadtunk az ifjúság új nevelési formáján, jöhet maga a játék!

Először is válasszuk ki, hogy *External Joystick* [I] vagy billentyűzettel akarunk menedzselni. Ezután egy kis bevezető következik, ahol mi és a bal kezünk, *Clide* elhatározzuk, hogy szeretnénk meggazdagodni. De mivel lehet manapság sok pénzt keresni? - ötlük fel a kérdés bennünk. Hát egy rocksztár menedzselésével!

Vonuljunk át a videoterembe, ahol kiválaszthatjuk, hogy hány énekest akarunk menedzselni (1-től 4-ig), majd Clive levetíti nekünk a lehetséges sztárokat a vadonatúj szovjet gyártmányú Sokol videonkon. A palettán a következőket találhatjuk: *Wacky Jacko*, azaz a jó öreg Michael Jackson; az szexis Tina Turnert *Tina Turnoff*-á keresztelve; *Elvin Dwigth* álnéven a kalapjáról híres Elton John-t; Michael Hutchins vadonatúj barát-nőjét, *Kylie Minogue*-t, akit *Bimbo Baggins* néven futtathatunk, és aki egyébként Michael barátunkat tetten érte egy jó meleg ágyban; *Iggy Carrust* névbe burkolózott az Iggy Pop vezéregyénisége; *Holly Johnson* ki tudja miért, de *Dorisssey* néven szerepel; a volt Genesis énekes, *Phil Collins* a keresztnevét *Bill*-re változtatta; a csinos Madonna egy híres focista nevét kölcsönözte; az egyetlen igazi rocksztár *Lumme*, azaz Lemmy a Mötörhead-ből; tizedikként két, számomra ismeretlen leányzót választhatunk; a következő sztárról is ugyan ennyit tudok; Prince-t láthatjuk a képernyőn ezután, majd az elálló fülű és szerény Rick Astley bukkan fel; *Dick Knackered* azt hiszem mindenki számára ismerős, nem más, mint Mick Jagger; a Magyarországon is járt Tracy Chapman következik ezután; majd Freddy Mercury a Queen-ből *Eddy Quicksilver* néven; a világhírű szintetizátormester, *Jean Michael Jarre* pedig *John Michael Jars* néven mutatkozik be nekünk; a TV képernyőjén *Kate Bush* jelenik meg ezután; a következő ismétetlen ismeretlen (ő már nagyon nyálás zenét játszott); *Peter Gabriel* a következő ügyfelünk; a jó öreg Frank Zappa zippzárba burkolózza jelenik meg *Frank Zipper* néven; két ismeretlen után (zenei ismereteim úgy látom a csúcsfokra hágott) a nagy *Rupert Beattlet*, *Paul McCartney*-t mutatja be nekünk Clive; ismét egy ismeretlen (ugye megmondtam!); *Annie Lenox* a vezetéknevét *Smallpoxra* magyarosította; *Brian Adams* a következő; *Steve Wonder* néz ránk napszemüvegén keresztül, s inkognitóját megtartva *Sneezy Wonderful* néven menedzselhető; végül, de nem utolsósorban pedig a *Beatles* dobosát, *Ringo Star*-t láthatjuk.

A kiválasztásnál vegyük figyelembe az énekesek fizetését is, amely egy hétre vonatkozik. Ha több sztárt választottunk, adjunk nevet az együttesnek, majd vegyük meg a hangszereket és a technikai felszerelést. A kínálat: Vadonatúj - £7000, másodkézből való - £4900, agyonhasznált - £2100. Kezdetben 5000 fontunk van. Ha mindezt végigcsináltuk Clive kíváncsian érdeklődik, hogy most mi a fenét csináljunk. Az alábbi menüpontok közül választhatunk:

1., *PRACTICE* (Gyakorlás): Célszerű a játék elején addig gyakorolni, amíg a macskazenét fel nem váltja valami tűrhetőbb muzsika. A gyakorlás 1-5 napig lehetséges egyszerre. A terem bérleti díja 100 font, de vigyázzunk, mert minden péntek fizetésnap. Így a játék elején ne gyakoroljunk sokat, mert különben csődbe megy a cégünk. Gyakorlás közben nem árt, ha néha felépünk valahol, hogy egy kis pénz álljon a házhöz.

2., *GIG* (Koncertezés): A fellépések a legnagyobb pénztermelő források. Kocsmákban (£10/éjszaka), klubokban (£1000/éjszaka), egyetemi klubokban (£2800/éjszaka), koncerttermekben (£7000/éjszaka) és stadionokban (£16500/éjszaka) untathatjuk a közönséget borzalmas macskanyávogásunkkal. Azt hiszem, mondanom se kell, hogy ne a stadionokba menjünk legelőször, mert az a kincstárunk kiürítését vonná maga után. Ezután kiválaszthatjuk, hogy mennyiért adjunk egy jegyet, majd azt, hogy hány napig tartson a turné (2-7 éjszaka). A jegyeket ne adjuk túl drágán, mert ez kis nézőszámot vonna maga után, s így bevételünk is igencsak megcsappanna. A koncertturné alatt a zenészeknek vannak egynéhány hülyeségei: például egy 50000 fontba kerülő lézershow. Ezt megúszhatjuk, ha a turné megkezdése előtt adunk neki egy kis ajándékot. (Ha ezt nem adjuk, és a show-t sem fogadjuk el, akkor az énekesünk otthagyja a fogát

egy atomháborúban. Hogy ez hogy jutott a programozó Colin James eszébe? Elég fáradt napja lehetett...

3., **PUBLICITY** (Reklám): Az együttes körüli botrányokat a "The Stun" című szennylapban jelentethetjük meg. A reklám ingyenes, viszont néha rosszul sül el a dolog. Ha a címdalton egy félpucér hölgy látható, akkor nem volt egy jó reklámunk. Akkor sincs szerencsénk, ha az együttes egyik tagjának tragikus halálát a Stun-ból tudjuk meg. (Ezt elkerülendő: ismét adjunk kis sztárunknak ajándékot.) A játék céljának eléréséhez erre a menüpontra semmi szükség nincs.

4., **GIFTS** (Ajándékok): Már sokat emlegettem az ajándékok fontosságát, de mit is adhatunk a zenészeknek. Képregényt (£1), meghívhatjuk vacsorázni (£50), meghívhatjuk egy felesre (£100 - lehet, hogy ebből az egy felesből több is lett?), vehetünk nekik ruhát (£5000), ékszereket (£10000) és kocsikat (£20000) ajándékba.

5., **RECORD** (Lemezkészítés és kiadás): Ez a menüpont csak akkor jelenik meg, ha elfogadtuk valamelyik lemezcég ajánlatát. (Erről később lesz szó.) Ha még nem készítettünk lemezt, akkor először kiválaszthatjuk, hogy milyen minőségű, méretű stúdiót választunk (2 Track - £2000; 4 Track - £5000; 8 Track - £10000; 16 Track - £20000; 48 Track - £50000). Természetesen a legjobb választásunk ki! Aztán beülünk a stúdióba, és máris készül a lemez. (A dalcímeket magunk is megadhatjuk, ezért célszerű olyan címeket adni, melyekből tudjuk majd, hogy milyen zeneszámmal is állunk szemben. Ez lehet *Mega Noise*, *Classic Hit*, *Monster Number*, *Super Tune* stb.) Ha már készítettünk lemezt, akkor kiadhatjuk a nagylemezt, illetve kislemezeket is. Ha kislemezt adunk ki, akkor kiválaszthatjuk, hogy melyik alkotásunk kerüljön rá, és Clive megkérdezi, akarunk-e videoklipet készíteni a kislemezhez. (Általában klip nélkül is felkerülhet egy kislemez a hitlistára, de ennek igen kicsi az esélye.) Ha készítünk hozzá videoklipet, akkor választhatunk a rendezők közül (Steven Cheeseburger - £3000000; Cecil Bidet Mills - £1000000; Busby Berserkely - £50000; Wrihy Scott - £10000; Julian Temple Meads - £5000 és végül a mi Clive-unk - £5.), kiválaszthatjuk a helyszínt (Bahamas - £50000; Hollywood - £20000; Pinewood - 10000; Cricklewood - £50), és a videoklipp "szereplőit" (Hosszúcombú modellek korbácsokkal, híres rajzfilmfigurák, olajfoltos szépfűk munka közben, dühösen lesújtó kalapács, mutáns boszorkányok és denevérek, koncert-show felvétel). Természetesen a zene típusához illő szereplőket válasszunk!

Ha már kiadtuk a lemezeket, akkor kezdődik az igazi játék, ahol a cél az, hogy minél többször felkerüljünk a listák élére, és így csináljunk platinalemezeket. A lemezek a bal alsó sarokban vannak, és ha kigyullad egy, akkor csináltunk egy platinalemezt. Erre összesen egy évünk van, így a játék elég nehéz. Persze azért ez még nem minden. A játék közben megszólalhat a telefon is, amelyben minden marhaságot mondhatnak. Például:

- Egy vállalat szponzoriként lép fel, reklámozni szeretné magát. Ezek általában 100000 fontot ajánlanak fel, de ne fogadjuk el, csak ha nagyon a csőd szélén állunk, különben ha a vállalat tönkremegy, akkor az árthat a hírnevünknek, mely a koncertek látogatóinak megfelelődsét jelenti. Ilyen cégek lehetnek a *Bullwiser Beer*, a *Lievie Jeans* és a *Joka Cola*.

- Egy hanglemeztársaság ki szeretné adni művészetünk gyűmölcsét. (Ekkor jelenik meg a **RECORD** menüpont.) Négy hívt kaphatunk. Az első 5000 fontot és 3% részesedést ajánl fel, a második 5%-ot és 10000 fontot, a harmadik 30000 fontot és 15%-ot, az utolsó pedig 50000 fontot és 20%-os részesedést. Ez utóbbit fogadjuk el, s addig gyűjtsünk egy kis pénzt. (Pár milliót!)

- Ellopták a lemezőnk, és csak egymillió font ellenében adják vissza. Ekkor két lehetőségünk van: Vagy kifizetjük a "váltságdíjat", vagy csinálunk egy új lemezt. Időhiány miatt érdemesebb kifizetni a követelt összeget.

- Egy tajvani városkában másolják a lemezőnk. Ekkor vagy veressük szét őket (*Send the boys around*), vagy pereljük be a kleptomániás céget a bíróságon. Mindkettő pénzbe kerül...

Ez lett volna hát a **ROCKSTAR MANAGER**. A játék kitűnő humorával biztos nem lesz unalmas. Viszont a játék legfőbb problémája az, hogy a vasárnapi hitlistákat mindig végig kell nézni, még akkor is, ha nem vagyunk rajta. Ez rengeteg időt elvesz, és unalmassá teszi a játékot. A Devilsoft javítása viszont ötletesen gyorsabbá teszi a játékot. Akiknek ez a verzió van meg, az bizonyára élvezettel fog menedzselni. Akinek nem, az is...

SAVAGES

Egyszer volt, hol nem volt, volt egyszer két programozó - *David Perry* és *Nick Bruy* -, akik úgy gondolták, igazán születetne Spectrumra is egy jó grafikával rendelkező játék. Így hát leültek hön szeretett gépük mellé, és elkezdtek a billentyűket nyomogatni. Ki is kerekedett belőle a három részből álló **SAVAGE** (Vadember) sorozat. Nem volt más hátra, csak egy szoftverházat kellett keresni, amely piacra dobja a játékot. Ekkor bukkantak rá a "PROBE SOFTWARES" cégre, amely hamarosan megjelentette az első részt a Spectrumosok nagy öröme. Folyamatosan adták ki a következő két részt is, melyek hamarosan a toplisták élén találták magukat. A programozók is, és a szoftverház is megkapta a saját kis pénzét. Így hát mindenki jól járt, s boldogan élnek, míg tönkre nem mennek.

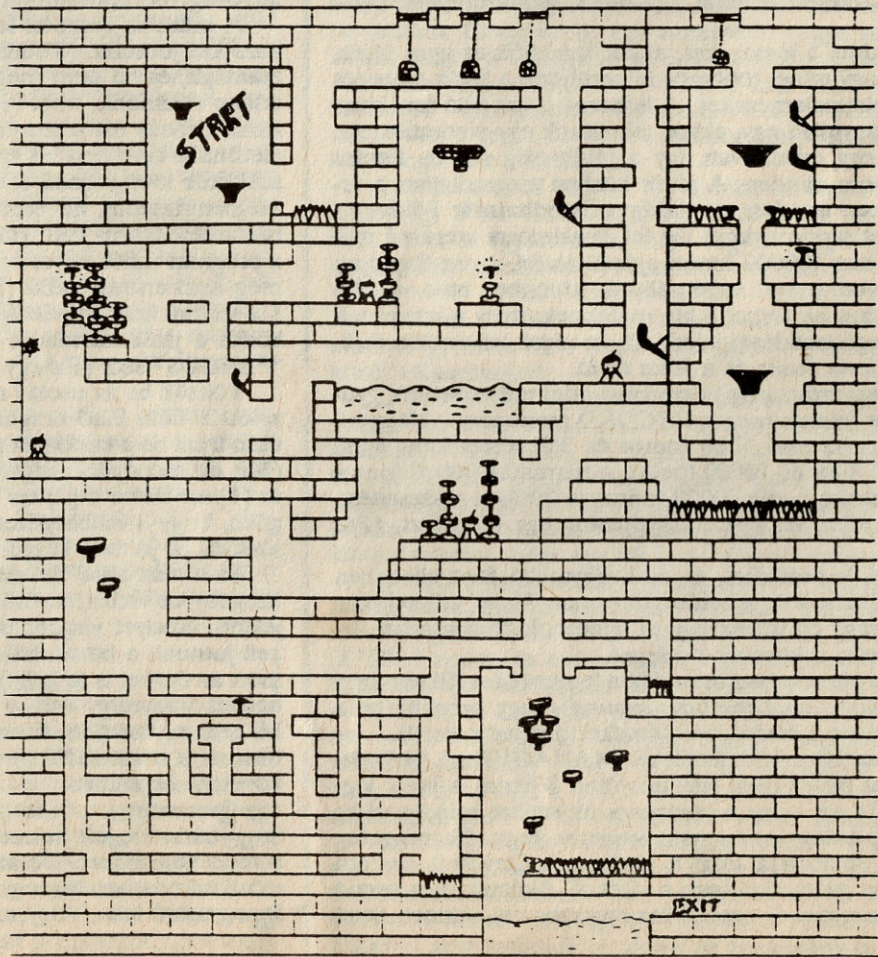
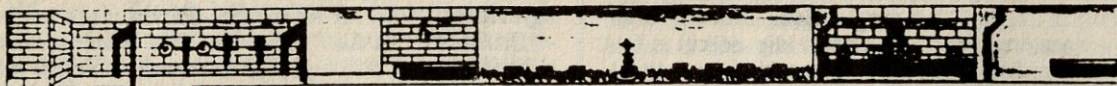
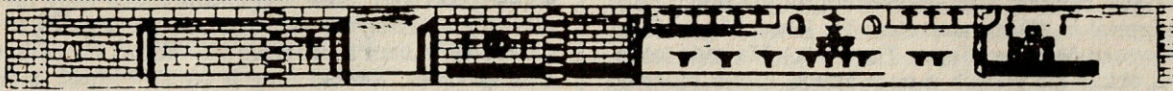
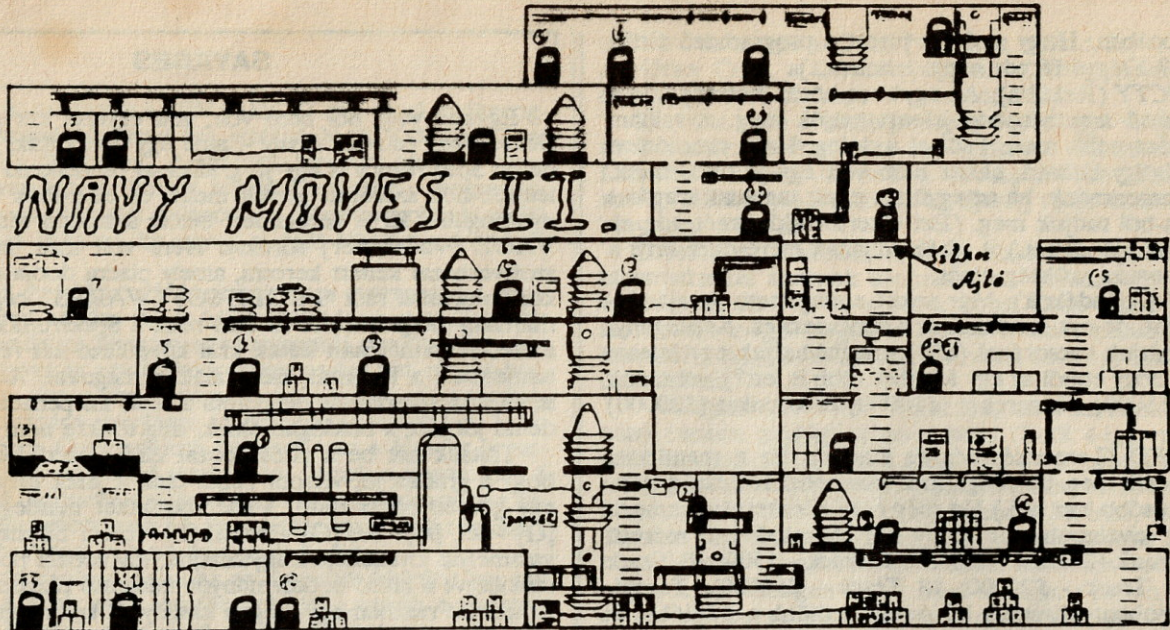
Töltsük hát be a játéksorozat első részét! Máris láthatjuk, hogy a grafika jó, viszont zenét csak a játék elején hallhatunk egy gombnyomás után. Vademberünket mindegyik részben a [Q] - fel, [A] - le, [O] - balra, [P] - balra billentyűkkel, illetve valamelyik joystickkal irányíthatjuk, lövöldözni pedig a [Space]-szel illetve a külső botkormányon található tűzgombokkal lehet. Itt az első részben egy idegen kastélyból kell megszöknünk. Ezt akadályozza néhány madár, no és a fura szörnyek sem éppen a segítségüket szeretnék felajánlani. Óriási nagy tüzek akadályozzák még a kastélyból való kijutást. De segítség is a rendelkezésünkre áll, hiszen a program különböző üzeneteket ad nekünk, mint például "Keress meg a liftet", "Másik úton menj" stb. Igaz, amiben segítené, arra saját magunk is rájövünk, annyira egyértelmű. Viszont az is igaz, hogy a játék közbeni dicséretetek - "Brillians", "Kitűnő" - növelhetik a reményeinket. Ha sikerült kijutni a kastélyból, a gép jó szokásait elfelejtve mindenféle gratuláció nélkül csak annyit közöl velünk, hogy megcsináltuk. Mintha mi olyan ostobák lennénk, hogy erre nem jönnénk rá magunktól! Ekkor tudjuk meg a játék második részének a kódját: "SABATTA".

A második rész betöltésekor az [S] billentyű lenyomása után írhatjuk be az előző kódot, melynek során három életet kapunk. A játékot természetesen kód nélkül is elindíthatjuk, de egy élettel nem hiszem, hogy sokra jutnánk. Az indítás a [K] illetve a [J] gombok leütésével történik, attól függően, hogy billentyűzettel, avagy botkormánnyal szeretnénk játszani.

A második részben kint vagyunk a kastélyból, viszont még haza kell jutnunk. (Tudniillik fellőtték szegény vademberünk pizsamáját, és ha nem megy haza, vadanyuci elnászpángolja.) Erdőben rohanunk, ahol furcsa fák "mosolyognak" ránk. Vigyázzunk, nehogy túl közelről nézzünk a szemük közé, mert az egy életünkbe kerülhet! És ez még mind semmi! Különböző szörny-szülőttek lövöldöznek ránk, akiket célszerű egy pontos lövéssel megsemmisíteni, ha haza akarunk jutni. A játékban összesen háromféle szörny van. Ha valamelyik fajtájukat kiírtottuk, akkor a program ismét udvariasra vált, és gratulál, de azt is közli, hogy még nem értünk haza. (Hűűű, mit kapunk mi ezért otthon!?) Ha viszont már mindenkit kinyírtunk, akkor kapunk egy újabb kódot a játék harmadik részéhez. Ez pedig nem más, mint a "FERGUS" szó. (Ez egy skót férfinév.)

Töltsük be az utolsó részt is! Itt már felfedezték a menükezelést a fiúk. Első dolgunk az, hogy a [3] billentyű lenyomása után írjuk be a kódot, és így itt is megkapjuk a három életünket. (Mit ne mondjak, rengeteg!) Visszakerülünk a menühöz, ahol az [1]-sel illetve a [2]-sel indíthatjuk a játékot szintén attól függően, hogy a billentyűzetet akarjuk e koptatni, vagy inkább a joystick áll jobban kézre.

Ez a rész kissé nehezebb, mint a többi, de élvezetesebb is! Szegény kis vademberünket már hazavittük, de az erejét egy sas jelenti, amelyet viszont börtönbe vetettek. Így ezzel a sással ki kell jutnunk a börtönből. De nem elég a kijáratot megkeresni, mert az őrköt is le kell fizetni valamivel. A börtönben összesen három aranytárgy van, a börtönkapu rácsánál pedig három ór. Mi sem egyszerűbb, fizessük le őket az arannyal, és máris kijutottunk a börtönből. Igen, csak hogy a húséges szörnyek ide is követtek bennünket (esküszöm, eladom a kutyámat, és veszek egy ilyen szörnyet. Húséges is, erős is, bátor is, és nem hinném, hogy bárki idegen be merne jönni a házba.), ezek is nehezítik a menekülésünket. De azt hiszem a mellékelt térkép, illetve az előző két részben összegyűjtött tapasztalatok segítségével ez pofonegyszerű lesz. Vagy mégsem...?



5
4
3
2
1

Postafiók 334

Bognár Balázs felháborodott hangon írja Petőházáról: ... a mélyen tisztelt olvasóközönség úgyszólván lenézi a gépet, és minden erejével azon van, hogy a gépet a legsötétebb oldalról világtítsa meg. Ezt az állításumat arra alapozom, hogy a "Postafiók 334"-ben közölt levelek enyhén szólva pesszimisták, és ANTIPRISE-osak. Ebben a kérdésben nem értem a lap álláspontját, hiszen az a maroknyi géptulajdonos nem azért veszi meg a lapjukat, hogy lépten-nyomon lehordják a gépet, hanem azért (legalábbis én így gondolom), hogy egy kis biztatást kapjon, és legalább ebben az egy lapban foglalkozzanak a gép előnyeivel, és optimistán álljanak hozzá a lap szerkesztéséhez. Itt konkrétan az utolsó számban megjelent SOUND: elemzésre gondolok, amely valóságosan leócsárolja a gépet. Mellesleg jegyzem meg, hogy aki a gép hanghatásait "pittyegő szoftver"-nek nevezi, annál két lehetőség áll fenn. Először: az illető még életében nem látott, és nem hallott ENTERPRISE-t a valóságban, hanem csak a szomszéd-tól hallott róla. Másodszor: Az illető nem igazán becsüli a gépet, a kínálókozó lehetőségeit, tulajdonságait nem az azonos kategóriájú gépekhez viszonyítja (C16, C+4, C64, Spectrum), hanem egyszerűen kijelenti róla, hogy rossz, szinte használhatatlan. Az ilyen kijelentések egy felelős szerkesztőtől eléggé feleltelenül hangzanak, hiszen a levelekre adott válaszokban éppen neki kellene állást foglalnia a gép tulajdonságai mellett (jó tulajdonságai mellett). Mellesleg megjegyzem, hogy nincs olyan lap ma Magyarországon (ezen kívül), amelyben nem védik meg meg a saját gépüket az olvasók leveleivel szemben.

A másik megjegyzésem az, hogy az ENTERPRISE nem szintetizátor, amely minden hangot előállít, valamint nem videofilmek elkészítésére alkalmas eszköz, hanem egy olyan számítógép, amely ugyan rendelkezik apró hiányosságokkal, de még ennek ellenére is kimagaslók az előbb felsorolt gépek közül. Szerintem az egyetlen igazi vetélytársa az Amiga, ÉS NEM ÉRTEM, HOGY MIÉRT NEM LEHET EZT BEVALLANI!

Kedves Bognár Balázs! A mélyen tisztelt olvasóközönséggel Önnek ugyanaz a baja, mint ami régebben volt a pártfunkcionáriusoknak a néppel: az emberek mindig csak a problémákat vették észre, mindig csak azokról beszéltek; a jól működő dolgokról soha sem szövegelt közülük senki. Mi, a lap szerkesztői, nem tehetünk arról, hogy ilyen "sötét" leveleket kapunk, abból válogatunk, amiből tudunk.

Az országban az Enterpress szerkesztői a legoptimistább emberek, máskülönbben nem vágtak volna bele mindenféle öngyilkos lapkiadási örületekbe. Eszünk ágában sincs a gépet lépten-nyomon lehordani! Amit viszont Ön sem tagadhat: sajnálatos tény, hogy a géptulajdonosok egy része letargiában szenved, kétségek között vergődik: korábbi kellemetlen tapasztalatra támaszkodva várjon-e tovább az állóvízben a forgalmazó(k) ígéreteinek megvalósulására; avagy gyorsan, amíg lehet, túladjon a gépén, és benevezzen egy Atari 520ST-re, Amigára stb. Nagy szükségük van ezeknek az embereknek a biztatásra, egyebek között ezt igyekszik minél teljesebben megtenni az ENTERPRESS. Lehet, hogy hihetetlenül és dicsekvésként fog hangzani az Ön számára, de sok gép menekült meg az eladástól az ENTERPRESS megjelenése következtében...

A SOUND: témát (figyelmes olvasás után kiderül...) szerzője (aki nem azonos a felelős szerkesztővel) vitaindítóan szánta, így a beérkező válaszokra ő fog reagálni. (Csak úgy megemlítem, hogy a gép fejlesztői a soros vonal kezelésén kívül szintén a hanggal elégedetlenek leginkább, nem véletlenül kerül az új gépbe Midi interfész.) A "pittyegő"-höz pedig még hozzátehetjük: murtyogó, zuborgó, nyunyorgó, hortyogó, kartyogó, sistergő, repedt fazék stb. Ha Ön elégedett a közkézen forgó programok hangzásával, akkor lelje örömet bennük! Én személy szerint (szerkesztőtársaim jelentkezik) kitartok az igényes zenék, effektek, zörejek mellett. ENTERPRISE-t egyébként még tényleg nem láttam a valóságban, de van egy ismerősöm, akinek van egy másik ismerőse, akinek a gyengénlátó nagymamája egyszer már megfigyelt egy ilyet egy elmosódott fényképen.

A gép tulajdonságai mellett nem a szerkesztőknek kell kiállniuk, ez a gyártó, vagy a forgalmazó feladata (lenne...). Az En-

terpress indításakor fő szempont volt, hogy korrekt lapot csináljunk, ezért nem hisszük, hogy a gépet meg kellene védenünk az olvasók leveleitől: mindenkinek joga van véleményének kifejtésére. Hogy ez másoknak tetszik-e, vagy sem, az az illetőtől függ. Viszont ez az illető - például Ön is - kifejtheti ellenvéleményét, az ENTERPRESS erre mindig lehetőséget fog adni!

Soha nem szerettem a gépek összehasonlítását, mert mindig alapvetően értelmetlen, véget nem érő viták kerekedtek belőlük. Ha csak az alapadottságokat vesszük figyelembe, akkor az ENTERPRISE tényleg jobb a felsorolt gépeknél. De ha arra gondolunk, hogy például C64-re hány ezer jobbnál-jobb felhasználói program van, akkor a mérleg mégiscsak a C64 javára billenne, hiszen az végeredményben jobban tudja szolgálani igényeimet. Nem hiszem, hogy az Amigának ellenfél lenne az ENTERPRISE, bár ez utóbbiból tényleg elő lehet "amigás" hatásokat csiholni. Nem szeretném, ha ezt az okfejtést is "antiprise"-osnak venné, egyszerűen arról van szó, hogy racionálisan kell gondolkodni. De sürgősen abbahagyom, hiszen Ön a végén még a fejemet veszi. Az ENTERPRISE tehát mindenkinek az, ami; nekem például (szerkesztőtársaim ismét jelentkeznek) a szívem csücskét jelenti.

Szekeres Tamás Veszprémből írt, hosszú leveléből csak egy rövid részt idézünk: ... Kellene egy megszakítás-programozásról szóló cikk. Én nem találok a scroll rutinhoz a karakterkészlet kezdőcímét.

A megszakítások kezeléséről az assembly sorozatban lesz majd szó, és addig is valószínűleg jelenik meg ilyen témájú rövidebb írás. Egyébként lapunk legelső számában a "Percenként hatvan keretszín" című anyag már foglalkozott ezzel a témával. Második kérdése nem egészen világos: Miért van szüksége a karakterkészlet kezdőcímére scroll rutin írásához? Valószínűleg futó feliratok készítésére gondol. A karakterkészlet kezdőcímével kapcsolatos tudnivalókhöz hozzájuthat, ha elolvassa az LPT sorozat második részét.

Dudás István, Törökbálint: Három éve vagyok ENTERPRISE tulajdonos, és főként programozásra használom a gépet. Az utóbbi másfél évet a gépi kódú programozás rejtelmeinek megismerésére fordítottam, és kisebb-nagyobb rendszerek fejlesztésének gondolatával barátkozom. Most egy színvonalat elérő szövegszerkesztő fejlesztésével foglalkozom.

Pár hónapja vagyok lemezegység tulajdonos, és természetesen fejlesztéseim is effelé orientálódnak. A szövegszerkesztőhöz szükség lenne alapvető lemezműveletekre, amit az EXDOS programozása foglal magába. Próbáltam felhajtani megfelelő szakirodalmat az EXDOS-szal kapcsolatban, de csak a parancsok használatával és szintaktikájával foglalkozó irodalmat találtam. Olvastam a múltkorai számban, hogy nemsokára közreadják az EXDOS közvetlen kezeléséről a tudnivalókat. Ha jól számolom, akkor ez leghamarabb a júliusi-augusztusi számban jelenhet meg először. Mivel gimnazista vagyok, számomra a nyári szünet lehetne a legalkalmasabb időszak, ezért ezekre az információkra most lenne szükségem. Ha lehetséges, legyenek szívesek velem megosztani ezeket az ismereteket, vagy adják meg a kapcsolódó szakirodalmat jegyzékét.

Nagyon örülök, hogy ilyen komoly programon dolgozol. Lehet, hogy szövegszerkesztő elkészítése után végre elfelejthetjük a WP-t? De jó is lenne...

Láthatod, hogy az EXDOS ismereteket még nem közöltük, a szerzője dolgozik az anyagon. Ha kíváncsi vagy a véleményemre, akkor én azt javaslom, hogy hagyj a csudába az EXDOS direkt kezelését! Neked az EXOS 26-al aktivizálható DOS parancsok is elegendőek, egy szövegszerkesztőnek nincsenek összetett fájl- és lemezműveletei. Direkt kezelésre majd akkor lesz szükséged, ha mondjuk a Norton Commander-hez, a Disk Doctor-hoz, a Speed Disk-hez stb. hasonló programokat akarsz majd az EP-re adaptálni (természetesen Peter Norton megkérdezésével, nehogy koldusbotra juttasd e fontos piac megkaparintásával), vagy esetleg virtuális tárkezelést próbálsz a gépre erőltetni (buliból), és valamilyen szegmentált lapcímezést megvalósító rutint kell írnod. A szakirodalom jegyzékét azért

nem adom meg, mert a címek birtokában sem valószínű, hogy megkaptad őket. Egyszerűbb, ha megvárod a mi írásunkat.

Sípos Jenő budapesti olvasónk írja: *Azzal a kéréssel fordulok a szerkesztő úrhoz, hogy egy kis segítséget kapjak gépelés szempontjából. Kérésem a következő: Van néhány játékprogramom és ASMON-om is, de nem tudom, hogy az örökéletkódokat miképpen tudnám beírni a gépembe. Szeretném, ha az unokámnak is megtudnám ezt mutatni. Két éve vettem a gépet, és eddig nem tudtam elképzelni azt, hogy 68 éves fejjel előbb nekem kell megtanulnom a gépet kezelni; ha meglátogat az unokám, meg tudjam neki tanítani amit kell.*

Az örökéletkódok bevitelére nagyon egyszerű. Szögletes zárójelek közé tettük azokat a billentyűket, amelyeket az egyes parancsokhoz (betöltés, módosítás, mentés) le kell nyomni. Természetesen az őket követő számokat is nekünk kell bebillentyűzni, és a magnó kezelése is ránk marad. Nagyon fontos tudnivaló, hogy az általunk közölt örökéletkódok csak akkor élenek, ha a betöltés után az Ön gépe ugyanazt a "Last address" értéket jeleníti meg, mint amit mi adtunk meg.

Szabó Zoltán Budapestről az alábbi kérdéssel fordult hozzánk: *... Hogy lehet az, hogy az ENTERPRESS előfizetési csekket együtt adtuk fel a szomszédal a postán, és ő már megkapta, én pedig a színét sem láttam még. Érdekességként egy másik szomszédom sem kapta még meg.*

A magyarázat egyszerű: Önnek is és a másik szomszédjának is az első szomszédot kellett volna megbíznia a csekkek befizetésével! Úgy látszik neki szerencséje van a Postával. Komolyra fordítva a szót: Sajnos, nem ez az egyetlen negatív jelenség lapunk terjesztésével kapcsolatban. Azok az előfizetőink jutnak biztosan az ENTERPRESS-hez, akik közvetlenül a Mátrix Kft.-nél rendelik meg azt. Egyelőre nem tudunk mást mondani,

mint azt, hogy kiadónk majd rákérdez ügyére a Posta illetékésénél. És még valami: A beküldött scroll rutinja - feltehetően a "kézi nyomtatás" miatt - nem működik. Mivel irányítószámát elfelejtette megírni, arra kérjük, hogy elvárásainknak megfelelő formában küldje be ismét a programot.

A szív az élet motorja . . .

SZIV

1991. szeptember 4 - 5 - 6.

MEGYEI MŰVELŐDÉSI
KÖZPONT
Székesfehérvár, Liszt F. u. 1.

Székesfehérvári Informatikai Vásár

Engedje, hogy bekapcsoljuk a vérkeringésbe!

MÁTRIX Kft.

mikrovilág

Az ENTERPRESS előző számai korlátozott példányszámban még megrendelhetők a kiadó címén (MÁTRIX Kft. 8000 Székesfehérvár, Dózsa Gy. tér 10.), vagy megvásárolhatók a Műszaki Könyvruházban (Bp. VI. ker. Liszt F. tér 9.) és a FÓKUSZ Könyvruházban (Bp. VII. ker. Rákóczi út 14).

Tisztelt Olvasóink!

Arra kérjük Önöket, hogy utórendeléseiket és megrendeléseiket ne a szerkesztőség, hanem a kiadó (Mátrix Kft.) címére küldjék, mert így sokkal gyorsabban juthatnak hozzá kedvenc lapjukhoz. Köszönjük!

A szerkesztőség

Apróhirdetések

A legjobb lemezkezelő program a VIGADOS! Már 850 KB-ja formázza a lemezt! Teljes menüvezérlés, ablaktechnika, egértámogatás! Ára: 300 Ft + lemez (50 Ft) + 10 oldalas leírás (50 Ft). Küldött lemezre is lehet, a leírás nem kötelező.

Megrendelhető: VICSOTKA GYULA, 2143 KEREPESTARCSA, Pf. 41.

Nyomatóvásárl

- A/4, A/3 papírméret
- traktoros és frikciós papírtovábbítás
- LQ nyomtatás

Ára: csak 8000 Ft! Érdeklődjön a Mátrix Kft. címén (8000 Székesfehérvár, Honvéd utca 8. Telefon: (22) 12-619)

Eladó memóriabővítés EXDOS lemezillesztő 8500-, 3.5"-os, 720KB-os új lemez meghajtó 7500 Ft-ért.

Bozai Gábor, 8000 Székesfehérvár, József utca 70/A. fsz. 1. Tel.: (22) 10-665

LEVELEZÉS

A géppel kapcsolatos témákban levelezniél:
Bognár Balázs, 9443 Petőháza, Bartók Béla u. 11.

Budai Attila, 2119 Pécel, Várhegy út 12.

Feczkó Krisztián, 8200 Veszprém, Ányos u. 1/3. Tel.: (80) 29-493

ENTERPRISE klubok,

ENTERPRISE köré csoportosult baráti társaságok,

ENTERPRISE témákban levelezni kívánók

címeit várjuk!

A Mikrovilág minden számában két oldalnyi terjedelemben foglalkozik ENTERPRISE-os témákkal.

hirdetések, felhívások

HIRDETÉSFELVÉTEL

Az apróhirdetések ára: 1 Ft karakterenként. A szöveget és a befizetést igazoló nyugtát (rözsaszín postautalványon) az alábbi címre kérjük elküldeni:

MÁTRIX Kft.

ENTERPRESS

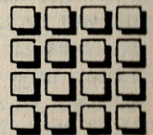
8000 Székesfehérvár

Honvéd utca 8.

Bankszámlaszám: OTP 679-022096-9

Közületi hirdetőknak kívánságra hirdetési árajánlatot küldünk.

Megjegyzés: a nem saját fejlesztésű szoftverek másolásával foglalkozó üzletek hirdetéseit nem áll módunkban elfogadni.



Tisztelt leendő Szerzőtársak!

Kérjük Önöket, hogy anyagaik elkészítése, beküldése előtt feltétlenül kérjenek ingyenes tájékoztatót a feltételekről, a szerkesztőség által felállított tartalmi és formai elvárásokról.

Az érdeklődők leveleit levélcímünkre várjuk:

ENTERPRESS

1399 Budapest, Pf. 701/334