

ENTERPRESS

KÉTHAVILAP AZ ENTERPRISE SZÁMÍTÓGÉPEK FELHASZNÁLÓINAK

Kis ügy

— avagy gondolatok vállalkozásról, jogról, demokráciáról —

Biztosan emlékeznek arra a — lapunk 92/2 számában megjelent — cikkre, amelyben egy kedves olvasónk, Koronkai István ötletén felbuzdulva elhatároztuk; alapítvány segítségével próbáljuk megteremteni azokat a hiányzó pénzüsségeket, amelyek a lap életbentartásához feltétlenül szükségesek.

A beadvány elkészült, az ideiglenes számlát megnyitattuk és azt hittük, hogy ettől kezdve nincs is más tennivaló, mint bízni a jószándékú adományozókban.

Nem ez történt. Nemsokára kaptunk egy barátságos levelet a Fejér Megyei Bíróságtól. lme:

„Fejér Megyei Bíróság,
Pk. 61.054/1992/2. szám.

Végzés

A megyei bíróság az Enterpress Alapítvány nyilvántartásba vételére irányuló kérelmet a Pp. 95. § alapján hiánypótlásra visszaadja és felhívja az alapítót, hogy azt a végzés kézhezvételétől számított 30 nap alatt, — elutasítás terhére mellett — az alábbi hiányok pótlásával terjessze vissza:

1.1 Az alapítványi vagyon a célokhoz viszonyítva kirívóan alacsony, ezzel az alapítvány céljának megvalósítása nyilvánvalóan nincs biztosítva.

Ez legalább 50.000.- Ft-ra kiegészítendő. Enne elkülönített megléte is igazolandó is /bank/. Ellenkező esetben a nyilvántartásba vételt meg fogja tagadni.

2.1 Az alapító okiratban a vagyon felhasználási módja bővebben leterendő.

3.1 A kuratórium három tagú. Kettő az alapító vezetője. Eszerint az alapítónak a kuratóriumban meghatározó szerepe lenne. Ez azonban nem lehet. A kuratórium tagsága másként jelölendő ki.

Székesfehérvár, 1992. március 30.

Dr. Szijártó Lajos sk.
bírószági tanácsos

Tanulságos levél. Érdemes is elemezni mert lehet hogy több van benne, mint ahogy azt egy ilyen kis ügy alapján első látásra gondolnánk.

— Nézzük mindjárt az első kifogást:

„...Az alapítványi vagyon a célokhoz viszonyítva kirívóan alacsony...az alapítvány céljainak megvalósítása nyilvánvalóan nincs biztosítva.”

Nna! Három éve csináljuk az Enterpress-t, elköltöttünk rá cca. 500.000 Ft -ot és kb. 1000 munkórát. (Óvatosan számolva 200 forintjával ez csaknem ugyanannyi, mint az előző, s nem nagy művészet a két összegből kihozni az egymilliót, ha egyáltalán mérhető az a szellemi tőke, amely az újságban megtestesült szerencsétlen kis országunk ha nem is lassan, de biztosan inflálódo forintjaival).

Nos, akkor Dr. Szijártó Úr első nekifutásból, lazán kinyilvánítja hogy céljaink megvalósításához — s e r i n t e — az alapítványi összeg nem elegendő és ez ráadásul szerinte még nyilvánvaló is!

Első felindulásomban felhívtam telefonon a fent említett illetőt és megpróbáltam esetelni azt az egyszerű ténnyt, hogy én nem újság szerkesztési üzletágbeli konzultációt kértem a tisztelt bíróságtól, hanem pusztán azt az egyszerű dolgot ami tudomásom szerint ebben az országban minden ivarképes magyar állampolgárt megillet, neve-

zetesen, hogy j e g y e z z é k be az általam meghatározott célokra az általam finanszírozott eszközökkel az Enterpress alapítványt.

Nem addig a' mondá Ó a telefonba. A pénz mindenképpen kevés, „nem szoktak (!) ilyen kis összeggel bejegyezni alapítványt.”

Idézem tanulságul — milyen messze lehet, no nem földrajzilag, hanem szellemileg Budapest Székesfehérvártól — dr. Almásy Mária bírónő fenti témában adott nyilatkozatát a SANSZ c. újság 1991. augusztus 3-i számából: „Meghatározza-e jogszabály bármely alapítvány indulótőkájének kötelező összegét? — Nem...” (id. SANSZ, az alapítványok lapja, 1991. augusztus 3. oldal 3. bekezdés.)

Akkor mit kell kiegészíteni és mennyire??

De menjünk tovább!

Második kifogás (bocs; pont):

„bővebben” (?). No comment.

Harmadik kifogás:

„A kuratórium három tagú.”

(Hopp, ezt megértette.) „Kettő az alapító vezetője.” (Ezt is! Menni fog ez. Csak így tovább!) „Eszerint az alapítónak a kuratóriumban meghatározó szerepe lenne.” (Naná, de most figyeljenek, ez a legjobb rész:) „Ez azonban nem lehet.”

Csak így. Nemes egyszerűséggel, sallangmentes stílusban. Semmi ezért, meg azért, vagy amazért, vagy mert tiltja a törvény, vagy mert ebben az esetben eleve, csipőből gazembernek minősül az, aki ilyet tesz.

Nem. Egyszerűen csak így: „EZT AZONBAN NEM LEHET!!!”

Lássuk, hogy mit is mond erről a dologról a fent idézett újságban dr. Almásy Mária bírónő:

„— Lehet-e a kuratórium tagja az alapító? — A Legfelső Bíróság állásfoglalása kimondja, hogy az alapító ne legyen a kezelőszerv tagja...A bírói gyakorlatban a kuratóriumot illetően elfogadott, hogy ha például egy öttagú kuratóriumban három személy külsős és kettő alapító, a bejegyzés megtörténik.”

„a Legfelső Bíróság állásfoglalása.” Itt tehát már nem a törvényről, hanem annak valakik általi értelmezéséről beszélnek! Nem egészen ugyanaz, de mindegy. Szél ellen... Mikor is született ez az állásfoglalás:

„Egy éve érvényes már ez a határozat.” És azelőtt? Akiket korábban bejegyeztek azok mind szentéletű koszorúslányok voltak, az azutának pedig nyilvánvalóan becsutelen gazemberek. Valahogy így? Kezdem rosszul érezni magam a saját bőrömben. Mert ugye: „... az alapítványiokhoz fűződő kedvezmények — például az, hogy levonható az adóalapból — számos kiskaput kínálnak az alapítók számára.”

Tehát ez az igaz ok. Ezzel véget is ért volna tisztavérűségű alapítványunk sorsáról szóló beszámoló.

Engedjenek meg nekem azonban még valamit. Nevezetesen azt, hogy megköszönjem mindazoknak akik az újságban megjelent alapítványi felhívást megtisztelték hozzájárulásukkal. Nem voltak olyan sokan (sajnos, hiszen arra számítottunk, hogy legalább az Enterpress előfizetőinek nagyrésze hajlandó erre az áldozatvállalásra) hogy ne tudnám felsorolni őket:

Kustos Endre 300.-Ft
Blasko István 500.-Ft
Bíró Imre 150.-Ft
Koronkai István 200.-Ft
Somlóvári László 300.-Ft
Köszönjük a kiadó nevében:

Juhász István ügyvezető

TARTALOM 92/4

KURZUS

Assembly 12.	2-3
A Pascal 12.	4-5
Az IS-LISP 2.	6

PROGRAMOZÁSTECHNIKA

EPSON kompatibilis nyomtatás	7-8
------------------------------	-----

HARDVER

Kooprocessort az ENTERPRISE-ba!	9
---------------------------------	---

TESZT

Programozási verseny	10-11
----------------------	-------

TIPPEK-TRÜKKÖK

Gyorsítás, fejbeállítás	11
-------------------------	----

KÖNNYED MŰFAJ

ASTERIX, térkép	12, 14
DIZZY 3.5	13

MINDENFÉLE

Postafiók 334	15
Hirdetések, felhívások	16

ASSEMBLY

12. rész

Tartozom még Tisztelt Olvasóimnak a FILEDUMP program működésének magyarázatával, de merem remélni, hogy közülük már sokan megfejtették. Szó lesz a billentyűzetről is.

Még mielőtt belekezdünk: ismét érkezett hozzám néhány levél, amelyben azt panaszolják a levélírók, hogy hiába gépelték be pontosan, sorszámokkal együtt (!) a programlistát az editorba, azt nem tudták lefordítani. Nos, ismét elmondom, hogy az assembly listák elején álló sorszámokat *nem kell, nem szabad, tilos* az editorba begépelni. A sorszámok azt a célt szolgálják, hogy egyszerűbb legyen a magyarázat.

Tehát a FILEDUMP

A filedump program elején először címkeknek adunk értéket: *fchn* (9) lesz az olvasásra megnyitott fájlcsatorna száma. Az *asmscr* (10) az ASMON legnagyobb képernyőcsatornájának száma, ide dolgozik a program. A *chr* (11) az egy sorban megjelenítendő hexadecimális értékek és a hozzájuk tartozó (látható) karakterek számát adja. Az *x* (12) értékkel a karakteres mezőre fogunk majd pozícionálni egy ravaszhúzással.

A programot azzal indítjuk, hogy (biztos ami biztos alapon) lezárjuk az *fchn* csatornát (15-16). Ezt azért célszerű megtenni, mert, gondolom, olvasóim közül sokan kíváncsiságból ide-oda beleírtak (nagyon helyes, ebből lesz a cserebogár!) a programba, s lesték, hogy mi történik. Ilyenkor - hiba esetén - a program megállt, és szerencsés esetben visszakaptuk az ASMON parancssortát, a csatorna pedig nyitva maradt. Ha ezek után a T. kíváncsikodó újra elindította a programot, előfordulhatott, hogy az EXOS (jogosan) visszautasította a csatorna újbóli megnyitását, elejéről olvasását. Egy már nyitott csatorna megnyitása nagyobb (és veszélyesebb) hiba, mint egy lezárt csatorna lezárása.

A következő négy sorban (18-21) megnyitjuk a fájlcsatornát olvasásra. Bizonyára többen észrevették, hogy az EXOS 1 utáni megjegyzésbe "írás"-t írtam, ami természetesen rossz. Elnézést kérek! Ha a nyitás során hiba volt (magyarul: az EXOS nem tudta megnyitni a csatornát, az akkumulátor tartalma nem nulla), akkor a RET NZ hajtódik végre, a program futása pedig befejeződik. Itt most ismét elmondom: ha nagyobb és igényes programot akarunk készíteni, akkor kell egy olyan modul írni, amely szépen kiértékeli a hibákat, és szöveges magyarázatot küld a felhasználónak. (Egy lusta barátom régebben egyik programjába sem épített hibakezelő és informáló modult, de még a hibakódot sem írta ki a képernyőre. Egyszer szavá tettem neki ezt. Azóta minden programja rendelkezik ezzel a modullal, amely hiba esetén lefut, és a következő sokatmondó sort írja ki hibás futáskor: "One or more error(s)." Erre nem tudtam mit lépni...)

A *cnt* címkével ellátott szón (149) fogjuk tárolni a fájl-offszet, induláskor ennek nulla értéket adunk (23-24). Ezt az offszetértéket írjuk majd ki a sorok elejére.

A fájl blokkonként olvassuk be, ezt végzi a következő négy sor (26-29). Itt van az *rdloop* ciklus eleje is, mert többször kell majd blokkot olvasnunk. A beolvasott blokkot a *buff* területre (136) tároljuk le. (Semmi jelentősége nincs annak, hogy én 256 bajtnyi helyet foglalok le, a fájlból csak *chr* darab bajtót töltök le). Később arra is rájön mindenki, hogy miért rakom el a bajtokat, miért nem dolgozom fel őket azonnal. (Ez is egy jogos megoldás lenne, de én nem ezt választottam.) Az EXOS 6 után illene egy kis hibadetektálást végeznünk.

Tudjuk azt, hogy blokkolvasás után BC-ben a még olvasásra váró bajtok száma van. A mi esetünkben ez négy okból érdekes:

1) A fájl végét nem az EOF figyelésével, hanem BC tartalmának vizsgálatával vesszük észre.

2) Ha BC-ben a hívás után 0 van, akkor az azt jelenti, hogy be tudta olvasni a *chr* számú bajtót, még nem ért a fájl végére. Ez síma eset.

3) Ha BC-ben a hívás után 0-tól eltérő, de *chr*-rel nem egyező érték van, akkor biztosan a fájl végére értünk, de van még néhány beolvasott bajt. (Ez akkor áll elő, ha a fájl mérete nem

egész számú többszöröse a *chr*-nek.)

4) Ha BC-ben *chr*-el megegyező szám áll, akkor az azt jelenti, hogy egy árva bajtót sem tudott beolvasni az EXOS 6. (A fájl mérete egész számú többszöröse *chr*-nek.)

A következő négy sor ezeket az eseteket dolgozza fel. Az akkumulátorba töltjük a *chr* értéket (31). Az EXOS 6 után BC-ben, de mint tudjuk valójában csak az alacsonyabb értékű C regiszterben lesz a még beolvasandó bajtok száma. Az akkumulátorból kivonjuk C értékét (32). Ha C-ben *chr*-rel egyenlő szám volt, akkor egyetlen bajtót sem tudott olvasni az EXOS, biztosan a fájl végére értünk (1. az 1. és a 4. esetet). A kivonás eredménye ekkor 0 lesz, ki kell lépni a programból (33).

Ha a C tartalma *chr*-től különböző volt, akkor a kivonás után A-ban a ténylegesen beolvasott bajtok száma lesz (2. és 3. eset), és tudjuk, hogy még nem értünk a fájl végére. (Persze, ha A-ban az érték *chr*-rel nem egyező, akkor valójában már a fájl végénél vagyunk, de még nem dolgozzuk fel.)

A ciklusokat DJNZ-vel alakítjuk majd ki, B-be töltjük ezéret az akkumulátor tartalmát (34). Ezután az értéket kétszeri felhasználáshoz (hexa-rész, karakteresrész) ráteszünk a veremre (36, 37).

Lássunk valamit!

Olyan helyzetben vagyunk, amelyben tudjuk a beolvasott karakterek számát, a beolvasott karakterek a pufferben vannak, és az offszetszámlálónk is rendben van. Nincs tehát más dolgunk, ezeket kell megjelenítenünk, felhasználnunk.

A *table* memóriarészen helyeztem el a hexadecimális karaktereket (146), innen egyszerű indexeléssel fogom kiszedni a karaktert.

Elsőként az offszetet kell kiírom. Mindannyian tudjuk, hogy a *cnt*-nél letárolt offszetérték alacsony bajtja a *cnt*-n, a magas a *cnt*+1 helyen van.

Nekünk először a magas értéket kell kiírnunk. Ezt két lépésben tesszük meg, mert egy 8 bites számot (high ill. low rész) kétjegyű hexadecimális számmal tudunk leírni, tehát a két lépés a két megjelenítendő ASCII karaktert jelenti. Ugyanígy kell majd az alacsony értéket kiírnunk. (Összesen 4, hexadecimális karaktert jelenítünk meg, ami egy 16 bites számot (offszetérték) takar.)

HL az offszetérték magas bajtjára mutat kezdetben (40), IX mutat a konverziós tábla elejére (41). A *cycle* ciklus rakja ki a két 8 bites hexadecimális számot (high, low), ezért ennek kétszer kell majd lefutnia (42). A *cycle* elején a ciklusszámláló értékét mentjük el (43).

Az éppen érdekes hexa karaktert úgy fogjuk elérni, hogy egyszerűen a számmal magával fogunk a *table*-be indexelni. Ennek értéke nyilván a 0-15 tartományba eshet. Ennek kiszámításánál hasznosítható nagyon jól a RLD utasítás.

Az akkumulátort megtisztítjuk, tartalmát töröljük (44), mert az RLD ezzel dolgozik.

Most írunk oda, ahol a high ill. a low rész két-két karakterét tesszük ki, ennek is kétszer (45) kell lefutnia (gyakorlatilag egy 8 bites számot teszünk ki).

Belépünk a *cycle2*-be, ahol elmentjük a ciklusszámlálót (46). Jöhet az RLD! (Remélem mindenki emlékszik rá!) HL az offszet magasabb értékére mutat, az akku tartalma nulla. Az RLD után (47) A-ba kerül a high rész felső 4 bitje. A kapott (index) értéket beleírjuk magába a programkódba (48)! HL értékét nem akarjuk majd elrontani (nem lilik), ezért elmentjük az akkumulátortartalmát (49). Az LD B₁(IX+0) utasítás 3 bajtót, a harmadik bajton van az eltolás értéke (ez a gépi kód, egyes trükkökhöz ismerni kell a gépi kódot is), ide (50) írjuk be azt az IX-hez adandó értéket. (Az a 0, ami a listában szerepel, nem érdekes, az eltolás a program futása közben változik.) Ennek végrehajtása után B-ben lesz az az ASCII kód, ami az adott 4 bithez tartozik. Nincs más dolgunk, egyszerűen ki kell írunk (51-52).

Visszaállítjuk az akku (mert ilik, sói, már muszáj) és a ciklusszámláló tartalmát (53-54). Amikor a *cycle2* ciklus másodszorra fut le (55), az A tartalma az adott rész következő (alsó) 4 bitjét tartalmazza, s ezt fogja kiírni.

A záró RLD-t (56) nem lenne kötelező végrehajtani, de azért érdemes, mert HL és az A regiszter tartalma a háromszori RLD miatt az eredetire áll vissza. (De ennek itt nincs jelentősége.)

Csökkentenünk kell a HL-t (57), így az az érvényes offszetérték alacsony bajtjára mutat. Elő kell vennünk a ciklusszámláló értékét is (58), és jöhet a következő ciklus (59).

Elértük tehát, hogy megjelent a 16 bites offszetérték hexadecimális formában. Az offszetértéket növelnünk kell a következő kiíráshoz. Értékéhez nyilvánvalóan a *chr* értéket kell adni (62-65).

A következő részben egy kettőspontot teszünk ki határolójelnek, közönséges karakterírással (68-70). (Nem szóltam róla, mert gondolom mindenki számára világos, hogy a kurzort az EXOS mozgatja, ezzel semmi gondunk - egyelőre.)

Következik a hexa számsor kirakása. Először levesszük a veremről a beolvasott karakterek számát (73), az RLD miatt az akkumulátor tartalmát nullázzuk (74), majd HL-lel megcímezzük azt a puffert, amelyben a karakterek vannak (75).

A *loop0* ciklus (76) annyiszor fog lefutni, ahány karaktert beolvasunk (egy blokkban) a puffertérületre. A szám megjelenítése hexa formában ugyanúgy történik, mint ahogy az előbb láttuk (77-88). Az egyes számok közé szóközt teszünk (89-91). Növeljük HL-t, így az már a következő bajtra mutat a puffertben (92), a rutin ezt fogja a következő menetben kiírni hexában. Ezután felvesszük a ciklusszámlálót a stack-ről (93), és ha van még megjelenítendő bajtjérték, akkor visszaugrunk a ciklus elejére (94).

Most jön egy érdekes húzás. A kurzort rá kell igazítanunk arra az oszlopra, amelyben a tényleges karakterek fognak szerepelni. Hogy miért kell külön rápozícionálni, miért nem elég egyszerűen néhány szóközt kitenni a hexa részt követően? A válasz egyszerű. Képzeljük el, hogy mi történne, ha így eszelekednénk akkor, amikor már csak mondjuk 5 bajtunk lenne (a fájl vége)! Megjelenítenénk az öt értéket hexában, aztán kitenénk azt a néhány határoló szóközt, végül megjelenítenénk a karaktereket. Az eredmény ronda lenne, mert a hexa részbe esnének a karakterek.

Nos, a kurzorpozícionáló escape-szekvencia ismeri a következő trükköt: ha valamelyik részébe (x ill. y koordináta) 0-t teszünk, akkor a rendszer azon nem változtat, a kurzor abban a sorban ill. oszlopban marad, ahol éppen áll. Nekünk arra van szükségünk, hogy a kurzor y pozíciója ne változzon, ezért ezt nem bántjuk, az x pozíciót pedig 30-ra állítjuk be. (A pozícionáló szekvenciánál a 32 jelenti a 0 bázisértéket, ehhez kell hozzáadni a kívánt értéket. Még azt szeretném megjegyezni, hogy ez nem nagy trükk, sok más rendszer, bővítés és programnyelv tudja így pozícionálni a kurzort, nem csak az EXOS.)

Ezt a pozícionálást végzi el a blokkíró rész, amely a pozícionáló szekvenciát küldi el az ASMON képernyőjére (97-100).

Jöhet a karakterek megjelenítése. A feladat egyszerűbb, mint a hexaszámok kitétele, egy dologra azonban figyelniünk kell. Mint tudjuk, az ASCII kódrendszerben a 32-es kód (szóköz) alatt nem nyomtatható karakterek vannak. Éppen ezért, a karakterről el kell döntenünk, hogy megjeleníthető-e? Ha nem az, akkor is ki kell tennünk valamit, a mi programunk "-ot rak ki.

Először le kell vennünk a stack-ről a beolvasott, megjelenítendő karakterek számát (103), majd HL-lel megcímezzük a puffert (104).

A *loop3* ciklus jeleníti meg a karaktersort, a ciklus elején - ahogy már megszoktuk - elmentjük a ciklusszámlálót (105). Az akkumulátorba töltjük a HL-lel megcímezett bajt értékét (106), azután összehasonlítjuk az első megjeleníthető karakterrel, a szóközzel (107). Ha az érték nagyobb vagy egyenlő, akkor egyszerűen átugrunk (108) a karakteríró részre, egyébként rácsorgunk a következő utasításra, és A-ba töltjük a "-ot (109). A

következő utasítások hatására megjelenik a karakter (110-112).

Növeljük HL-t azért, hogy az a következő megjelenítendő karakterre mutasson a puffertben (113). Felvesszük a ciklusszámlálót (114), és ha van még karakter, akkor visszaugrunk a ciklus elejére (115).

Elkészült egy teljes sor! Jöhet a következő, amit egy sorral lejjebb kell megjeleníteni, ezért csinálunk egy soremelést (118-121). (Hatására a kurzor automatikusan a sor elejére ugrik. Egyébként itt van jó példának két nem nyomtatható karakter, a CR (Carriage Return - kocsis vissza, értéke 13) és az LF (Line Feed - soremelés, értéke 10). Ha ezeket a karakteres részben gondolkodás nélkül kitennénk, igencsak nagy zagyvaságot okoznának.)

Miután készen vagyunk, visszaugrunk a blokkolvasó részre (124). Ha aztán ott nincs több olvasható karakter, akkor a vezérlés a *total* címkénél kezdődő utasításokkal folytatódik, amelyek lezárják a fájlszámot (126-128).

Hát ennyi az egyik lehetséges, korántsem teljes megoldás.

Haladjunk az "anyaggal"!

Ismerjük a képernyő- és a fájlkezelést. De van még egy nagy problémánk: nem tudunk kommunikálni a géppel. Erre szolgál a billentyűzet. (Nocsak, nocsak! *A szerk.*)

A billentyűzet tipikusan beviteli eszköz (csak olvasásra lehet megnyitni), az EXOS-ban KEYBOARD: a periféria elnevezése. Mivel csak egy billentyűzet jár a géphez, így csak egy csatorna lehet a billentyűzethez rendelve. Az előző mondatból már kiderült, hogy a billentyűzetet is a "csatornaszemlélettel" alapján kell használni: a megnyitás, olvasás, lezárás teljesen ugyanúgy megy, mint pl. a fájlknál.

A billentyűzetről karaktereket és blokkokat is tudunk olvasni. A blokkok itt a funkcióbillentyűkhöz rendelt szövegeket jelentik, ezt is a KEYBOARD: kezeli. Nagyon hasznos az ún. *csatornaolvasási-állapot* funkció. Ennek segítségével programunk futása közben megnevezhetjük, hogy van-e olvasható karakter a billentyűzet saját puffereiben. Hogy hol lehet ezt használni? Pl. írunk egy programot, amely - amíg nem nyom le egy billentyűt a felhasználó - demózik.

Másik igen hasznos szolgáltatása ennek az eszköznek a STOP billentyű figyelése. Egyik rendszerváltozó (STOP_IRQ) igénybevitelével gyönyörű szoftvermegszakításokat lehet készíteni, amellyel ügyes hibakezelést lehet végezni pl. fájlműveletek közben. (Erre jó példa a SPRED program, amely ilyen szoftvermegszakítások segítségével védi magát mindenféle hibáktól.) A billentyűzetnél be lehet állítani a késleltetést és az ismétlés sebességét.

Másik fontos tudnivaló az eszközről, hogy a belső és a két külső botkormányt is kezeli. Ezeket a *speciális funkció* (EXOS 11) hívásával pöccgöztethetjük. A funkció hívása előtt A-ba kell tölteni a billentyűzet csatornaszámát, B-be 9-et (a JOY funkció), végül C-be az illető botkormány azonosítóját: 0-beépített, 1-külső, 2-külső. A hívás után C-ben lesz az adott botkormány állása. A bebillent bitek jelentése: bit0 - jobb, bit1 - bal, bit2 - le, bit3 - fel, bit4 - tűzgomb. Természetesen ezek kombinációját is kaphatjuk.

Mostanra ennyit. Megemlítem, hogy már a sorozat minden olvasója belekezdhet önálló programocskák kidolgozásába, hiszen a fontosabb ismeretek (utasítások, az ASMON, képernyő-, billentyűzet- és fájlkezelés, csatornaszemlélet stb.) kéznél vannak. Csak néhány ötlet: scroll rutinok, effektek szöveges és grafikus képernyőkre; fájlmásoló, fájlredaktor stb. Mindenkinek próbálkoznia, gyakorolnia, tapasztalnia kell, csak így tud haladni.

— HC —

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

12. rész

Az elmúlt néhány folytatásban sokat foglalkoztunk a Turbo-Pascallal, kissé elhanyagolva azoknak az olvasóinknak a lelkivilágát, akiknek be kell érniük a HiSoft Pascallal. Ez nem valamiféle szándékos diszkrimináció volt, egészen egyszerűen a Turbo-Pascal egy teljesebb, gazdagabb implementáció, több lehetőséget nyújt a felhasználónak, mint akaratlan vetélytársa, amelyet eredetileg egy egyszerűbb környezetre fejlesztettek ki; így többet lehet - és kell is - vele foglalkozni. A Turbo-Pascal lehetőségeinek nagy részét ismertettük; ami kimaradt, azt kisebb részben most pótoljuk, a maradékot az avatott felhasználó amúgy is csak egy részletesebb leírás birtokában tudja használni.

A fordító direktívák

Amivel még mindenképpen adósok vagyunk, az a *fordító direktívák* rövid összefoglalása. Miért is van ezekre szükség, és egyáltalán, mik is ezek? Nos, egy programnyelv - legalábbis napjainkban - úgy készül, hogy minél nagyobb mértékben független legyen egy adott géptípus, modell, konfiguráció sajátosságaitól. Azonban a lefordított programnak egy konkrét gépen (azaz típuson, modellen, konfiguráción) kell futnia. Vannak más sajátosságai is a program futásának, az ember-gép kapcsolatnak stb., amit nem lehet nyelvi szinten elintézni. Itt lépnek be a fordító direktívák. Ezek olyan utasítások, amelyek a fordítóprogramnak szólnak, és előírják számára a fordítás során előállítandó program valamilyen tulajdonságát. A direktívák tehát nem programutasítások, nem keletkeznek belőlük gépi kód. Nagy vonalakban: a *forrásprogram*, a *programszöveg* a fordítóprogram által feldolgozott *nyersanyag*, a direktívák pedig egyfajta utasítások, hogy hogyan is dolgozzon ezzel a nyersanyaggal...

A direktívák igen sajátos módon adhatók meg. A nyelv eredeti leírása csak egyetlen helyen engedélyezi *tetszőleges* információ megadását, mégpedig a *megjegyzésekben*. Ezért a Turbo-Pascal (és más Pascal-implementációk is) egy speciális megjegyzést vezet be a direktívák megadásához. Tudjuk, hogy a Pascalban a megjegyzés a kapcsos zárójelben álló tetszőleges szöveg, amely akármilyen hosszú lehet (alternatív megoldás a * és a *) jelkombináció alkalmazása, például olyan környezetben, ahol a terminál vagy a nyomtató nem ismeri a kapcsos zárójelket). A fordító direktíva olyan megjegyzés, amely dollárjellel kezdődik; ezt követi a direktíva törzse. Ez minimális korlátozást jelent csak a megjegyzések használatára.

A direktíva törzsének fő eleme a direktívát azonosító egybetűs kód, ezt követi a direktíva paramétere. A legtöbb direktíva valamilyen jellemző be- vagy kikapcsolt állapotával kapcsolatos, ezért a direktívák többségének paramétere a + vagy - jel, ezek jelentése értelem szerű. Más esetben egy fájlnev vagy egy érték lehet a paraméter. Ennek megfelelően a direktívák valahogy így néznek ki a program szövegében:

```
{ $A+
  (*$FILENAME.EXT*)
  { $WB}
```

Vigyázzunk, hogy a nyitó kapcsos (vagy összetett) zárójel és a dollárjel, illetve ez utóbbi és a direktíva betűkódja között ne álljon szóköz.

Minden direktívának van egy *alapértelmezés szerinti értéke*. Ha nem tüntetjük fel a direktívát, akkor ez az alapérték lesz érvényes. Egyes direktívák *lokálisak*, azaz csak a bekapcsolásuk és kikapcsolásuk közötti programszövegre érvényesek. Más direktívák *globálisak*, azaz az egész programszövegre kifejűk hatásukat; ezeket csak egyszer lehet megadni. Mindegyik direktívánál megadjuk az *alapértelmezés szerinti értéket*, és feltüntetjük a direktíva *lokális* vagy *globális* jellegét.

Az A direktíva: abszolút kód

Az A direktíva a rekurzív hívások használatát szabályozza. Ha a direktíva *aktív*, akkor a program *abszolút kódot* generál, azaz rekurzív hívás nem használható. A direktíva *passzív* állapotában használhatunk rekurzív hívást. Az alapértelmezés szerinti érték az *aktív* állapot. Ha tehát rekurzív programot akarunk írni, a program elején adjuk ki a

```
{ $A-}
```

direktívát. Készüljünk fel arra, hogy ilyenkor a program kód-

ja valamivel hosszabb lesz, a végrehajtás pedig lassabb. Ertérés lesz a verem használatában is, a rekurzív program sokkal jobban igénybe veszi a vermet.

Akit érdekel ennek mechanizmusa, elmondhatjuk, hogy mindaddig, amíg a programban nincs rekurzív hívás, az eljárások és függvények saját, belső változóit tulajdonképpen bárhová teheti a fordítóprogram, akár statikusan is kijelölheti azok helyét ugyanúgy, ahogy a (fő)program - globális - változóival teszi. Ezek címzése viszonylag egyszerű és gyors. Ha azonban rekurzív hívást akarunk végrehajtani, a lokális változókat csak a veremben foglalhatjuk le. Csak így érhető el, hogy minden egyes hívásnál a változókból egy új készlet jöjjön létre, és a visszatéréskor ezek a változók megszűnjének létezni. Ekkor azonban ezeket a változókat a processzor csak relatív címmel érheti el, ez egy kicsit tovább tart.

Ha már szóba került a tárkezelés rekurzió esetén, újra szólnunk kell arról, hogy a hívások egymásba ágyazásának van egy fizikai korlátja, mégpedig a verem számára rendelkezésre álló memória. Ha ugyanis túl sok hívást ágyazunk egymásba, a verem számára kijelölt memória túlszordul, és ez a program elszállását okozhatja. Ha ez a veszély fenyeget, használjuk a verem ellenőrzését előíró K fordító direktívát (lásd később).

A B direktíva: logikai eszköz kijelölése

A B direktíva a standard bevitelhez és kivitelhez hozzárendelt eszközt választja ki. Alapértelmezés szerinti, azaz *aktív* állapotában a CON: logikai eszközt, *passzív* állapotában a TRM: logikai eszközt választja ki. Talán még emlékszünk rá, mindkettőt ugyanazt a *fizikai* eszközt, a képernyő-billentyűzet kombinációt jelenti. A különbség az, hogy a CON: logikai eszköz *puffereelt* bemenetet ad, azaz a beírt információt szerkeszthetjük, törölhetjük mindaddig, amíg az ENTER billentyűvel le nem zártuk; a TRM: logikai eszköz viszont a leütött billentyű kódját *azonnal* továbbítja a programnak. Ha tehát mi magunk akarjuk a bevitelt irányítani a standard input-output fájlban, ki kell adnunk a

```
{ $B-}
```

direktívát (a direktíva globális).

A C direktíva: Ctrl-C és Ctrl-S használata

A C direktíva a Ctrl-C és a Ctrl-S billentyűkombináció használatát szabályozza. Mint tudjuk, a CP/M operációs rendszerben - és így az IS-DOS-ban is - a Ctrl-C billentyűkombinációval általában leállítható - megszakítható - a program, a Ctrl-S kombinációval pedig a túl gyors vagy túl hosszú képernyőkírás állítható meg és indítható le újra. A Turbo-Pascalra ez úgy érvényes, hogy a program megszakítása READ illetve READLN utasítás végrehajtása közben történhet meg. Ha a C direktíva *aktív* (és ez az alapértelmezés szerinti és szerinti állapot), a leírt két funkció működik; a direktíva *passzív* állapotában nem. Ha azt akarjuk, hogy a programunkat ne lehessen ilyen módon leállítani, adjuk ki a - globális -

```
{ $C+}
```

direktívát. Ezt azonban csak a hibátlan, belőtt programmal tegyük meg, különben saját magunkkal is kitalhatunk.

Az I direktíva: hibakezelés, beszúrt fájl

Az I direktíva kétféle célra is használható. Egyik esetben a beviteli-kiviteli hibakezelés vezérlésére használhatjuk. Ilyenkor a direktíva kapcsoló jellegű. Alapértelmezés szerint, azaz bekapcsolt állapotában a program maga ellenőrzi a bevitel-kivitel helyességét. Ilyenkor tehát hiba esetén a program futása félbeszakad, és kapunk egy hibadzenetet a hiba - számmal jelölt - kódjával.

Saját célra így is megfelel a program, ha azonban másokat is szerencsétlenné akarunk a művünk feletti gyönyör érzésében, az összes lehetséges, előre látható hibát a programban le kell kezelnünk. Ezt teszi lehetővé az I direktíva kikapcsolt állapota, amikor is hiba esetén a program futása folytatódik, mintha mi sem történt volna. Valami azért történik: az IORESULT függvény meghívásával mindig ellenőrizhetjük, hogy nem volt-e hiba az utoljára használt beviteli vagy kiviteli utasítás végrehajtása során. Idézzük fel: az IORESULT függvény értéke 0, ha a művelet hibátlan volt, és ettől eltérő értéket ad vissza, ha valamilyen hiba volt.

Mindezekből az következik, hogy egy jól megírt programban egy fájl-másolás kezdete valahogy így fog kinézni:

```
VAR
  FF, { forrás-fájl }
  CF { cél-fájl }
  : FILE;
  FFN, { fájlnevek }
  CFN
  : STRING;
...
WRITE( 'Kérem a forrás-fájl nevét: ' );
READLN( FFN );
ASSIGN( FF, FFN );
{ $!- } RESET( FF ) { $!+ } { Itt vesszük kézbe a
      sorsunkat }
IF IORESULT = 0 THEN BEGIN
  WRITE( 'Sajnos, nem tudom megnyitni a fájlt: ' );
  READLN;
  WRITELN( 'Vége...' );
  HALT;
END { IF };
...
```

A fájl-megnyitás idejére "átvesszük" a hibakezelést a Pascal rendszertől saját kezünkbe. Ugyanezt kell tenni cél-fájl esetében is, és legalább ez utóbbinak a lezárásakor. Ha következetesek vagyunk, akkor a beolvasást és a kiírást is ugyanígy lekezeljük, hiszen bármikor előfordulhat egy lemezhiba, vagy csak a szabad hely fogy el a lemezen.

Arra ügyeljünk, hogy az IORESULT függvény értéke a hívás után mindig nullára áll, azaz nem lehet a hibakódot így felidézni:

```
IF IORESULT = 0 THEN BEGIN
  WRITELN( '*** Hiba! A hibakód: ', IORESULT : 4 );
END { IF };
ugyanis ekkor mindig nullát kapunk. A helyes megoldás:
```

```
VAR
  IORES : INTEGER;
  { Ideiglenes változó a hibakódnak }
...
{ $!- } { Itt van a kritikus művelet } { $!+ }
IORES := IORESULT;
IF IORES = 0 THEN BEGIN
  WRITELN( '*** Hiba! A hibakód: ', IORES : 4 );
END { IF };
...
```

Az I direktíva másik alkalmazása az ún. include fájl, azaz beszúrt fájl megadása. Ha olyan hosszú a forrásszöveg, hogy a Turbo-Pascal már nem tudja kezelni, szétszabdálhatjuk több önálló darabra, ekkor mindig csak egy fájl van a memóriában. Egy másik lehetőség, hogy bizonyos feladatokhoz kész eljárás- és függvénykönyvtárral rendelkezünk, és ezt érintetlenül akarjuk a programunkhoz fűzni. A programunkhoz változatlanul hozzáfordítandó forrásszöveg az include-fájl.

A fordítóprogram ott fogja az include-fájlt beszerkeszteni a programunkba, ahol az I direktíva áll, Legyen a könyvtár neve LIBRARY.LIB, a program már elkészült és tesztelt másik részének pedig PROG1.PAS; ekkor így fűzhetjük össze a három részt a fordítás idejére:

```
PROGRAM PROG;
{ Itt állhatnak a saját deklarációk }
{ $! LIBRARY.LIB }
{ $! PROG1 }
{ Itt következnek a most szerkesztett programszöveg }
...
```

A programrészek ilyen módon végzett beszúrását segíti a Turbo-Pascalnak az a könnyítése az eredeti Pascal specifikációhoz képest, hogy a konstans-, típus-, változó-, eljárás- és függvénydeklarációk tetszőleges sorrendben, keverve is állhatnak a programban. Ez azt jelenti, hogy a beszúrt fájlok minden korlátozás nélkül deklarárlhatják saját adat- és programszerkezetüket.

A K direktíva: a verem ellenőrzése

A K direktíva, mint korábban utaltunk rá, a verem ellenőrzését írja elő. Az alapértelmezés szerinti, *aktív* állapotban a program minden hívás előtt ellenőrzi, hogy van-e elég hely a veremben a hívott eljárás vagy függvény lokális változói számára. Ha kevés a hely, a hibajelzést kapunk és a program leáll. A direktíva *passzív* állapotában nincs ellenőrzés. Ha tehát már beléptünk a programot, és vagy nem használunk rekurzív hívást,

vagy pedig biztosan tudjuk, hogy nem fogjuk túlterhelni a vermet, kikapcsolhatjuk az ellenőrzést a

{ \$K- }

direktíva megadásával; a program futása így valamivel gyorsabb lesz.

Az R direktíva:

az érvényességi tartományok ellenőrzése

Az R direktíva az indexek érvényességi tartományának, valamint a skalár- és résztartomány típus értelmezési tartományának ellenőrzését engedélyezi. Alapértelmezésben a direktíva *passzív*, azaz ilyenkor programhiba esetén "túlindekelhetünk" a tömb határain. Ha csak *olvassuk* a tömböt, ez nem okoz túl nagy problémát (leszámítva, persze, azt a kellemetlen élményt, hogy a program hibás eredményt ad). Ha azonban *írunk* is a tömbbe, elronthatjuk létfontosságú változók értékét, sőt akár a program kódrészébe vagy valamilyen rendszerterületre piszkíthatunk, ekkor a program nagy valószínűséggel elszáll. A résztartomány vagy felsorolási típus esetén ez a súlyosabb hiba nem fenyeget, de részben elvesztjük az ilyen változók használatát indokló előnyöket. A program belövésének idejére tehát érdemes a direktívát bekapcsolni:

{ \$R+ }

Ha már fut a program, és nem várható, hogy a felhasználó a saját adataival esetleg újra kiakaszthatja (a bevitt adatok érvényességét, ha az kritikus, amúgy is mindig ellenőrizni kell), akkor kikapcsolhatjuk a direktívát.

Az U direktíva: életveszély!

Az U direktíva rendeltetése az volna, hogy engedélyezze a program megszakítását *bárhol* (azaz nem csak READ és READLN utasítás végrehajtásakor; lásd a C direktívát). Hasznos segítséget jelenthetne programbelövéskor egy véletlenül beépített végtelen- vagy csak nagyon hosszú - hurok esetén. Sajnos, a direktíva bekapcsolt állapotában az általunk használt Turbo-Pascal (3.01-wes verzió) Enterprise-on elszáll.

A V direktíva: a stringparaméterek

típusellenőrzése

A V direktíva a string változó-paraméter típusellenőrzését végzi. A direktíva alapértelmezés szerinti, *aktív* állapotában az eljárásnak vagy függvénynek paraméterként átadott string és a formális paraméter hosszának meg kell egyeznie. A direktíva *passzív* állapotában nincs ellenőrzés, ekkor eltérő hosszúságú paraméter is átadható. Mivel általában ez a kedvezőbb állapot, tiltsuk meg a típusellenőrzést:

{ \$V- }

A W direktíva: a WITH utasítások mélysége

A W direktíva a WITH utasítások egymásba skatulyázásának maximális mélységét adja meg. Alapértelmezésben ez az érték 2; a direktívával a mélységet 1 és 9 között állíthatjuk be.

A WITH utasítás ismertetését annak idején elmulasztottuk; így azt most röviden pótoljuk. A rekord típusú változók használatakor, mint tudjuk, az egyes elemekre hivatkozáskor a rekord nevét is meg kell adni. Ez sokszor kényelmetlen, különösen, ha többszörösen egymásba skatulyázott rekordokról van szó, és feleslegesnek tűnik, ha például ugyanannak a rekordnak sok-sok elemét kell felsorolni. Hogy ez a dolog elkerülhető legyen, az érintett utasításokat egy ún. WITH blokkba zárhatjuk, és ekkor a blokkon belül a rekord nevét nem kell megadni, csak az egyes elemekét. Az alábbi deklaráció esetén:

```
VAR
  REKORD : RECORD OF
    A, B, C : INTEGER;
  END { RECORD };
(így néz ki egy értékadás a WITH utasítás nélkül)
```

REKORD.A := REKORD.B + REKORD.C

és így, ha használjuk a WITH utasítást:

```
WITH REKORD DO BEGIN
  A := B + C;
END { WITH };
...
```

Az X direktíva: a tömbök optimalizálása

Az X direktíva alapértelmezés szerinti, *aktív* állapotában a tömbkezelés futási idő szerinti optimalizálását váltja ki, míg *passzív* állapotában a fordítóprogram a memórafoglalást igyekszik a minimumra szorítani.

IS-LISP

2.rész

Függvények (I. táblázat)

Kapcsolat az EXOS-al

A LISP alapján (így az IS-LISP is) magasszintű programozási nyelv. Ez a tulajdonság azt jelenti, hogy más típusú gépeken futó LISP értelmezők forrásprogramjait is képes (kis számú módosítás után) végrehajtani. Így azonban az ENTERPRISE egyedi tulajdonságai (grafika, hang stb.) háttérbe szorulnak, hiszen más operációs rendszer nem biztos, hogy ugyanúgy kezel mindent, mint az EXOS.

Az IS-LISP a EXOS-al való kapcsolat révén jobban kihasználhatóvá teszi a gépközelet (gépfüggő) dolgok felhasználását a programozási munkában.

Az EXOS változók elérését az alábbi eljárások (subr típusú függvények) végzik: (EXOS-READ < változó >), amely az adott EXOS-változó értékét adja vissza; (EXOS-WRITE < változó > < érték >), amely az adott EXOS-változó átírása után, az új (megváltoztatott) értéket adja vissza; (EXOS-TOGGLE < változó >), amely az adott (lényegében két állapotot meghatározó) változó tartalmát billenti át.

Légrádi Gábor

Függvény	szolgáltatott érték	leírás	típus
(BEAM < kifejezés >)	NIL	Amennyiben a kifejezés nem NIL, akkor a rajzolás az adott videocsatornán bekapcsolódik, egyébként kikapcsolt lesz	subr
(BORDER < szám >)	NIL	A keret színének átírása	subr
(CLOSE < szám >)	Szám	Ha a szám egy érvényes csatorna száma, amelyik megnyitott file, úgy a file lezárásra kerül	subr
(DEL < csatorna >)	Csatorna	Az adott EXOS csatorna lezárására	
(EOF)	T vagy NIL	Teszteli az aktuális input csatorna állapotát	subr
(FLATTEN < struktúra >)	Lista	Eltávolítja a struktúrából a leágazásokat, így az eredménye egy lista	subr
(FLUSH < csatorna >)	NIL	Kiüríti az adott számú csatornát	subr
(GRS < csatorna >)	Szám	A grafikus csatorna beállítása az előző érték visszaküldésével	subr
(INK < szín >)	NIL	Előtérszín állítása a kiválasztott palettáról	subr
(JOY < szám >)	Szám	A botkormány leolvasása	subr
(MEMBER < x > < y >)	NIL vagy lista	A visszaadott érték NIL, ha X nem eleme Y-nak, egyébként a kapott lista az X elemmel kezdődik	subr
(MESSOFF MESSON < szám >)	Szám	A rendszer által adott üzenetek vezérlése. 1- szemetgyűjtés eredménye ennyi byte-ot szabadított fel (OFF); 2- a szemetgyűjtések száma (OFF); 4- hibaszám (ON); 8- hibakövetés (ON); 64- a QUOTE függvény vezérlése (ON).	subr
(OPEN < szám > < file név >)	Szám	File megnyitása	subr
(PAINT)	NIL	Kifestés	subr
(PALETTE < c0 > ... < c7 >)	NIL	A palettaszínek beállítása	subr
(PAPER < szín >)	NIL	A papír színének beállítása	subr
(PEEK < cím >)	Érték	A cím tartalmát adja	subr
(PLIST < azonosító >)	Lista	Az azonosító tulajdonságait tartalmazó listát adja vissza	subr
(PLOT < x > < y >)	NIL	Abszolút grafikus koordináta megadás	subr
(PLOTX < x > < y >)	NIL	Relatív grafikus koordináta megadás	subr
(PLOTMODE < szám >)	NIL	Rajzolás vezérlése. 0- egyszerű rajolás; 1- OR; 2- AND; 3- XOR	subr
(PLOTSTYLE < szám >)	NIL	A rajzolt vonal milyensége állítható be	subr
(POKE < cím > < érték >)	Érték	A címre az értéket írja	subr
(RANDOM < szám >)	Szám	A véletlenszámok előállítása	subr
(RANDOMIZE < sorszám >)	Sorszám	A véletlenszám-generálás vezérlése az adott sorszámtól kezdődik	subr
(RDS < csatorna >)	Csatorna	Módosítja az adott sorszámú csatornát és visszaadja a megelőző csatornaszámot	subr
(RECLAIM)	Szám	Tárolóhely visszavételt kezdeményez, a kapott számértéket öttel megszorozva a szabad memória méretét kapjuk	subr
(REDIRECT < régi > < új >)	NIL	A kimeneti műveletek átirányítása a csatornák között	subr
(SETATTRIBUTES < szám >)	NIL	Grafikus attribútum beállítása	subr
(SETCOLOUR < szám > < palettaszín >)	NIL	A palettaszín beállítása	subr
(SET-TIME < időstring >)	NIL	Az idő beállítása	subr
(SETVIDEO < felbontás > < színek > < video x > < video y >)	NIL	Videomód definiálása	subr
(SNDS < csatorna >)	Csatorna	Adott csatorna átírása	subr
(SPRINT < kifejezés >)	NIL	Formázott kiírás	subr
(WRS < csatorna >)	Csatorna	A kimeneti csatorna átirányítása	subr

EPSON kompatibilis nyomtatás az EP karakterkészletével

Köztudott hogy az Enterprise operációs rendszere 128-as karakterkészletet használ. A gyakorlatban ezt 32-159 közötti kódtartományban lehet használni. A 128-159 közötti karaktereket célszerű ékezetes betűkre és grafikus karakterekre cserélni. Az ilyen szöveg nyomtatását kétféleképpen is meg lehet oldani:

- Grafikusan

- A nyomtató karakterkészletének átdefinálásával

Mivel a második megoldás sokkal rugalmasabb és gyorsabb, ennél maradunk. Az EP 9*8-as, a nyomtató 8*11-es karaktermérettel dolgozik. Ebből következik, hogy vagy az alsó vagy a felső sort nem tudjuk értelmezni. A nyomtató 11 pontját úgy kell használnunk, hogy egymás után két pont nem lehet beállított állapotú. A nyomtató szoftvere gondoskodik arról, hogy az ilyen hibás pont ne kerüljön nyomtatásra. Megadható ezen kívül, hogy a printer a karakter nyomtatásánál az alsó vagy a felső 8 tűjét használja. A nyomtató a 11 pontból csak 9-et használ a karakterek definiálására és a maradék 2 a betűk közötti távolságot adja. Normál módban mind a 11 pont kiíródik. Van egy másik írásmód is ahol a karakterek méretével arányosan többkevesebb pont íródik ki. Azért, hogy a nyomtatott íráskép hasonlítson az EP képernyőjén megjelenő szövegre, célszerű erre a módra kapcsolni. Mivel az EP karakter 8 pont széles, az arányos karakterszélesség kezdő és végértékét 0-nak és 7-nek fogjuk definiálni. Ide tartozik még hogy a sortávolságot 9 pont magasra állítjuk.

Az alapértelmezésű nyomtató 32-134 közötti karaktereket képes nyomtatni (kivételek a 127-es kódú delete). Lehetőség van a karakterkészlet kiterjesztésére is, de a vezérlő kódokat így sem lehet felhasználni karakternyomtatásra. E problémát megoldhatjuk, ha a vezérlőkódokat nyomtatható kódszámra cseréljük.

A nyomtató karakterének definiálása:

ESC,&*,CHRSTART,CHREND,a,EP 0-7,0,0,0*

A két CHR-rel megadható a készlet alsó és felső értéke. Ennek ellenére egyenként küldjük ki a karaktereket, mivel a sorrendet a cserék miatt elrontottuk.

Az "a" egy flag (jelző-) byte: az alsó 4 bite=7 (arányos írás END). A következő 3 bite=0 (arányos írás START). A felső bit határozza meg, hogy a printer felső vagy alsó 8 tűje nyomtasson. Ha a karakter felső bájta nulla, akkor ezt a bitet töröljük és a definíciót a karakter második bájttól számoljuk. Ezzel lehetőség nyílik a "lenyúló" karakterek alsó részének kinyomtatására.

Az EP karaktereit felépítő 0-7 bájtokat a karakterkészletből olvassuk ki. (Az EP karakterképei a 255-ös szegmens B480h-B8Fh tartományában vannak. Az első 128 bájtt a karakterek felső sorát, a következő 128 a második sorát és így tovább tárolja. A 9*128 bájtt kiadja a 128 karakter 9 sorát.) A 0. bájtt a karakter bájttájak 7. bitjeiből, az 1. a karakterbájttok 6. bitjeiből stb. tevődik össze.

A három nulla bájttal 11 bájtosra egészítjük ki a nyomtató-karaktert.

A program használata

A KP-t elkészíthetjük ASMON-al az assembly kód lefordításával (6-os fejrész), de gyorsabb megoldás a BASIC-betöltő program begépelése és futtatása. Ezután már csak a KP-t kell LOAD-dal beolvasni. Mivel rendszerbővítésről van szó, a betöltés után a kikapcsolásig az a RAM-ban marad, miközben szabadon mozoghatunk a BASIC, WP stb. között.

A nyomtatás a KP parancs kiadásával az alapértelmezésű csatornáról fog történni, amely általában (de nem kötelezően) EDITOR-csatorna. A fájlok nyomtatásához a

KP fájlnév

parancsot kell kiadni. A WP-ből nem az F3-mal, hanem az [F8] KP [ENTER] begépeléssel nyomtathatunk.

A KP a nyomtatás előtt a következő írásképet állítja be: Sortáv=9 pixelsor, arányos, dupla nyomtatás. Ha más írásképet szeretnénk elérni, akkor előbb állítsuk be, majd a KPD parancsot használjuk, amely nem fogja a nyomtatót elállítani. Persze az aktuális karakterkészletet mindkét parancs a nyomtatóra küldi.

(Hsoft)

```

100 PROGRAM "kp.bet"
110 OPEN #1:"kp" ACCESS OUTPUT
120 DO
130 READ IF MISSING EXIT DO:AS$
140 DO UNTIL AS$=""
150 PRINT #1:HEX$(AS$(2));
160 LET AS$=LTRIM$(AS$(3))
170 LOOP
180 LOOP
190 CLOSE #1
200 DATA 0006 3602 0000 0000 0000 0000 0000 0000
210 DATA 79D6 02CA 31C1 30C0 B020 06D9 CD28 C009
220 DATA C9CD C1C0 3805 1841 0E03 C9CD 15C1 AF4F
230 DATA C9CD 01C1 4850 2020 2020 7665 7273 696F
240 DATA 6E20 312E 300D 0A00 C9CD 28C0 CD01 C120
250 DATA 2020 2020 2080 2031 3939 3220 4873 6F66
260 DATA 742E 000A 4550 534F 4E20 686F 6070 6174
270 DATA 6962 696C 6973 206E 796F 6074 6174 6173
280 DATA 2061 7A20 4550 2061 6874 7561 6C69 7320
290 DATA 6861 7261 6874 6572 6865 737A 6C65 7465
300 DATA 7665 6C2E 2050 6172 616E 6373 6F68 3A20
310 DATA 4850 2C48 5044 0D0A 00C9 856F D024 C9CD
320 DATA B4C0 7E23 666F C921 17C1 4806 0013 7E23
330 DATA B7C8 B928 07C6 04C0 B4C0 18F2 C5D5 1A13
340 DATA EDA1 201A EAD8 C0E5 CDBC C0E3 2323 CDBC
350 DATA C0E3 FDE1 D9D1 C118 1A91 D947 37C9 0923
360 DATA 2323 23D1 C118 C7E3 545D AF47 4FED B178
370 DATA 2F47 792F 4F3E FFF7 08E3 C9FD E902 4850
380 DATA A6C1 43C0 0348 5044 A5C1 43C0 0004 05C8
390 DATA 1AFE 20C0 1318 F7CD C1C0 3805 1841 0E02
400 DATA C9CD 27C1 DD21 0000 E5FD E128 3618 7812
410 DATA 2600 D57C F701 D128 10FE F928 D808 7CF7
420 DATA 0308 DE00 C924 20EA C901 0400 F710 5CD5
430 DATA 0104 0154 F710 CD7D C108 0163 0104 01F7
440 DATA 1018 DBDD 3937 CD15 C1DD F918 5D03 863A
450 DATA F2BF B73E E520 F2DB 86CB 5F20 F2F3 3AF3
460 DATA BFF6 10D3 B5E6 EFD3 B5FB C9B7 3E20 F5F5
470 DATA F521 32C2 0603 C02A C2F1 CD21 C2F5 CD87
480 DATA C1F1 CD87 C111 8000 4BF1 F5CB FF6F 2684
490 DATA 7EB7 3E87 2003 19CB B8CD 87C1 E506 017E
500 DATA 19A1 C6FF CB10 30F7 8BCD 87C1 E1CB 3930
510 DATA EB06 03AF CD87 C110 FAF1 3CFE A020 8021
520 DATA 35C2 0604 F130 0206 0BCD 2AC2 0120 0116
530 DATA CDF7 103E FFF7 0520 0978 CD21 C2CD 87C1
540 DATA 18F1 FEE4 C0AF C9FE 7FD8 FEAO D0C6 21C9
550 DATA 7E23 CD87 C110 F9C9 1826 001B 2501 001B
560 DATA 301B 7031 1847

```

;KP.ASM (C) 1992 Hsoft

;Object file name KP

;EXOS modul type

```

0
DRG 0CDDAH ;C=AKCIOKOD
LD A,C
SUB 2
JP Z,COMMAND
DEC A
RET NZ ;MEM KEZELT AKCIOKOD
OR B ;S=SZTRINGHOSSZ
JR NZ,HELP10
EXX ;URES HELP KERESE
CALL PRNHLP ;HELPSZOVEG KIIRAS
EXX ;RC,DE MEGORZES
RET
HELP10: CALL STREQUO ;SZTRING AZONOSITAS
JR C,HELP20
DEC DE ;ISMERTLEN
LD B,C
LD C,3
RET
HELP20: CALL JPIY ;VEGREHAJTAS
XOR A ;NINCS HIBA
LD C,A
RET
PRNHLP: CALL STRPRINT
DEFB "KP version 1.0",13,10,0
RET
PHHELP: CALL PRNHLP
CALL STRPRINT
DEFB " " ,128," 1992 Hsoft.",13,10
DEFB "EPSON kompatibilis nyomtatás az "
DEFB "EP aktuális karakterkészletével."
DEFB " Parancsok: KP,XPD",13,10,0
RET
ADDHLA: ADD A,L ;HL=HL+A
LD L,A
RET NC
INC H
RET
HLHLA: CALL ADDHLA ;HL=(HL+A)
LD A,(HL) ;HL=(HL)
INC HL
LD H,(HL)
LD L,A
RET
STREQUO: LD HL,CONSTR ;PARANCSZO AZONOSITO
LD C,B
LD B,0
INC DE

```



```

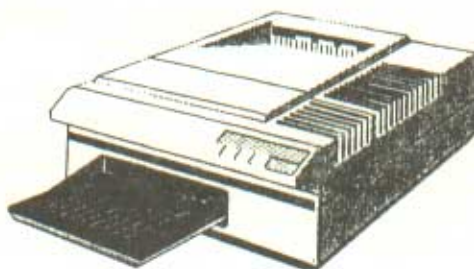
STREQU: LD A,(HL) ;BC(DE)=(HL)?
        INC HL ;NEM: CARRY=0
        OR A ;IGEN: CARRY=1
        RET Z ;HL=COMMANDCIM
        CP C ; ;
        JR Z,STREQU10 ; ;
        ADD A,4 ; ;
        CALL ADDHLA ; ;
STREQU10: JR STREQU ; ;
        PUSH BC ; ;
        PUSH DE ; ;
STREQU20: LD A,(DE) ; ;
        INC DE ; ;
        CPI ; ;
        JR NZ,STREQU30 ; ;
        JP PE,STREQU20 ; ;
        PUSH HL ; ;
        CALL HL,HL ; ;
        EX (SP),HL ; ;
        INC HL ; ;
        INC HL ; ;
        CALL HL,HL ; ;
        EX (SP),HL ; ;
        POP IY ; ;
        EXX ; ;
        POP DE ; ;
        POP BC ; ;
        DEC DE ; ;
        LD A,(DE) ; ;
        SUB C ; ;
        EXX ; ;
        LD B,A ; ;
        SCF ; ;
        RET ; ;
STREQU30: ADD HL,BC ; ;
        INC HL ; ;
        INC HL ; ;
        INC HL ; ;
        INC HL ; ;
        INC HL ; ;
        POP DE ; ;
        POP BC ; ;
STRPRINT: JR STREQU ; ;
        EX (SP),HL ;A HIVAST KOVETO
        LD D,H ;SSTRINGET A DEFAULT
        LD E,L ;CSATORNARA KIIRJA.
        XOR A ; ;
        LD B,A ; ;
        LD C,A ; ;
        CPIR ; ;
        LD A,B ; ;
        CPL ; ;
        LD B,A ; ;
        LD A,C ; ;
        CPL ; ;
        LD C,A ; ;
        LD A,255 ; ;
        EXOS B ; ;
        EX (SP),HL ; ;
        RET ; ;
JPIY: JP (IY) ; ;
COMSTR: DEFB 2,"KP" ;A PARAMCSSZAVAK LISTAJA
        DEFW KPRN,PHELP ;COMMANDCIM ES HELPCIM
        DEFB 3,"KPD" ; ;
        DEFW KPRNO,PHELP ; ;
SPACE: DEFB 0 ;NINCSEN TOBB SZO
SPACE10: INC B ;A SZOKOZKARAKTEREK
        DEC B ;ATUGRASA
        RET Z ; ;
        LD A,(DE) ; ;
        CP " " ; ;
        RET NZ ; ;
        INC DE ; ;
        JR SPACE10 ; ;
COMMAND: CALL STREQU0 ;PARAMCSSZO AZONOSITAS
        JR C,COMMAND10 ;UGRAS HA FELISMERTE
        DEC DE ;ISMERETLEN PARANCS
        LD B,C ;VISSZA MAS BOVITESEKNEK
        LD C,2 ; ;
        RET ; ;
COMMAND10: CALL SPACE ;SZOKOZ UGRAS
        LD IX,0 ; ;
        PUSH HL ;HL=PARAMCSKEZELO CIME
        POP IY ; ;
COMMAND20: JR Z,COMMAND100 ;UGRAS HA NINCSEN FAJLNEV
        DEC DE ; ;
        LD A,B ; ;
        LD (DE),A ;FAJLNEVHOSSZ BEIRAS
        PUSH DE ;KEZDO CSATORNASZAM
        LD A,H ;FAJLNEVCIM MENTES
        EXOS 1 ;CSATORNASZAM
        POP DE ;OPEN FAJLNEV
        JR Z,COMMAND40 ;FAJLNEVCIM VISSZA
        CP 249 ;URRAS NA MEG LETT NYITVA
        JR Z,COMMAND35 ;CHANNEL EXISTS?
        EX AF,AF' ;HIBAKOD MENTES
        LD A,H ;CSATORNASZAM
        EXOS 3 ;MEGSZUNTETES
COMMAND33: EX AF,AF' ;HIBAKOD VISSZATOLTES
        LD C,0 ;A PARANCS FEL LETT
        RET ;ISMERVE
COMMAND35: INC H ;CSATORNASZAM MOVELES
        JR NZ,COMMAND30 ;0-255
        RET ;MINDEH CSAT. FOGLALT
COMMAND40: LD BC,4 ;DEF. CSATSZAM OLVASAS
        EXOS 16 ; ;
        LD E,H ;A DEF.CSAT. ES A NYITOTT
        PUSH DE ;CSAT. SZAMOK ELMENTESE
        LD BC,256+4 ;A DEF CSAT. ATIRASA
        LD D,H ; ;
        EXOS 16 ; ;
        CALL COMMAND100 ;NYOMTATAS
        EX AF,AF' ;HIBAKOD MENTES
        POP DE ;A DEF. CSATORNASZAM
        LD H,E ;VISSZAIRASA
        LD BC,256+4 ; ;
        EXOS 16 ; ;
        JR COMMAND32 ;A NYITOTT CSAT. LEZARASA
COMMAND100: ADD IX,SP ;A VEREMCIM ELMENTESE
        SCF ;CARRY=1
        CALL JPIY ;VEGREHAJTAS
        LD SP,IX ;VEREMERTEK VISSZATOLTES
        JR COMMAND33 ; ;
        BYTOUT: OUT (0B6H),A ;A BAJT NYOMTATORA KULDESE

```

```

BYTOUT10: LD A,(0BFF2H) ;SZOFTVERMEGSAKITAS VOLT?
        OR A ; ;
        LD A,DESH ;STOP KEY PRESSED
        JR NZ,EXIT ; ;
        IN A,(0B6H) ; ;
        BIT 3,A ;KESZENLETI JEL OLVASASA
        JR NZ,BYTOUT10 ;HA NINCSEN KESZEN, VISSZA
        DI ; ;
        LD A,(0BFF3H) ;B5 PORT AKTUALIS ERTEK
        OR 10H ;SET 4,A
        OUT (0B5H),A ;PORTOLVASAS ENGEDELYEZES
        AND DEFH ;RES 4,A
        OUT (0B5H),A ; ;
        EI ; ;
        OR A ; ;
        LD A,32 ;CARRY=0
        PUSH AF ;KEZDO KARAKTERSZAM
        PRN10D: PUSH AF ;CARRY ELMENTES
        PUSH AF ; ;
        LD HL,PK1 ;KARAKTER DEFINIALASA
        LD B,3 ; ;
        CALL BLOKKOUT ;3 BYT KIKULDES
        POP AF ; ;
        CALL DEKOD ;KARAKTER KONVERTALAS
        PUSH AF ; ;
        CALL BYTOUT ;2-SZER KIKULDES
        POP AF ; ;
        CALL BYTOUT ; ;
        LD DE,12B ;KARRESZ. MOVEKMHENY
        LD C,E ;C=BITMASZK
        POP AF ; ;
        PUSH AF ; ;
        SET 7,A ; ;
        LD L,A ; ;
        LD H,0B4H ;HL=8480+(CHR and 127)
        LD A,(HL) ;HL=KAR. FELSO BYTJA
        OR A ;A FELSO BYTE MULLA?
        LD A,7+12B ;A FELSO SORTOL
        JR NZ,PRN110 ;FOGJUK KIKULDENI
        ADD HL,DE ;A MASODIK SORTOL
        RES 7,A ;FOGJUK KIKULDENI
        CALL BYTOUT ;TIPUSBYTE KIKULDES
        PRN200: PUSH HL ;KEZDOCIM MENTES
        LD B,1 ;B=BYTE PUFFER
        PRN300: LD A,(HL) ; ;
        ADD HL,DE ;HL=A KAR. KOV. BAJTJA
        AND C ; ;
        ADD A,255 ;BITMASZKOLAS
        RL B ;BIT A CARRY-BA
        JR NC,PRN300 ;CARRY A B-BE
        LD A,B ;B-SZOR
        CALL BYTOUT ;A BYTE KIKULDESE
        POP HL ; ;
        SRL C ;BITMASZK GORGETES
        JR NC,PRN200 ;B-SZOR
        LD B,3 ; ;
        PRN400: XOR A ;3-SZOR NULLA KIKULDES
        CALL BYTOUT ; ;
        DJNZ PRN400 ; ;
        POP AF ; ;
        INC A ; ;
        CP 160 ;KOVETKEZO KARAKTERSZAM
        JR NZ,PRN100 ;VEGET
        LD HL,PK2 ;VISSZA HA NEM
        LD B,PK3-PK2 ;NYOMTATO VEZERKODOK
        POP AF ;MINIMALISAN
        JR NC,PRN450 ;A CARRY VISSZATOLTESE
        LD B,PK4-PK2 ;NEM KELL IRASKEP BEALL.
        CALL BLOKKOUT ;IRASKEP BEALLITAS IS KELL
        LD BC,256+32 ;BYTOK KIKULDESE
        LD D,11000000B ;AZ EDITOR OLVASASJELZO
        EXOS 16 ;IRASA: A TELJES PUFFER
        LD A,255 ;AZONNALI OLVASASA
        PRN500: ;DEFAULT CSATORNA
        EXOS 5 ; ;
        JR NZ,PRN900 ;KARAKTER OLVASAS
        LD A,B ;UGRAS HA HIBA TORTENT
        CALL DEKOD ;A KAR. KONVERTALASA
        CALL BYTOUT ;A KAR. KIKULDESE
        JR PRN500 ;A KAR. OLVASNI
        CP 228 ;END OF FILE?
        PRN900: RET NZ ;NEM
        XOR A ;A FAJLVEGE NEM HIBA
        RET ; ;
        CP 127 ;0-126?
        RET C ;IGEN
        CP 160 ;160-255?
        RET NC ;IGEN
        ADD A,33 ;(127-159) PLUSZ 33
        BLOKKOUT: LD A,(HL) ;HL=A SZTING CIME
        INC HL ;B=BAJTOK SZAMA
        CALL BYTOUT ; ;
        DJNZ BLOKKOUT ; ;
        DEF 27,"E",0 ;FELHASZNALDI K.K. DEFIN
        PK1: DEF 27,"E",1,0 ;FELHASZNALDI K.K. BE
        PK2: DEF 27,"O", ;SORTAV=9 PIXELSOR
        PK3: DEF 27,"O",49 ;ARANYOS IRAS BE
        DEF 27,"G", ;KETTOS NYOMTATAS
        PK4: END ; ;

```



Kooprocesszort az ENTERPRISE-ba!

Mint mindannyian tudjuk, az ENTERPRISE mindent első-rőn kinűnő számítógép. Van azonban néhány, magáról megfeledezett géptársunk (Mandelbrot-ábrákat szeretnek kirajzolni), akik azt a kijelentést engedik meg maguknak, hogy az EP lassú! Ami, ha magunkba nézünk, igaz.

A gondok enyhítésére (megoldásról aligha beszélhetünk, mert nincs gyors gép, csak keveset tud a program) Ari Sándor műhelyében született egy kooprocesszor-kártya, amelynek a buszkiterjesztő egységhez átigazított változatát ismertetem az alábbiakban.

A kártya "szíve" az Advanced Micro Devices AM9511 aritmetikai processzora (Intel megfelelője a 8231A). Ez egy olyan speciális áramkör, amely csak bizonyos számolási műveleteket (pl. gyökvonás, szögfüggvények számítása) képes elvégezni, azokat azonban sokkal gyorsabban, mint az általános célú mikroprocesszorok.

A szóbanforgó aritmetikai processzor (APU) háromféle: 16 bites fixpontos, 32 bites fixpontos, és 32 bites lebegőpontos számmal dolgozhat. Az ábrázolható lebegőpontos szám tartomány a $+/-(-2.7 \cdot 10^{20} \dots 9.2 \cdot 10^{18})$ és persze a 0.

Az adatregiszterek veremszervezésdek (stack). Egyetlen bejáratuk van, amelyből az utoljára beírt adat olvasható ki először. A verem mérete 16 bájt, nyolc 16 bites, vagy négy 32 bites szám fér bele, amelyeken a FORTH nyelvhez hasonló módon végezhetünk műveleteket.

Az APU működési módja a következő:

- Beírjuk az adatokat az adatregiszterekbe,
- beírjuk az elvégezni kívánt művelet kódját (pl. $\sin(x)$ -re 02h) a parancsregiszterbe,
- megvárjuk, amíg az APU elkészül,
- kiolvassuk a művelet eredményét az adatregiszterekből.

(A királyi többes itt az általunk, vagy más által (bár az nem olyan jó) írt programot jelenti, amelyet természetesen a főprocesszor, azaz a Z80 hajt végre.)

Lássuk a kártyát! Az APU (U1) adatbuszát U3 illeszti a Z80 adatbuszához. U2 választja ki a megfelelő I/O címeket. Itt is kis ráhagyással dolgoztam (az APU 4 címen látszik egyszerre), de ezáltal az áramkör egyetlen IC-re redukálódott. 50h az adatport címe, 51h pedig a parancsporté. U4 két része állítja elő a kétféle órajelt az AM9511DC (2 MHz), vagy az AM9511-4DC

(4 MHz) számára, amelyek közül a JP1-gyel választhatjuk ki a megfelelőt.

U5C a reset-jelét állítja elő az APU számára. U5A és U5B köti össze az APU READY kimenetét a Z80 -WAIT bemene-tével. Ezzel a megoldással az APU "megállítja" a Z80-at a számítások időtartamára, így nem kell a programból figyelni státuszbitet nézegetésével, hogy elkészült-e már pl. a gyökvonás. Ha azonban még haladóbbak vagyunk, megtehetjük, hogy párhuzamos működésre vesszük rá a két processzort, és amíg az APU számol, a Z80 pl. képernyőt rajzol. Ehhez JP2-t bontanunk kell (valamint rengeteget és nagyon ügyesen programoz-nunk!). Ilyenkor az ún. állapotzó - amelyet a parancsport olvasásával kaphatunk meg - 7. bitje mutatja, ha az APU még dolgozik. (A fiam bejön a szobámba, megnézi a 7. bitemet, és máris látja, hogy dolgozom-e, avagy sem.)

A végére maradt a tápfeszültségeket előállító áramkör. Az AM9511 élemedett kora azon is látszik, hogy szerencsétlen a +5 volt mellett még +12 voltot is óhajt a működéséhez. Na, ezt állítja elő a bal alsó sarokban látható T1-T4, D1-D4, U7 komplexum. U6 pedig a már megszokott 5 voltos stabilizátor, némi hidegítő kondenzátorral ellátva.

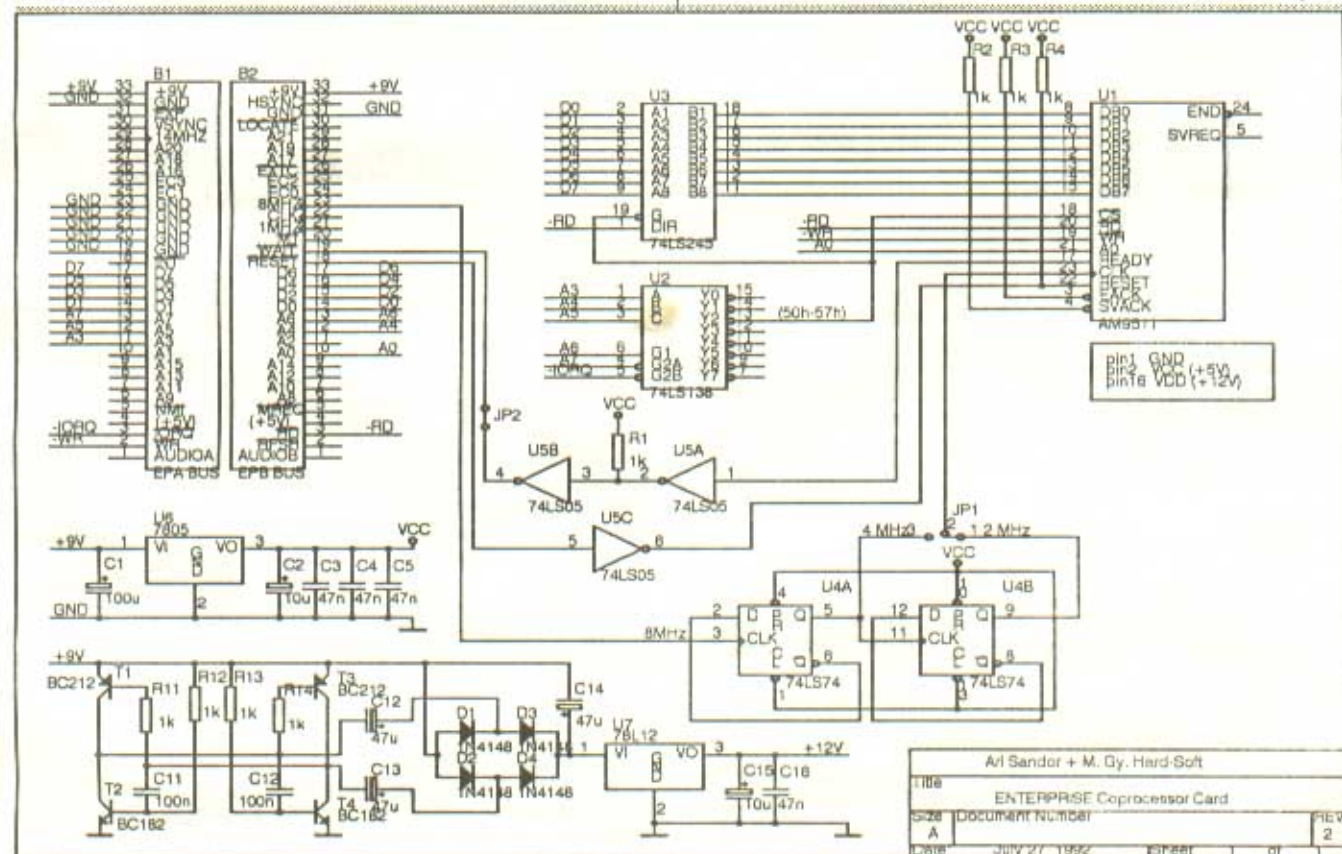
Most már majdnem mindent tudunk, csak az furdalhatja a tisztelt olvasó oldalát, hogy vajh mennyivel lesz gyorsabb a gépe, ha beszeres egy ilyen kártyát? Ari Sándor összeállított egy táblázatot néhány egyoperandusos lebegőpontos műveletre. A számok 10000 művelet idejét jelentik másodpercben.

művelet	APU	BASIC	APU/BASIC
SIN	24	592	24
COS	24	842	35
TG	31	1704	54
SQRT	7	1569	224 (!)
ASIN	39	2600	66
ATAN	32	860	27
LOG	28	1390	49
EXP	28	930	33

Az átlag 64-szeres sebességszorzó (az APU javára).

Végül, akit bővebben érdekel az APU programozása, az Intel Component Data Catalog 1982 című kiadványból tájékozódhat, a 8231A címszó alatt.

Mészáros Gyula



IBM kompatibilis számítógép-részegységeket, háttértárolókat, lemezeket és egyéb kiegészítőket. Cím: 1132 Budapest, Visegrádi utca 6. Telefon: 112-8604

Programozási verseny: Jöttek, láttak, győztek



Rovatunkban ez alkalommal a pályázati felhívásunkra beérkezett programokat ismertetjük.

Játékprogram

RUDI

Koroknai István, Ivánca

A játékban természetesen Rudi a főszereplő. Rudi fűlig szerelmes egy gyönyörű (2 karakter széles és 2 karakter magas) leányzóba, s hogy bizonyítsa szerelmét sok-sok szívet visz neki.

Minden pálya (összesen 15) 17•10 mezőből áll. A szívek megszerzését nehezíti, hogy itt-ott vigyorgó halálfejek lesnek rájuk. Az igazi nehézséget azonban nem ez jelenti: Rudi ugyanis nem léphet rá olyan mezőre, amelyen már járt, így a játék közben gondolkodnunk is kell. Annyi könnyítést azért ad a program, hogy az egyes sorokat (amelyben Rudi éppen álldogál) jobbra-balra mozgathatjuk. Persze Rudinak igyekeznie kell a szívek összeszedésével, mert meglehetősen kevés idő alatt kell megszereznie a szíveket, és minden pályán oda kell rohannia kedveséhez.

A programban használt figurák, karakterek szépen ki vannak dolgozva, a színek összeállítása is megfelelő. Igazi "home computer-esre" sikerültek a hangeffektek.

Persze azért van (lenne) mit javítani a játékon. Amikor elindítjuk azt, hosszú másodpercekig csak üres képernyőt látunk, így az az érzésünk volt, hogy a program kiakadt. Nem került volna semmibe egy két megnyugtató üzenet kifirása. A másik furcsaság, hogy ha egyszer már összeszedtük az összes szívet, a program nem tesz ki többet, azaz a hátralévő pályákon csak annyi a dolgunk, hogy szívünk hölgyéhez menjünk. Mi egy angol gépen próbáltuk ki a programot, s annál ha letelt az idő, egyszerűen véget ért a program futása, s azt csak újabb betöltés után tudtuk elindítani. Miután pedig végigmentünk az összes pályán, összeszedtük az összes szívet, gratulációképpen összeomlott az LPT. Hiányzik a pontszámok játék közbeni kijelzése is.

ÚJ ROBINSON

Sási Péter, Sárospatak

A program egy szöveges kalandjáték: Hajótörést szenvedtünk, a vihar egy szigetre vetett bennünket, s innen kell hazajutnunk mindenféle tárgyak és egy kapzsi varázsló segítségével.

A program átdefiniálja a gép karakterkészletét olyanra, mint amilyen a folyóírás (script), grafikát nem használ. Ezt akaratán kívül neheztésnek szánhatta a szerző, mert azt érte el vele, hogy hosszas szemeröltető olvasás után csorgott a szemünkből a könny. Azt viszont köszönjük, hogy megkaptuk a játék megoldását, így nem kellett hosszú hetekig a szigeten maradnunk.

A szerző fogalmazási stílusa nagyon tetszett. A program használatát jelentősen megkönnyíti, hogy egy menüben mindig megjelenik, hogy mit vehetünk fel, merre mehetünk stb. Nem árt, ha a játékos tájékozott a földrajzban, mert a program Egri Jánoshoz hasonlatosan kérdéseket tesz fel (pl. melyik a leg-hosszabb folyó vagy alagút stb.).

MINER

Bodnár Tamás, Tatabánya

A játék címe sokat elárul, egy ún. pack-man típusú játékról van szó. Egy bányában vagyunk, ahol oldalra és lefelé áshatunk, illetve a képernyő két szélén elhelyezett létrákon fel-le mászkálhatunk. A programban ellenfelünk egy bányarém, aki helyes kis folyósok ásása közepette felénk tart, hogy befalja a kis figuránkat. Ügyelnünk kell arra is, hogy a már kiásott járatokba beleeshetünk, ami nagy zuhanás esetén halálos is lehet.

A program a karakterek átdefiniálásával karakteres képernyőn fut. A játéktér megjelenésekor a program frója által készített, leginkább egy skót népdalra hasonlító zenét hallhatunk. Ha azonban elindítjuk a kis bányászunkat, a zene abbamarad,

és csak egyszerű hangeffektek szólnak tovább.

A játék meglehetősen egyszerű, de szerzőjéről tudni kell, hogy kezdő a programozásban, s ez az első nagyobb lélegzetű programja. Ha ezt nézzük, igen szép teljesítményt nyújtott.

MASTER MEMORY

Big-Eared Soft, Szentendre

Memóriánkat beállításunknak megfelelően próbáló program. Egy 10•10-es pályán rejt el a gép betűket. Két azonosat kell találnunk a koordináták megadásával. Ha a két betű azonos volt, jutalompontot kapunk. Ha nem volt egyforma a két betű, akkor azokat ismétetlen elrejtja a gép, s valamelyik következő menetben - emlékezve rájuk - azonosíthatjuk, felhasználhatjuk azokat.

Megadható a kártyák, a játékosok száma, s a számítógépet is bevonhatjuk a játékba. Nagyon tetszik, hogy a játék állását kimenthetjük, s azt később visszatölthetjük.

A program szépen megszerkesztett grafikus képernyőn fut, és J.S. Bach zenéjét sítolja - hamisan.

CYCLOPS

Ványi Péter, Nagyvisnyó

A Cyclops táblás, logikai játék. Célja, hogy a táblán látható golyók közül (meghatározott levételi sorrend után) csak egyetlen maradjon. Az első golyót egyszerű eltüntetni, hiszen bárhova léphetünk. Ezután azonban már csak úgy léphetünk valamelyik irányba, hogy átugrunk egy másik golyót. Ekkor az átlépett golyó eltűnik. Ha ügyesek vagyunk, a végére egy golyó marad, és jön a következő szint. A golyókból háromfajta van: az átlagos, a bonus (több pontot ér a levétele) és a time (levételekor az idő visszaáll az indulási értékre.)

Az egyszerű szabályokra épülő játék kivitelezése mind programozástechnikailag, mind esztétikailag magas színvonalú: kiváló példa arra, hogy Basic-ben is lehet grafikus képernyőn futó és (relatív) gyors játékot írni.

CSERNOBIL

Zalka Béla, Budapest

A programra nem igazán illik a "játék" szó, itt egy atomerőmű szimulátorról van szó (a reaktor 440 MWatt-os, ekkora Pakson a BBEP-440-es is). Aki valamennyire tisztában van egy atomerőmű felépítésével, működésével, és a programra rászánja az időt, kellemesen szórakozhat a szimulátorral.

Adott a reaktor, amelyet a kontrollrúd ki/beugdosásával tudunk szabályozni (ha leeresztjük, az lefékezi a reakciót, csökken a reaktor teljesítménye, kevesebb hő keletkezik), a hőt a primerkör vezeti el a reaktorral a primerpumpa segítségével. Ezt az elvezetett hőt hőcserélőn keresztül a szekunderkörbe juttatjuk, itt a szekunderpumpa hajtja a kört. A szekunder kör működési a generátort. A programban a kontrollrúd kiemelését, a primer-, a szekunder- és a vészűtőpumpa teljesítményét kell szabályoznunk (ez utóbbi közvetlenül a reaktort hűti vész esetén). Ezután a szimulátor kijelzi, hogy ezek hatására másnapra milyen változások (hőmérsékleti és teljesítményadatok) lennének az erőműben. Mindezeket természetesen egy blokkvázlaton láthatjuk. A játék addig megy, míg el nem fogy a hasadóanyagunk, ekkor a gép értékeli a teljesítményünket. A program megnyerte tetszésünket, de van vele egy nagy bajunk: szimulátorhoz (kimondatlan törvény) dokumentációt kell írni, ehhez pedig nincs. (Egy tütő példa: az IBM PC-n használható Flight Simulator 4.0-hoz közel 500 oldalas leírás készült.) Ha egy átlagos felhasználó kezébe kerül ez a jól sikerült szimulátor, akkor két percen belül megnyomja a RÉSET gombot (felrobbantja a reaktort), mert nem érti a működését.

MEMÓRIA

Zalka Béla, Budapest

A játék tartalma teljesen megegyezik a Master Memory-ével, csak ennek minden vonatkozásban sokkal gyengébb a kivitelezése.

GYORSÍTÁS

A programfutás sebességét gyorsíthatjuk a következő két lehetőséggel:

1., A DAVE chip 0BFH portján kikapcsoljuk a várakozást a memóriáhozáférési műveleteknél. Adjuk ki az

OUT 191,12

parancsot. Hatására a futásidő kb. 86%-ra csökken.

2., A megszakítást a kritikus helyeken kikapcsoljuk, majd helyreállítjuk.

POKE 56,201 (A futásidő kb. 81%-ra csökken.)

POKE 56,245 (Helyreállítás.)

A két eljárást együttesen alkalmazva kb. 70%-ára csökkentjük a futási időt. Ezek a beállítások bármely nyelvből használhatóak.

Hsoft

FEJBEÁLLÍTÁS

Mi, Enterprise tulajdonosok a legkritikább esetben találkozhatunk a LOAD ERROR... üzenettel. Ha mégis ilyen üzenetet kapnánk, megoldást jelenthet a magnó fejének beállítása. Íme, a beállítást támogató program!

10 SET STATUS OFF

20 PRINT 'ÁLLÍTSD BE A MAGNÓ FEJÉT!'

30 PRINT 'JÓ HELYZETBEN A JOBB FELSŐ SAROKBAN'

40 PRINT 'EGY CSÍK JELENIK MEG'

50 POKE 47360,100

60 SPOKE 255,15026,11

70 RUN "

Pirosft

SZEDD MEG MAGAD!

Butuza Tamás, Kaba

A szöveges kalandjáték célja, hogy az összeszedett tárgyak eladásával minél több pénzt gyűjtsünk össze. A program röviden leírja a helyszínt, majd egy-két szavas parancsokat adhatunk ki. A program 38 szót ért meg, megjelenése nagyon kezdetleges (csak szövegeket küld, képeket nem rajzol).

(Megjegyzés: Játékot (címe: Zsuga-Bubus) küldött a Pirosoft is, de kazettájuk CRC hibás volt. Másik versenyzőnk, Mahner Lajos szerint "Drog-Team" című szöveges kalandjátéka csak német gépen fut a VLOAD miatt. Nekünk azonban német gépen sem indult el a program, mert nem tudott a gép megfelelő számú videolapot nyitni. Furcsa.)

Demók

ELITE-DEMO

G. O. Crew, Budapest

A LPT-s demó négy részből áll. Az első egy igényes címlap a Golden Orb Crew-ről, itt úsznak mindenféle közlemények, üdvözlések. A második részben Mandelbrot-képeket láthatunk. A harmadik részben az ENTERPRESS címlapját és a kiadó címét mutatják meg a demó szerzői. A negyedik részben olvashatjuk a szerzők címét stb. A demó készítői valószínűleg nagyon kedvelik Jarre zenéjét, mert leginkább az ő szerzeményeit hallhatjuk a demó alatt, a Dave hangzásával.

CLASSMATES-DEMO

Classmates Software Studio, Budapest

A program készítői egy igen egyszerű demót küldtek. Induláskor Tut-Anch-Amon fáraó maszkja tűnik fel, majd a szerkesztőség és a kiadó címét olvashatjuk. Ezután három, szinuszosan süllyedő-emelkedő "rúd" között egy negyedik (piros-fehér-zöld színű) mozgó rúdon mindenféle, a lapot dícsérő (zavarba ejtő...) mondatok olvashatók. Kár, hogy miután lefutott az összes felirat, csak az üres rudak mozognak, de emellett nincs semmi látványos.

Ilyen még a neppereknél sincs...

SPRED release 1.5

Felhasználóbarát Entersprite kompatibilis sprite editor

- Tömértelen funkció
- Pull-down menürendszer
- Esztétikus kivitel
- Exdos használhat
- Beépített help
- Magyar nyelvű .WP leírás

Mindaz gyorsan, gépi kódban!

Ára csak 299 Ft!

Befizetésedet rózsaszínű postautalványon várjuk. Ha nem küldesz 5.25"-os lemezt/kazettát, akkor még 40 Ft-ot adj az árhoz. A postaköltség a program árában benne van.

Cím: ARSS, 1399 Budapest, Pf. 701/334.

SPRED r1.5 ... és leesik az állad.

Vége a Sinclair uralomnak!

Megalakult az Országos ENTERPRISE Klub.

Mindenki jelentkezzen, aki lépést akar tartani gépe fejlődésével!

Kérje részletes tájékoztatónkat válaszborítékkal!

Tagtoborzó: Silye Gabriella

5358 Tiszaörvény, Rákóczi út 4.

NOVA-DEMO

Nova & His, Mór

A már a demó kezdetén közli címünket a szerző, és nem maradnak el gyenge zenei kísérlet mellett az elmaradhatatlan találos kérdések, üdvözlések stb. A következő, stroboszkópnak is használható interlace képeken láthatjuk a szerzőt, majd egy PC után különféle női testrészek tűnnek fel természetes valóságukban, jól kivethető közelségben. Az úsztatott mondatokkal kapcsolatban: szerencséje a szerzőnek, hogy nem helyesírási versenyt hirdettünk, mert kizártuk volna.

ORK MEGADEMO

ORKsoftware, Ajka

A készítők bemutatkozásként egy pazarul elkészített intro-képernyővel indítanak, a "szokásos" feliratokkal, mozgásokkal. A második rész látványos hullámzást mutat be egy pontmezőn, különböző ábrákkal. Ezután egy elég jó grafikát láthatunk, amelyen éppen egy vizisárkány fal be egy vitorláshajót. A harmadik rész színbemutató: mindenféle alagzatok csillognak-villognak (némelyikük repülés közben) a képernyőn. A következő rész két képből áll, amelyek Amigán készültek. Az elsőn egy valami van törrel a praclijában. A másodikon is egy valami van, ez rõhög is.

Az ötödik részt a szerző csak azért írta, hogy hosszabb legyen a demó (ahhoz képest nagyon szép az az effekt, ahogy szétporlasztja a betűket!). A grafikus képen a szép "ORK soft" felirat felett afféle "agyfolyást" olvashatunk. Jellemző a szerzőre, hogy ebben a részben jutottunk az eszébe, itt esik szó a lapról... Ejnye, ejnye!

A következő rész elején ismét egy álló grafikát láthatunk (egy fantasztikus táj felett reptől ORK-ék zűrűhájója.) Ezt követően csíkozódó (adáskimaradás?) színorgia mellett egy kezdő elektromos gitáros játékát élvezhetjük.

A "The End" partitúra ismét egy álló grafikával indít (ezen a híres Kéknyakú Lila tart a kezében egy marsallbotot, és éppen azt nézi, ahogy egy föliázott franciaagy gyászzenei kísérlet mellett felszáll a cupákos Zöld-Nyálka tóból), majd a készítők nevét olvashatjuk.

(Megjegyzés: Basic demót küldött Butuza Tamás, de kazettája CRC hibás volt.)

Eredményhirdetés az utolsó oldalon!

Asterix and the magic cauldron

Időszámításunk előtt 50-ben Julius Caesar a római légiók élén megtámadta Galliát. A szervezetlenül és elszigetelten védekező gall törzsek ellenállását a technikai fölény és a túlerő rövid időn belül felmorzsolta, az egyetlen nagyobb létszámú haderőt felvonultató *Vercingetorix* vezér Alesia-nál döntő vereséget szenvedett. (Kész történetész ez az ipse! - EPY)

Egész Gallia a rómaiak fennhatósága alá került. Azaz csak majdnem az egész: egy kis falu, *Gallfalva* váltig ellenáll a hódítóknak. A falu főnökének, *Hasarengazfixnek* legvitézebb harcosa *Asterix*, az alacsony növésű, de annál furfangosabb észjárású harcos. *Asterix* a terveithez szükséges fizikai erőt egy különös létyóból, a *Magicoturmix* druida által főzött varázsitalból nyeri, ami sajnos nem állandó hatású. *Asterix* elválaszthatatlan barátja a darázdereke *Obelix*, aki gyermekkorában belesett a varázsitalos kondérba, nála az emberfeletti erő állandósult állapot (a játékban sajnos csak fiktív szereplő lesz)

Egyszer *Magicoturmix* druida varázsital helyett pálinkát főzött, de úgy látszik, eme nemes nedű készítéséhez nem értett annyira, mert a kotyvaléktól a varázsüst szétrobbant és darabjai szerteszét repültek. Mivel "Vasdedény" bolt nem található a közelben, *Asterix* indul el, hogy összeszedje a varázsüst darabjait.

Hát ennyi lenne röviden, tömören a képregény és a program története, amit *Spectrumra* először a *Melbourne House* adott ki 1986-ban.

A program betöltése után megjelenik a demo-kép, amelyen megismerhetjük az összerakandó varázsüstöt és a játék szereplőit (*Asterix*, *Obelix*, római katona, vaddisznó). A játék a belső vagy a külső joy tűzgombjának megnyomására indul. Ha a játék közben megnyomjuk a [STOP] billentyűt, a program kiírja: "GAME OVER" és új játékot kezdhetünk. A játékban egész szép képernyők vannak, melyeket a gép képrészekenként épít fel. Ez nem túl praktikus, mert amikor átmevünk egyik képernyőről a másikra, 4-5 másodpercet várni kell, míg a gép édesdeden rajzolgat.

A képernyő bal felső részén látható élelmiszerkészletünk, sült vaddisznó képében. Kezdetben 5 darab áll rendelkezésünkre, de az idő múlásával darabonként fogynak. Ha minden vadmalacunk elfogyott - függetlenül az életek számától - a játék véget ér, mert spontán módon éhenhalunk. Készletünket az erdőben szaladgáló vaddisznók elejtésével gyarapíthatjuk, a készlet max. 9 darabból állhat. A képernyő jobb felső részén, a pontszámától balra levő *Asterix*-fej mellett látható életeink száma, amely kezdetben szintén 5. Életet akkor veszhetünk, ha a rómaiakkal vagy vaddisznókkal való küzdelemben energiánk elfogy. Ezt az energiát egyszer megnövelhetjük, ha az életektől balra látható varázsitalos butykost kiürítjük ("hosszú tűzgombnyomás").

A játék a gall faluból (GAULISH VILLAGE) indul. A játék egyes helyszínein - pl. a falu főterén - találkozhatunk néhány csomaggal, amelyek némi elemőzsiát illetve rőzsét tartalmaznak. Ezek felvételéért 50 illetve 100 pontot szerezhetünk. Mikor abba a pozícióba érünk, amelyben a csomag van, a program a képernyő közepén egy ablakot nyit, amelyben felvehetjük a csomagot. Ez a későbbiekben is így fog történni: ha nemcsak vándorlás, hanem valamilyen cselekmény (harci tárgy felvétele) zajlik, az a képernyő közepén egy ablakban látható.

Miután összeszedtük a faluban a csomagokat, induljunk el keleti irányba. Az erdőbe (FOREST) jutunk, ahol feltölthetjük élelmiszerkészletünket, ugyanis néhány vadmalac kóricál erre. A rőfik elejtéséhez meg kell küzdenünk velük, ami a következőképpen történik: Ha az ablak oldalán megállunk, a malacka hamarosan felénk vágódik. Vad rohamát egy tockossal fogjuk megtörni, mert ha elér minket, az ablak oldalán fehérrel jelzett energiánk csökken. *Asterix* háromféle támadási formát ismer: [tűz+le]=hasbarúgás, [tűz+fel]=orrbaverés, [tűz+jobb-ra]=leütés. A vadmalacok ellen legjobban a leütés használható. A támadó malacka minden tockos után visszaugrál egy keveset, emiatt az élet dolgain, de nem túl tanulékony állat, mert

mindaddig újra támad, amíg véglegesen le nem esapjuk (elfogy az energiája). Ekkor azonban égnek vetett lábakkal elnyugszik, a készlet számlálója pedig villogva jelzi, hogy eggyel több malacunk van. Vigyázzunk, csak a négylábon közlekedő malackák fogyaszthatóak, a kétlábúakkal felesleges harcra keverednünk, csak az energiánkat poecékoljuk. Miután feltöltöttük élelmiszerkészletünket, haladjunk tovább kelet felé, egészen Totorum római táborig (CAMP TOTURUM). A kapuban egy római katona áll őrt, akit finoman arrébb kell hessegetnünk. A küzdelem hasonlóképpen zajlik a vadmalacoknál megismertekhez, bár a rómaiak folyamatosan támadnak, könnyen elveszthetjük életeinket. Ha a rómaiak sikerül levernünk, érdemei elismerése mellett eltávozik a Mennyekbe, mi pedig folytathatjuk utunkat továbbra is keleti irányba.

Továbbra is Totorum tábor területén vagyunk. Az első sátor mellett forduljunk délre! A helyszínen található egy kulcsot, amit fel kell vennünk. Ezután menjünk vissza egészen az erdő azon részéig, ahol az első vadmalacokkal találkoztunk. Innen északi irányba fordulunk és elmevünk Compendinum táborig. Miután - néhány római kivert fogaival övezve - bejutottunk a táborba, menjünk keletre, ahol megtaláljuk az üst első darabját. Miután felvettük, fogassuk el magunkat az őrről. Ez a következőképpen történik: ha találkozzunk vele és a program megnyitotta a képernyő-ablakot, ne csináljunk semmit. Az őr a közelünkbe jön, mutogat egy darabig, majd az ablak tetején megjelenik a "SURRENDER" (megadta magát) felirat.

Asterixet Rómába hurcolják és az éjszaka már egy cellában éri. A Totorumban felvett kulccsal kinyithatjuk a cella ajtaját, és kimehetünk a folyosóra. Miután agyoncsaptuk az őrt, menjünk be a nyugati irányban levő helyszín első cellájába. Itt megtalálhatjuk az üst második darabját. Ezután egy ideig nincs semmi teendőnk, meg kell várnunk a hajnalt. Amikor a nap első sugarai beszűrődnek a cella ablakán, kimehetünk az ajtón (ha a hajnal a másik cellában virrad ránk, nem tudjuk megszerezni az üst második darabját, mert az arénából nem tudunk visszajönni a börtönbe).

Mielőtt a cellaajtón át kijutunk az arénába - ha eddig még nem fogyasztottuk el -, érdemes meginni a varázsitalt. Az arénában ugyanis egy kisebb római légió akarja rajtunk bemutatni az élveboncolás nevű történetét. Jobbról folyamatosan egy-egy római csörtet be, akit célszerű egy ütással agyoncsapnunk, mert különben ő teszi ezt velünk. Külön kedvesség a programozóktól, hogy mozogni sem tudunk, egy helyben kell várnunk a támadókat. Van néhány kevésbé militarista beállítottságú római is, ezek a mellékajton távoznak. A tortúra addig tart, amíg 10 római szét nem cincáltunk. Ezután felvehetjük az üst harmadik darabját. Távozzunk a küzdőtérrel, a kapun keresztül kijuthatunk Róma főutcájára. Itt először déli, majd nyugati irányba kell mennünk. A két ház között levő fák között megtalálhatjuk az üst negyedik darabját.

A Diadalúton keresztül menjünk ki Rómából - itt ismét találkozhatunk *Obelix* barátunkkal -, majd Aquarium táboron keresztül menjünk el az erdőbe, ahol a térképen jelzett helyen megtalálhatjuk az üst ötödik, egyben utolsó darabját is. Ütközben még el kell vezetnünk néhány római katonát az enyészet útjára, de ez már nem lesz olyan nehéz. Miután felvettük az üst utolsó darabját, a gép kifejezi kézzemeleg gratulációját, kiírja az általunk elért pontszámot, majd visszaadja a demo-képet és lehet újra próbálkozni.

Hát ennyi a megfejtés. Aranyos program, csak a zenét fejtették ki belőle. Egyszer azért érdemes megfejtetni! *Lola*

ASTERIX	
Grafika:	7
Zene/FX:	0
Játszhatóság:	3
Összhatás:	5

DIZZY 3.5

Mióta a "Dizzy-sorozat" annak idején befejeződött, nem találkozhattunk a kis tojáskával az Enterpress hasábjain. A sorozat intenzíven most sem folytatódik (sokak bánatára-EPY) de Dizzy-leírások időközönként fognak megjelenni (pl. *Dizzy IV. Magicland Dizzy; Spellbound Dizzy*).

Miután az OLIVER-fivérek megírták a Dizzy harmadik és negyedik részét, és mielőtt a negyedik részt a CODEMASTERS kiadta volna, gondoltak egyet, hogy miért ne nehezíték a játékosok dolgát, megírták a Dizzy-sorozat "Karácsonyi Katasztrófa-különkiadását", a Dizzy három és felet. Ti. túl egyszerű lenne az, ha *Fantasy World*-ből rögtön *Magicland*-be jutnánk.

A program története az, hogy miután kiszabadítottuk Daisyt és hazavitük, az élet nyugalmasan zajlott tovább Tojáslandben. Legalábbis egy ideig. Aztán Dizzynek üzleti ügyben külföldre kellett utaznia. Amíg az üzleti tárgyalásokon ült és a szerződéseket írogatta alá, addig az elpusztított Zaks varázserejét összeszedve újból életre kelt, és elrabolta Tojásland összes lakóját. Egyedül a kis Danny tudott megmenekülni egy furcsa "antigravitációs csizma" segítségével.

Célunk az, hogy megtaláljuk a *Magicland*-be vezető utat, hogy a sorozat következő részében ki tudjuk a tojásembereket szabadítani. (Ez a logika! Ha nem írja, nem is jövök rá!-EPY) Ennyit a történetről, és most következék a játék!

A starthelyről menjünk jobbra, ugorjunk fel a platóra, onnan ugorjunk tovább jobbra a lépcsőre. Menjünk föl, rajta végig. Nocsak, nocsak... Itt lebeg Danny, és rajta van a varázscsizma! Beszélgessünk csak egy kicsit vele!

- Segítség...! Én vagyok az, Danny! - kiáltja kétségbeesetten.
- Mit csinálsz itt fent??? - tátja el száját csodálkozásában Dizzy.

- Felvettem ezt a fura csizmát, és felrepíttem! -rebeigi tovább sápadtan.

- Danny, hol vannak a többiek? - gyanús Dizzynek ez a nagy csönd.

- Zaks, a gonosz varázsló visszajött! Mindenkit elrabolt, és elvitte őket a bűvös királyságba!

- Az ördögbe is! Valahogy el kell jutnom oda, hogy megmentsem őket!

- Dizzy! Miért nem próbálsz engem valahogy lehúzni innen? Segíts!!!

Miután befejeztük a kedélyes társalgást, hagyjuk ott lebegni barátunkat, és az utolsó előtti lépcsőfokról ugorjunk balra. Dizzy és Daisy házának platóján landolunk. Menjünk a ház tül-ső végébe és nézzük meg, mi van ott. Nocsak, egy indítókar! Micsoda felfedezés!!! Vegyük fel, majd a plató széléről ugorjunk balra. A starthely melletti sziklán kötünk ki. Vegyük fel az ott látható hosszú kötelet. Ezután menjünk jobbra! Itt egy nagy víz állja az utunkat. Sebaj, ússzunk át rajta. Hoppá! Hát ez nem jött össze. Dizzy nem tud úszni. (Pedig három és fél rész alatt már megtanulhatott volna! - EPY) Akkor próbálkozzunk más-hogy: ugorjunk át. Ezt úgy tegyük meg, hogy menjünk vissza oda, ahol Danny lebeg, és a legfelső lépcsőfokról ugorjunk el jobbra. Jé, ez a víz nagyobb, mint gondoltuk! Újra megfulladunk... Valahogy meg kellene szerezni Dannytól azt a varázscsizmát. Egyszerű: menjünk el a vízpartra, ott van egy nagy fa. Álljunk meg a jobb oldalán, és használjuk a hosszú kötelet. Ekkor Dizzy földobja azt Dannynek, és ő lemászik rajta. Ezután következik még egy kis diskurzus:

- Köszönöm, Dizzy! - hálálkodik Danny - Itt a csizma!

- Biztonságos?

- Persze. Te sokkal nehezebb vagy, mint én!

A társalgás után Danny leteszi mellénk a csizmát. Vegyük fel. De jó, már tudunk vizen járni! Na, azért nem egészen, ezzel csak nagyot lehet ugrani. Most már átugorhatjuk a vizet. Menjünk vissza föl, és tegyük meg. Sikerült! Egy felhő vagy ködfolt tetején landoltunk, amin elkezdünk süllyedni, majd a vízbe esünk, és megfulladunk. Ezért ahogy megérkezünk a felhő tetejére, ne álljunk meg, hanem AZONNAL (!) ugorjunk tovább jobbra. Így elkerülhetjük az uszodát.

Miután ezt sikeresen végrehajtottuk, ugorjunk fel a sziklára, melyen a fátyla van úgy, hogy felugrunk a szikla legalsó fokára, onnan pedig a legfelsőre. Vegyük fel az itt levő rövid kötelet, és hagyjuk itt helyette az antigravitációs csizmát. Ugráljunk le, majd menjünk fel a jobboldalt látható fülkébe, majd ott hasz-

náljuk a hosszú kötelet, mely a röviddel összekötődik indító-kötéllé, amit használunk az ott levő gép hátsó kerekénél. (Ez a gép egy teleport-autó, mely átszállít minket *Magicland*-be.-EPY) Okos vagy, EPYkém, de maradj csöndben! Tehát ekkor Dizzy a kötelet rátekeri a fogaskerékre, majd használja az indítórudat, mire a gép beindul. Ezután vissza kell menni a fülkébe és teleportálódunk Zaks országába...

És itt a vége a játéknak, miután a gép kírja szövegét a játékosnak: *Jól van, eljutottál Varázsországba!!! Most már csak meg kell mentened mindenkit... És valahogy haza kell jutni...! A kaland folytatódik a Dizzy IV.-ben (MAGICLAND DIZZY)! Bye bye!*

A program az átlagos Dizzy-színvonalat követi, csak az a baj, hogy a zenét lespórolták róla, és effektből is kevés van. *Lola*

DIZZY 3.5	
Grafika:	6
Zene/FX:	0
Játszhatóság:	6
Összhatás:	4

CHEAT-MIX

R-TYPE

[R] 1B00 [ENTER] BFFF [ENTER] R-TYPE [ENTER]

Last address: 79BE

(A 2. filet kell betölteni!)

[M] 31C6 [ENTER] 00 [ESC]

[M] 2689 [ENTER] X [ESC] (X=életek száma)

[S] 1B00 [ENTER] 79BE [ENTER] R-TYPE [ENTER]

GUN-RUNNER

[R] 12E7 [ENTER] BFFF [ENTER] GUNN.PRG [ENTER]

Last address: AFFF

(A 3. filet kell betölteni!)

[M] 6E13 [ENTER] 00 [ESC]

[M] 866B [ENTER] 00 [ESC]

[M] 8676 [ENTER] 00 [ESC]

[M] 7013 [ENTER] 00 [ESC]

[S] 12E7 [ENTER] AFFF [ENTER] GUN.PRG [ENTER]

AFTER THE WAR

[R] 2000 [ENTER] BFFF [ENTER] WAR.PRG [ENTER]

Last address: BFFE

(A 2. filet kell betölteni!)

[M] 7EF9 [ENTER] 00 00 00 [ESC]

[M] 8C8C [ENTER] B7 [ESC]

[M] 8C69 [ENTER] C9 [ESC]

[S] 2000 [ENTER] BFFE [ENTER] WAR.PRG [ENTER]

SATAN 2

[R] 1600 [ENTER] BFFF [ENTER] SATAN2.PRG [ENTER]

Last address: BAFF

(A 3. filet kell betölteni!)

[M] 6A01 [ENTER] B6 [ESC]

[M] 6A14 [ENTER] B6 [ESC]

[M] 6907 [ENTER] B6 [ESC]

[S] 1600 [ENTER] BAFF [ENTER] SATAN2.PRG [ENTER]

ASTERIX AND THE MAGIC CAULDRON

[R] 1000 [ENTER] BFFF [ENTER] ASTERIX.PRG [ENTER]

Last address: AFFF

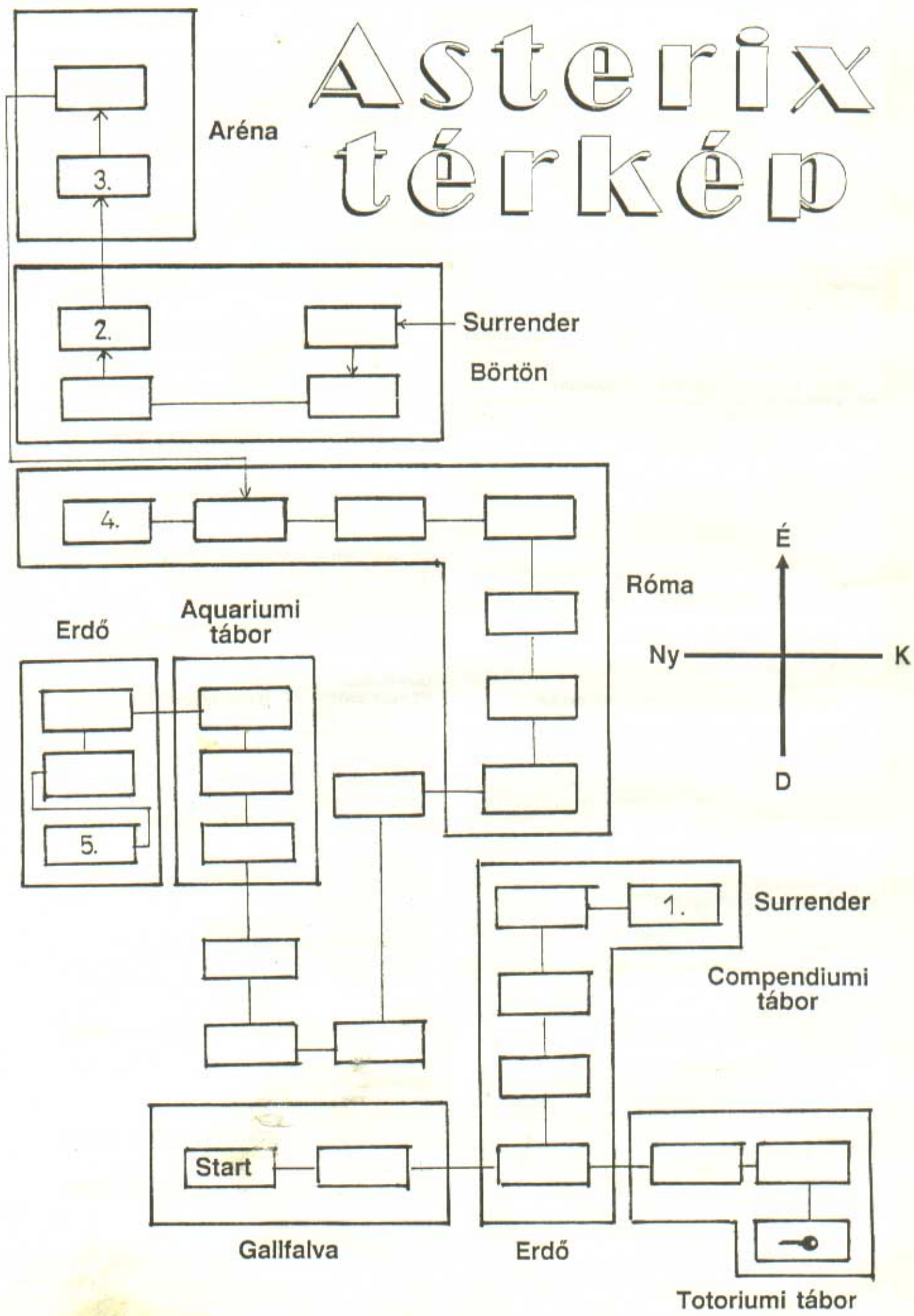
[M] 3F76 [ENTER] 00 [ESC]

[M] 3F36 [ENTER] 00 [ESC]

[S] 1000 [ENTER] AFFF [ENTER] ASTERIX.PRG [ENTER]

Lukács Árpád

Asterix térkép



Postafiók 334

Az idő csak telik - mintha egyre gyorsabban múlna -, és a levelezési rovat adóssága egyre nagyobb. A terjedelem korlátozott, ugyanakkor egy-egy levél olyan problémát vet fel, amellyel behatódobban kell foglalkozni, természetesen a sorára váró többi levél rovására. Most próbálunk kicsit törleszteni az adósságból; nagyon régi levelek is várnak válaszra...

Pankasz Gábor 13 éves budapesti olvasónk, akinek egyébként a *Tippék-trükkök* és a *Könnyed műfaj* a kedvenc rovata, érdeklődik, hogy lehet-e a szekresztől vagy a kiadótól programot rendelni. A pozitív válaszban reménykedve azonnal meg is rendel a *Grange Hill* nevű programot.

Kedves Gábor! Ha fellapozod az ENTERPRESS korábbi számain, biztosan megtalálod sokszor hangoztatott állásfoglalásunkat - akár a sokszor *ars poetica* hangú és stílusú szerkesztőségi cikkek egyikében, akár itt, a levelezési rovatban egyik-másik válaszban - a programok adás-vételével kapcsolatban. A válaszuk: nem. Volt ugyanakkor egy próbálkozásunk a Programkötő Szolgálat beindítására. Ez a szolgáltatás elvileg létezik, de fizikailag nem működik. Nem kaptunk olyan programokat, amelyek igazából megfeleltek volna. Persze be kell vallanunk, hogy kicsit túlbecsültük képességeinket, mert a szolgálat szervezése közben eszméltünk csak rá, hogy milyen irgalmatlan nehéz egy ilyet megszervezni egy ekkora apparátussal. Sokat segített volna, ha az itt-ott elszórt felhívásainkra jelentkeztek volna Olvasóink, hogy segítsenek ebben a munkában. A másik baj, amit nem akarunk elhallgatni, hogy mi, szerkesztők, már túl sokat dolgoztunk a semmiért, a lap előállítása önmagában is elég feladatot jelent, nem akartunk egy újabb terhet a vállunkra venni.

Tudomásunk szerint van olyan szerző, aki rövid néhány hónap leforgása alatt 70-egynéhány példányban adta el programját. Ez a szám csak első hallásra tűnik szerénynek, ismerve a magyarországi programvásárlási kedvet, az ENTERPRISE gép helyzetét, ez egész jó eredménynek tűnik. Ugyanakkor nem tudunk segítséget adni egy olyan gyakorlatnak, amely ellentétes meggyőződésünkkel, igazságérzetünkkel, és csak egészen mellékesen, az érvényes jogszabályokkal is; ez pedig a mások által készített - akár megvásárolt, akár pedig barátitól, ismerőstől kapott - program továbbadása pénzért. Ennek a meggyőződésünknek nem csak hangot adunk, hanem az ezzel járó - igaz, csak jelképesnek tekinthető - anyagi áldozatot is vállaljuk, nevezetesen azzal, hogy nem közöljük azt a hirdetést, amelyik ilyen "szolgáltatást" reklámozna. Ez az oka annak, hogy a kért programot nem küldjük, nem küldhetjük el akkor sem, ha az esetleg valamelyik munkatársnál esetleg fellelhető volna. Reméljük, időközben a játékot sikerült megszerezned; ha nem, biztosan akad egy ENTERPRISE-tulajdonos, aki e sorokat olvasva elküldi.

Farkas Gábor ózdi olvasónk írja: *"Még csak nemrég vagyok az ENTEPRESS olvasója, mivel a gépem is új még, és az újságra is elég nehezen bukkantam rá. Ami nem tetszik: az Assembly és a Pascal leírása; aki magasabb szinten foglalkozik a géppel, az amúgy is meg tudja szerezni ezeket. A többi dolog tetszik, különösen a játékleírások és az örökéletkódok. Nem lehetne valamilyik számban megírni, hogy hogyan is kell bevinni az örökéletkódokat? Szerintem sokan vagyunk, akik nem tudjuk. Nem tudnák elküldeni (persze, csak ha megvan) az ALIEN leírását? Elég gyenge eredményt érek el ezzel a játékkal... A géppel aránylag meg vagyok elégedve. Ugyanakkor elhűlve nézem a C-64 tulajdonosokat, akiknek egyetlen kazettán 40 játék is elfér; nekem legfeljebb 18."*

Kedves Gábor! Ma, úgy tűnik, csak Gáboroknak tudunk válaszolni. Nos, köszöntünk, kicsit megkésve, mint ENTERPRISE tulajdonos és sortárs. Az, hogy az újságot késve találtad

meg, sajnós, nem meglepő számunkra, ezen már számtalanszor keseregünk-morgolódunk. Az viszont, hogy mit hagynál ki a lapból, pontosan megegyezik sok más olvasó véleményével, és pontosan ellentétes másokéval; sokan ugyanis elvesztegetett, elpocsékol helynek tartanak minden hasábmillimétert (ahogy ezt újságszerkesztőknél mondani szokták), amit játéokra vesztegetünk. A többség, szerencsére, helyesli azt, hogy minden rétegnek, érdeklődési körnek próbálunk adni valamit, és ennek mindaddig így kell lennie, amíg nem lesz külön újságjuk a csak-játékosoknak és a sosem-játékosoknak. Egyébként a programozási nyelvekbe való bevezetést azért tartjuk fontosnak, mert a hasonló tárgyú könyvek általában a szakemberekhez szólnak, bizonyos alapfogalmak ismeretét eleve feltételezik, így ezek az átlagos rendszerint fiatal - kezdő programozójelöltnek nehezen legyűrhető, érthetetlenek. A sorozatokban igyekszünk a témákat az átlagos felhasználó szemszögéből megközelíteni. Nem elhanyagolható a sorozatoknak az az - olvasóink által sokszor felemlített - hátránya, mégpedig, hogy lassúak, hosszú az idő az egyik folytatástól a másikig. Ez a hátrány, a sors fintoraként, most előnyként jelentkezik, hiszen már a következő számban reagálni tudunk a levelek alapján, ha valami érthetetlen volt, vagy esetleg tévedtünk volna. Nincs az a tankönyv vagy felhasználói kézikönyv, amelyik fejezetről-fejezetre alkalmazkodna, változna az olvasók jelzései alapján. Arról nem is beszélve, hogy nem létezik olyan könyv, amelyik kimondottan az ENTERPRISE jellegzetességeihez alkalmazkodva mutatná meg akármelyik programozási nyelvet vagy egyéb eszközt. Azt is elárulhatjuk, hogy a szerzők csak azt írják le, amit ki is próbáltak. Így az ENTERPRISE-os változat esetleges hibáira, eltéréseire is fel tudják hívni az olvasók figyelmét.

Az örökéletkódok betöltéséről írtunk korábban, de szívesen megismételjük: Be kell tölteni a gépbe az ASMON vagy a SIMON assembler-monitort, majd szó (pontosabban: betű) szerint azt kell tenni, ami a leírásban van. A szógletes zárójelben álló betű vagy szó egy-egy billentyű felíratát jelenti, ezeket olyan sorrendben kell megnyomni, ahogy ott szerepelnek, közben a program által kért értékeket - szintén a leírásból - be kell írni. Ennek során a program be fogja tölteni, majd a végén el fogja menteni a műtét tárgyát, az örök életre ítélt programot; ehhez kezelni kell a magnót, vagy esetleg lemezt kell eszerélni, értelem szerűen.

Végül el kell mondanunk, hogy a C-64-es gép magnója egyedi darab, direkt a géphez illesztve, így érik el a viszonylagos nagy adatsűrűséget. Lehet, hogy a kazetta kapacitása fontos annak, akinek az egyetlen célja, hogy 2000-3000 játékja legyen; aki viszont *használni* akarja a programjait, annak az időtrábló keresgélés elkerülése érdekében célszerűbb nagyon rövid, 5 perces kazettákat beszereznie, és minden kazettára csak egy programot tenni.

Nos, akaratunk ellenére, ennyi válasz fért ebbe a számba. A többiétől további türelmet kér

A felelős szerkesztő

Tudatjuk a Tisztelt Olvasókkal,
hogy az ENTERPRESS kiadása
1992. decemberének végén
megszűnik.

Utolsó, összevont (5. — 6.) számunk
decemberben jelenik meg.

PROGRAMOZÁSI VERSENY

EREDMÉNYHIRDETÉS

A Teszt rovatban közöltek alapján szerkesztőségünk és a kiadó az alábbi munkákat ítélte legjobbnak:

JÁTÉKPROGRAMOK

1. helyezett: **CYCLOPS**
Ványi Péter, Nagyvisnyó
2. helyezett: **RUDI**
Koroknai István, Iváncsa
3. helyezett: **MASTER MEMORY**
Big-Eared Soft, Szentendre

DEMÓPROGRAMOK

1. helyezett: **ORK MEGADEMO**
ORKsoftware, Ajka
2. helyezett: **NOVA-DEMO**
Nova & His, Mór
3. helyezett: **ELITE-DEMO**
G. O. Crew, Budapest

A nyerteseknek gratulálunk! Az első helyezettek 3000 Ft, a második helyezettek 2000 Ft, a harmadik helyezettek pedig 1000 Ft pénzjutalomban részesülnek. A nyertesekkel kiadónk felveszi a kapcsolatot.

mikrovilág

Az ENTERPRESS előző számai korlátozott példányszámban még megrendelhetők a kiadó címén (MÁTRIX Kft. 8000 Székesfehérvár, Zichy liget 10.), vagy megvásárolhatók a Műszaki Könyvtárházban (Bp. VI. ker. Liszt F. tér 9.) és a Fókusz Könyvtárházban (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

ENTERPRISE turbósítás 6 MHz-re, működő perifériákkal, 3.500,-Ft; 320 KB-ra bővítés 3.000,-Ft. Együtt 6.000,-Ft!
Bozai Gábor, 8000 Székesfehérvár, József A. u. 62/A. Tel.: (22) 10-665

Itt az idő, hogy megszerezd a legjobb EP játékokat az ORK Soft-tól! Kaland- és szöveges programok, kiváló grafikával, zenével! Nem Spectrum átiratok! Felbélyegzett válaszborítékért részletes ismertetőt küldünk.
Cím: ÖRKsoft, 8400 Ajka, Kilián Gy. u. 29.

ENTERPRISE rajzolóprogram, 5 grafikai segédprogram, másolóprogram és egy játék minden költséggel csak 488,-Ft-ért. Ha Ön rózsaszínű utalványon a pénzt elküldi, küldjük a kazettát.
SCORPION SOFT, 1399 Budapest, Pf. 701/1217

360 kB-os driver és EP-kontroller 10.000,- Ft-ért eladó!

Cím: Nemeséri Lajos, 8200 Veszprém, Március 15. u. 2/A.

Eladók ENTERPRISE-hoz joystick-illesztők (autofire, infra joystick is megy) utalvánnyal 970,- Ft/db
Abel Imre, 1116 Budapest, Fegyvernek u. 103.

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

HIRDETÉSFELVÉTEL

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

MÁTRIX Kft.

ENTERPRESS

8000 Székesfehérvár,

Zichy liget 10.

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



LEVELEZÉS

A géppel kapcsolatos témákban levelezne:
Bognár Balázs, 9443 Petőháza, Bartók Béla u. 11.
Czibere Lajos, 4027 Debrecen, Fűredi út 1. III/12.
Feczkó Krisztián, 8200 Veszprém, Anyos u. 1/3. Tel.: (80) 29-493
ifj. Márkföldi Béla, 4400 Nyíregyháza, Új u. 28.
Tóth István, 7305 Mecsekpölöske, Szegő u. 1.

KLUB

Budapesti Enterprise klub VSzM közösségi ház
Budapest, XI. ker. Fehérvári út 120.

*** ENTERPRESS Kéthetente az Enterprise számítógépek felhasználóinak *** III. évfolyam 4. szám *** 1992. július—augusztus *** Kiadja a MÁTRIX Kft., Székesfehérvár * Felelős kiadó: Juhász István ügyvezető * A kiadó címe: MÁTRIX Kft., 8000 Székesfehérvár, Zichy liget 10. Telefon:(22) 329-888 Telefax:(22) 329-888 *** Felelős szerkesztő: Ujlaki László * A szerkesztőség tagjai: Hajnal Csaba főszerkesztő, Devisoft, JCVI, Ari Sándor, Bozai Gábor, Haluska László, Lolasoft, Mészáros Gyula * A szerkesztőség csak levélben érhető el! A cím: ENTERPRESS, 1399 Budapest, Pf. 701/334. * Technikai szerkesztő: Szapper László *** Nyomja a Duna Print Kft., Dunaújváros * Felelős vezető: Farkas István *** HU ISSN 0866-1820 *** Terjeszti a Magyar Posta * Előfizethető a HELIUS, 1900 Budapest, vagy a MÁTRIX Kft. címén * Előfizetési díj egy évre 294 Ft, fél évre 147 Ft. *** Utlós számunk decemberben jelenik meg. *** Az ENTERPRESS-ben közreadott információk célja az, hogy segítsék, tudniakokkal lássák és a gép felhasználóit. A közölt programokat, kapcsolási rajzokat, leírásokat mindenki szabadon felhasználhatja, de tilos azokat a kiadó írásbeli engedélye nélkül másolni, terjeszteni. * A szerkesztőség kéziratokat nem őriz meg, és nem küld vissza; továbbá külön levelezésre nincs módja. *** ENTERPRESS © 1992 MÁTRIX Kft. *** 8000