

ENTERPRESS

KÉTHAVILAP AZ ENTERPRISE SZÁMÍTÓGÉPEK FELHASZNÁLÓINAK

Egy megvalósult ötlet

Minden újság működésének fontos mércéje az olvasói rovat, amely egy részének most a címlapon adunk helyet. Ebben jelenhetnek meg azok a vélemények, amelyek lényeges formai és tartalmi követelményeket fogalmaznak meg az olvasók részéről és amelyek alapvetően meghatározhatják a kiadvány arculatát is.

Most egy olyan levélrészlet közreadását tesszük meg amely — véleményünk szerint — minden érdekelt (elsősorban az olvasó) hasznára válhat.

Íme a levél:

„Tisztelt Szerkesztőség!

Olvasom az ENTERPRESS újságot és az a véleményem, hogy erre az újságra minden ENTERPRISE felhasználónak szüksége van.

A programozni kívánóknak ötletek, adatok, a játszani kívánóknak játékleírások, örökéletkódok teszik hasznossá a lapot.

Sajnos minden megjelent számban azt olvashatta az olvasó, hogy a lap gyenge életű.

Az előfizetők kevés száma azzal indokolható, hogy kevesen ismerik az ENTERPRESS-t.

Aminek a reklámok hiánya lehet az egyik oka!

Felmerült bennem egy gondolat:

Mivel egyedülállósága miatt minden felhasználónak jól jöhet, illetve jól jön ez a lap, áldozhatnánk rá egy kis összeget.

Alapítványt lehetne létrehozni és ennek értékét reklámcélra lehetne fordítani! Pl.: a Dallas előtt (hi-hi). Az ország különböző pontjain élő 20000 felhasználó közül többen szerezhetnek tudomást az újság létezéséről.

Az összeg amire gondoltam egy-kétszáz forint, és ha ezt az összeget a 2000 jelenlegi előfizető feláldozná, igen szép kis összeg állna rendelkezésre a lap életben tartására, reklámozására!

Mindezekkel fordulnék ísztelettel a szerkesztőségen keresztül az olvasókhoz.

Köszönöm, ha meghallgattak, és egy kis vállalkozó kedvet kérek az olvasóktól kis lapunk életben tartására, példányszáma növelésére...”

Koronkai István
IVÁNCSA, Ady E. u. 19.

Eddig a levél, és hogy ne essünk abba a hibába hogy sokat „molyolunk” egy jó ötlet megvalósításán, íme álljon itt (lásd 15. oldal) az Enterpress Alapítvány alapító okirata. Önmagáért beszél.

Milyen céljaink vannak ezzel az „újszülöttem”?

Nem árulunk zsákbamacskát, és nem hitegetjük Olvasóinkat azzal, hogy duplaszámot adunk ki, vagy, hogy azonnali színes borítót kap az újság, (bár erről sem mondunk le) ennél sokkal egyszerűbb a cél: mindenáron fenntartani a lapot és mindenáron növelni az érdeklődők (előfizetők) számát, s így közvetve a lap előállítására, terjesztésére fordítható összeget is.

Nyílt titok — és ezt is elmondtuk már számtalanszor — hogy a lapot egy maroknyi, lelkes Enterpress-t szerető kollégánk szerkeszti, akik az ismert általános gazdasági helyzet miatt (plusz mert mind-egyikük fiatal) nem igazán dúskálnak abban amiről az hír járja, hogy „nélküle az élet szép de vele még szebb. (Talán kitalálták, hogy az anyagi megbecsülésről beszél-ek.) Úgy gondolom, hogy minden kijuthatott az eddigi Enterpress fennállás óta nekik, csak ez nem. (Én csak tudom; én fizettem nekik, amikor éppen tudtam.)

De van más is!

Az még érthető volna, hogy több magyarországi ENTERPRISE forgalmazó széles nagy ívben ..., hiszen ők már lekaszálták ezt az üzletet (A Centrum Áruházakra és a NOVOTRADE-re

gondolok!) és úgy szépen magára is hagyták a felhasználókat, ahogy kell, de az hogy egy gyártó — akinek talán csak van valamilyen érdeke abban, hogy az eladásait növelje, felhasználóit kiszolgálja Kelet-Európában — ugyanezt tegye, ez nem fér bele abba a nagy „európai menetelés” című dologba.

Az általunk ismert „Képviselő”-vel többszáz forintos telefonszámla ellenére még csak beszélő viszonyt sem sikerült összehoznunk. (Na, a telefonszámla nem *Kopácsy Úr* pénztárcáját terheli.) Csak azt tudnám, hogy hogyan bírja elviselni zaklatásainkat. Nem különb egyébként a helyzet a fent említett úr közvetlen „kapcsolatánál” sem.

Ilyen körülmények között úgy tűnik, hogy egy kicsit egyedül maradtunk (úgy 20000-en). De még ez sem vette el eddig a kedvünket az ENTERPRESS kiadványtól. Van még egy-két dobásunk.

Tudjuk, minden alapítvány csak annyit ér, amennyit a vele kapcsolatban lévő áldoznak rá. Ezért a további füstölgéseket mellőzve jó szórakozást kívánunk a laphoz és kérjük, hogy akinek lehetősége, tehetsége, anyagi eszközei engedik segítsen!

A Kiadó nevében:

Juhász István
Juhász István
ügyvezető

TARTALOM 92/2

KURZUS

Assembly 10.	2-3
A Pascal 10.	4-5
A közvetlen lemezkezelés rejtelméi 2.	6

PROGRAMOZÁSTECHNIKA

Assembly segédrutinok	7-8
-----------------------	-----

HARDVER

Jó az egér a háznál, ha jó...	9
-------------------------------	---

TESZT

PCBRISE 1.2	10
-------------	----

TIPPEK-TRÜKKÖK

Ismét a CLICK-ről	11
-------------------	----

KÖNYVED MŰFAJ

Little Puff in Dragonland	12
The curse of Sherwood, ...	13-14

MINDENFÉLE

Alapító okirat, Postafiók 334	15
Hirdetések, felhívások	16

Assembly

10. rész

Ebben a részben nem mutatok be újabb programot, viszont megpróbálom tisztázni a különféle grafikus képernyőket felépítő bájtok szervezését, használatukat.

De van még egy múltkori tartozásom...

A scroll program működése

A program 13-15-ös sora amiatt kell, mert NAP (New Application Program) fájl készítettünk. Az EXOS ezt a programtípust csak akkor tudja elindítani, ha a megszakítástilítás és engedélyezés között a P0-n kijelöljük a stack-et. Ez egy hevenyészett eljárás, a specifikáció egyebet is előír. (Kitérő: Miért is kell valójában 5 fejlécű programmá fordítani? Nos, az Asmon olyan mohó, hogy induláskor az összes szabad szegmensre ráteszi a kezét. Így hiába próbálunk videolapot nyitni, memóriahiány miatt nem lehetséges.)

A 17-es sorban a 22. Exos változóba írt értékkel határozzuk meg a grafikus lapot. A 26-29-nél álló rész küldi el a 117-122 tartomány szekvenciáját. (A videoszekvenciákról a könyv 148-159-es oldalain olvashatunk.) Ezek rajzolják meg, és festik ki a kört. A program első 63 sorához már nem kell további magyarázat.

A 64-es sorban egy viszonylag összetett kifejezés áll, amely a videolap bájton mért nagyságát adja meg. Ennek felhasználásával akarjuk a lap Z80-as végcímét kiszámolni (Kezdcím + lapméret - 1 = végcím. Ki gondolta volna?)

Az x_size változót azért kell kettővel szorozni, mert a magasfelbontású grafikus lapok szélessége kétszer annyi bájttal, mint amennyit a 24. változóba frunk. Gondoljunk csak arra, amikor szoftver-szöveges (Text 80) típusú lapot definiálunk: Basicben a 80 karakteres szélesség eléréséhez SET VIDEO X 40-et frunk, a valóságban pedig a nyolcvan karakter miatt nyolcvan bájttal széles lesz az.

Az Exos csak úgy tudja használni az LPT-t (a demo gyártók öröme), hogy abban egy sor 9 bit magas. Mivel grafikus üzemmódról van szó - azaz a 9 bit magasságot effektíven kihasználjuk - 9-cel kell szorozni a video y-t. A -1 azért kell, mert a lapról egy bájtot már tudunk (a kezdcímét), ennek "nagyságát" (1 bájttal) már nem kell hozzáadnunk.

Összeadás után HL-ben előáll a lap végcím (65).

A lap mérete 20412 bájttal, azaz legrosszabb esetben 3 szegmensre helyezkedik el. A P1-P3-as tartományba helyezzük el a szegmenseket. Először (67-78) elmentjük, majd beállítjuk a lapkiosztást. (Végülis különösebb értelme nincsen, hiszen új alkalmazói programról van szó, és úgyis úgy kell "kirobbantani" magunkat a végén. De azért szokjuk meg a rendet!)

B-ben adjuk meg azt (80), hogy 200 pixellel akarjuk a kört balra mozgatni. De hogyan is kell ezt megoldanunk?

A sokféle lehetséges megoldás közül ez a program a lap végénél kezd a ténykedését. Sorban -jobbról balra- végigmegy egy pixelsoron. A pixelsorban éppen megcímzett bájttal bitjeit balra mozgatja, az üressé váló 0. bit-be (a legszélső a bájttal jobboldalán) pedig beteszi az előző bájttal kilépettet 7. bitjét. Ha egy pixelsor letelejéhez érkezett, akkor a legutolsó bájttal kilépő 7. bitjét a pixelsor végén álló bájttal 0. bitjébe írja.

A legkülső ciklus (cycl_0) a kétszáz lefutást frányítja. Mivel B a ciklusszámláló, HL-be pedig a nagyon fontos lapvégcím van, ezért ezeket az értékeket elmentjük (81-82), így nyugodtan "elhasználhatjuk" őket.

A középső ciklus (cycl_1) számlálója szintén a B regiszter lesz. Ez a ciklus megy végig az összes pixelsoron, ezért B kezdeti értéke az összes sorok száma (84), a középső ciklus ezután kezdődik.

Töröljük a C bitet (85), ne kerüljön szemézt az adott pixelsor végén álló bájttal 0. bitjébe. (Nem történne nagy katasztrófa nélküle, hiszen a bitet a végén úgyis felülírjuk.)

C lesz a belső ciklus (cycl_2) számlálója, kezdeti értéke az adott pixelsor hossza bájtokban mérve (86). A ciklus elején (87) a HL által címzett bájttal bitjeit balra mozgatjuk. A mozgatás után C bitben a kilépő 7. bit lesz, a bájttal 0. bitjébe pedig C bit előző értéke lesz. Csökken HL értéke (88), az az előbbi bájttal előtti bájtra mutat. Csökken a ciklusszámláló is (89). Ha még nem ért a pixelsor elejéig, akkor folytatódik a belső ciklus (90).

Ha a pixelsor elejére értünk, megnézzük, hogy a kilépett bit értéke egyenlő-e 1-gyel? Ha igen, be kell még állítanunk a pixelsor végén álló bájttal 0. bitjét 1-be (91).

Mindent egy külön kis rutin végzi, de nézzük csak, hogy ekkor hova mutat HL? Nem nehéz kitalálni, hogy az már az előző pixelsor feletti másik sor legvégére mutat - így lép ki a belső ciklusból -, nekünk viszont az ez alattira lenne szükségünk. Mi sem egyszerűbb! DE-be adjuk a sor hosszát bájtokban (95), ezt HL-hez adjuk (96), és rendben vagyunk. Beállítjuk a 0. bitet 1-be (97), a következő kivonáshoz C bitet nullázunk (98), és HL értékének visszanyeréséhez kivonjuk belőle az előbb hozzáadott értéket (99), és visszalépünk a középső ciklusba (100).

Ezután a program megnézi, hogy végigmegintünk-e az összes pixel-

soron (92), azaz a teljes képernyőn. Ha nem, újra lefut a pixelsorokat kezelő rész.

Ha a képernyő összes bitjét megmozgattuk, akkor jöhet a következő mozgatás, ismét le kell futtatni a scroll rutint. Visszaállítjuk HL (102) és a ciklusszámláló B értékét (103), kezdődhet az egész előről.

A "grafikus" bájtokról

A kétszínű üzemmód

Az előző példában magasfelbontású, kétszínű grafikat használtunk. Ennek a típusnak a kezelése a legegyszerűbb, hiszen a képernyőt felépítő összes bájttal bitjei egyértelműen meghatározzák egy-egy képernyőpontot. (Két "színt", két értéket különböztetünk meg. Ehhez egyetlen bit elegendő, így tehát egy bájttal 8 pixelt tartalmaz.) Az első pixel a b7, a második a b6 stb.

A színek miatt más a helyzet a többi képernyőtípusnál. Előjáróban szeretném felhívni a figyelmet arra, hogy a színkezelésben nincs különbség a magas és alacsonyfelbontású lapok között!

A négyzínű üzemmód

Négyféle értéket két biten tudunk ábrázolni. Ez a négyzínű üzemmódnál annyit jelent, hogy egy képernyőbájton 4 pontot tudunk ábrázolni (4 * 2 bit = 8 bit).

Az adott pixel színét a hozzátartozó két bit határozza meg, nézzük ennek a szervezését (1. ábra) egy teljes bájttal:

1. ábra

1	2	3	4
b7	b6	b5	b4
b3	b2	b1	b0

A grafikon ténylegesen megjelenő pontokat az 1,2,3,4 számok jelölik. Az egyes pixeleket színeikhez vezérléséhez a következő bitek vannak rendelkezésre (2. ábra):

2. ábra

pixel	bit	bit
1.	b7	b3
2.	b6	b2
3.	b5	b1
4.	b4	b0

Mit mutat a két táblázat? Az elsőn azt a négy pontot láthatjuk, amelyek grafikusan fognak megjelenni. Eszerint a legelső pixel a 7. és a 6. bitek jelenlétét jelenti. Azt viszont, hogy az itt megjelenő pixelnek milyen színe lesz, a 7. és 3. bitek kombinációja dönti el. Elképzelhetjük úgy a dolgot, mintha a pixel mindig "bekapcsolt" lenne, és a színe b7-től és b3-tól függene.

Már csak az a kérdés, hogy miképpen határozza meg a két bit kombinációja a tintaszínt (3. ábra):

3. ábra

INK	bit	bit
0.	0	0
1.	1	0
2.	0	1
3.	1	1

Ez természetesen az összes "színmeghatározó" bitre együttesen érvényes.

Biztos ami biztos, nézzünk egy példát! Azt akarjuk elérni, hogy a harmadik pixel (b3, b2 helye) vegye fel az INK 2 színt, a negyedik pedig (b1, b0 helye) az INK 1-et.

A harmadik pixel színét b5 és b1 írja le, az INK 2-höz b5=0, b1=1

A Pascal

10. rész

A Turbo-Pascal eljárásai és függvényei

Az előző számban nagyon röviden áttekintettük a szabványos Pascal nyelv eljárásait és függvényeit, itt-ott utalva az egyik vagy másik megvalósítás sajátosságaira. A közismert és méltán népszerű Turbo-Pascal sok-sok új lehetőséget ad a programozónak, nagyrészt a beépített eljárások és a függvények listájának kibővítésével. Most ezeket a lehetőségeket tekintjük át röviden. Azok se keseredjenek el, akiknek - lemezegységgel nem rendelkező - be kell érniük a HiSoft Pascal szolgáltatásaival; a bemutatott megoldások egy része, kisebb-nagyobb módosításokkal, a HiSoft Pascalban is megvalósítható egy kis kiegészítés útján. Sorozatunkban igyekszünk majd közölni egy ilyen nyelvi bővítést.

A stringek és kezeléseik

A stringek (magyarul fűzők) - a szót sokan stringnek, sokan viszont sztringnek ejtik - igen kényelmes és általánosan elterjedt módját adják a karakteres információ kezelésének. A szabványos Pascal, sajnos, nem tartalmazza a stringet mint nyelvi elemet, helyette az ARRAY OF CHAR típust vezeti be; ez egy tetszőleges, de előre meghatározott hosszúságúra deklarált karaktertömb. Ennek kezelése nem olyan rugalmas, mint a többi nyelv, pl. a BASIC string-kezelése. A Pascal ugyanis csak azonos hosszúságú karaktertömbök között enged meg műveleteket, pl. értékadást. Az is korlátot jelent a karaktertömbök használatánál, hogy a függvények visszatérő értéke csak skalár lehet, így karaktertömb eredményt nem lehet függvénytől visszaadni.

Ennek a nyilvánvaló hiányosságnak a kiköszöbölésére vezették be a Turbo-Pascalban a STRING típust. A korábbi Turbo-Pascal verziókban - így az ENTERPRISE gépeken is használható 3.01-es verzióban - a stringek hosszát (ez persze mindig a maximális hossz) deklarálni kell:

```
VAR
  S : STRING[ 80 ];
```

Az így deklarált S változó tetszőleges karaktersorozatot kaphat értéket, a sorozat hossza 0 és 80 között lehet. Ugyan most is csak azonos hosszúságú stringek között lehet műveleteket végezni, de ez az ellenőrzés szükség esetén kikapcsolható a (\$V-) fordításvezérlő opcióval (ezekről az opciókról majd később lesz részletesen szó).

A Turbo-Pascal későbbi verziói megengedik mind a fenti, deklarált hosszúságú, mind pedig a deklarátlanul hosszúságú stringek használatát:

```
TYPE
  STRING80 = STRING[ 80 ];
  STRING2  = STRING[ 2 ];
VAR
  S1 : STRING80;
```

```
S2 : STRING2;
S3 : STRING;
```

Az S3 string persze nem lehet akármilyen hosszú, a gépi megvalósítás határt szab a hosszúságnak, mégpedig 255 karakterben. Ez csak látszólag jelent korlátot: valójában ki akar több mint három képernyősornyi stringekkel dolgozni? A 255 karakteres hosszkorlátnak az oka egyébként a gépi megvalósításban rejlik. A Turbo-Pascal - sok más nyelvhez hasonlóan - egy hosszúddal megjelölve tárolja a stringet alkotó karaktereket. Mivel a hosszúddal egy bájtost, a string hossza 0-tól 255 bájtig terjedhet.

Lássuk a stringeken végezhető műveleteket. Nyilvánvaló művelet az értékadás. Esetleg kevésbé nyilvánvaló, de igen hasznos a relációs műveletek lehetősége. Az összehasonlítások az a string a kisebb, amelyikben - természetesen az adott kódtáblázatban - előbb adódik alacsonyabb kódú karakter, vagy amelyik rövidebb. A Turbo-Pascal beépített relációs operátorai így kiválóan alkalmasak angol szavak ábécébe rendezésére, de csak valamilyen trükköt alkalmazva lehet ékezetes, tehát mondjuk, magyar szavakat sorbaszedni velük. Megjegyezzük még, hogy a karakter (CHAR) típus kompatibilis az egységnyi hosszúságú stringgel; a stringet alkotó karakterek pedig egyszerű indexeléssel emelhetők ki, mintha a string karaktertömb volna.

A stringekkel való egyéb műveletekhez a Turbo-Pascal egy sor eljárást és függvényt ad. A tárgyalásnál s stringet, nc stringkonstanst, sv stringváltozót jelent, az esetleg használt indexszel; n numerikus értéket, mégpedig nc konstanst, nv változót, ne kifejezést jelent. A jobb érthetőség kedvéért bevezetünk még két jelölést: l hosszúságot jelöl, k pedig egy - később részletezendő - hibakódot; mindkettő egész érték. Az sv stringváltozó helyén - az előbb említett kompatibilitás miatt - karakterkonstanst is állhat.

A LENGTH függvény a paraméterül adott string hosszát adja meg; alakja:

```
LENGTH( s )
```

A függvény értéke a string tényleges hossza:

```
WRITELN( LENGTH( 'négy' ) );
```

eredménye 4, de

```
WRITELN( LENGTH( 'öt' ) );
```

eredménye nyilván 2.

A CONCAT függvény stringek összefűzésére szolgál, alakja

```
CONCAT( s1, s2 ... sn );
```

A függvény az s1, s2 ... sn stringet a felírás sorrendjében egymáshoz fűzi, és a kapott stringet adja vissza függvényértékként. Az eredmény nem lehet hosszabb 255 karakternél, különben hibajelzést kapunk. Ha az eredmény a string deklarált hosszánál nagyobb hosszúságú, a "felesleg" elvész. Megjegyezzük, hogy az függvény helyett használható a + operátor is, az alábbi alakban:

```
sv := s1 + s2 + ... sn
```

Az INSERT eljárás egy string beszúrását végzi egy másik stringbe; alakja:

```
INSERT( s, sv, n )
```

Az eljárás az s stringet beszúrja az sv stringbe, annak n-ik karakterpozíciójától kezdődően. Ha az eredményül adódó string túl hosszú, a "kilógó" rész elvész.

```
S := 'abcdefgh';
```

```
INSERT( '-', S, 5 ); WRITELN( S );
```

eredménye

```
abcd—efgh
```

A DELETE eljárás stringek egy darabjának a törlésére használható; alakja:

```
DELETE( sv, n, l )
```

Az eljárás az sv stringváltozóból törli az n-ik pozíciótól kezdődő, l karakter hosszúságú darabot (rész-stringet). Ha a string nemlétező darabját akarjuk törölni, nem történik semmi; ha a törlendő darab "kilóg" a stringből, értelemszerűen csak a string létező, törölhető "vége" törlik.

```
S := 'abcdefgh';
```

```
DELETE( S, 5, 8 );
```

eredménye

```
'abcd'
```

A COPY függvény stringek egy darabjának kiemelésére szolgál, alakja:

```
COPY( s, n, l )
```

A függvény az s stringből, annak n-ik pozíciójától kezdve kiemel egy l karakter hosszúságú rész-stringet, és ez lesz a függvény visszatérési értéke. Ennek megfelelően

```
WRITELN( COPY( 'abcdefgh', 7, 2 ),
```

```
COPY( 'ABCDEFGH', 7, 2 );
```

eredménye

```
ghGH
```

A POS függvény egy string előfordulását keresi meg egy másik stringben. Alakja:

```
POS( smit, smiben )
```

A függvény az smit string előfordulását keresi az smiben stringben. Ha ez utóbbi tartalmazza az előbbit, a függvény visszatérési értéke az előfordulás kezdő pozíciója lesz; ellenkező esetben a függvény értéke nulla. A

```
TYPE
```

```
STRING80 = STRING[ 80 ];
```

```
FUNCTION FNAME( S : STRING80 ) : STRING80;
```

```
VAR
```

```
PPOS : INTEGER;
```

```
BEGIN
```

```
PPOS := POS( ' ', S );
```

```
IF PPOS = 0 THEN BEGIN
```

```
  FNAME := COPY( S, 1, PPOS - 1 );
```

```
END ELSE BEGIN
```

```
  FNAME := S;
```

```
END { IF };
```

```
END { FNAME };
```

programrészlettel definiált függvény egy fájl-név részét emeli ki (itt most eltekintünk a vezető és záró szóközök levágásától és egyéb lehetséges problémától, amit egy "éles" függvénynek tudnia kellhet).

A fennmaradó két függvény a stringek és numerikus értékek közötti konverziót valósítja meg mindkét irányban.

A VAL eljárás egy string-alakban megadott számot konvertál (egész vagy valós) numerikus értéké:

VAL(s, nv, k)

Az eljárás az *s* stringben lévő számjegy- és egyéb karaktereket megpróbálja egész vagy valós számmá átalakítani, és ezt az értéket teszi az *nv* változóba. A változó értéke nem használható minden esetben: ha ugyanis az átalakítás sikeres volt, a *k* hibakód értéke 0, ellenkező esetben annak a karakternek a sorszámát adja meg a stringben, ahol a konverziót nem lehetett tovább folytatni.

VAL('1234', N, K); WRITELN(N : 8, K : 4);

eredménye

1234 0

míg

VAL('12A45678', N, K);

esetén K értéke 3 lesz, mutatva, hogy hol "bicsaklott ki" a konverzió, N értéke pedig meghatározatlan.

Egyszerűbb az eset a fordított konverzió esetében (egy szám mindig kírható számjegyeivel...). Az STR függvény, amelynek alakja

STR(no, sv)

az *no* numerikus kifejezés aktuális értékét teszi az *sv* stringváltozóba. Ennek értelmében

STR(2 * 2, S); WRITELN(S);

eredménye 4 lesz.

A VAL eljárást akkor érdemes használni, ha teljes egészében magunk akarjuk kezelni a numerikus bevitelt, azaz "hülyeség ellen védett" programot szeretnénk írni. A Turbo-Pascal ugyanis hibajelzéssel kilép, ha a READ vagy a READLN eljárás konvertálhatatlan adatot kap. Ha viszont csak szövegeket adunk meg, mindaddig vár a program, amíg valami "chetőt" nem kap.

Az STR eljárást például akkor használjuk, amikor grafikus képernyőre kell számértékeket írni (a Turbo-Pascal grafikus lehetőségeivel csak szöveget tudunk közvetlenül kírni).

A dinamikus változókkal kapcsolatos eljárások és függvények

Az adattípusok tárgyalásánál megismerkedtünk a *dinamikus változó* fogalmával. A dinamikus változót az különbözteti meg a statikus változótól, hogy míg ez utóbbi a program elindulásakor, illetve - ha egy eljárás (függvény) lokális változójáról van szó - az eljárás (függvény) meghívásakor létrejön, és egészen a program befejezéséig vagy az eljárás (függvény) végrehajtásának befejezéséig létezik, addig az előbbi csak akkor jön létre, ha ezt a program explicit módon előírja. A statikus változónak neve van, a dinamikus változót viszont csak egy mutatón - pointeren - keresztül lehet elérni. Az alábbi deklaráció

TYPE

ALAPTIPUS = { itt tetszőleges
(statikus) típusdeklaráció
állhat };

MUTATOTIPUS = ^ALAPTIPUS;

VAR

MUTATO : MUTATOTIPUS;

létrehoz egy tetszőleges alaptípust, a MUTATOTIPUS típus pedig erre az alaptípusra mutató mutatót (sic!). A MUTATO változó mutat majd az ALAPTIPUS típusú dinamikus változóra, ha azt - később - létrehozuk.

Ennek az első olvasásra talán körmönfontnak tűnő megoldásnak a gyakorlati programozói munkában igen sok előnye van. A legfontosabb, hogy csak ilyen módon tudunk olyan táblázatokkal, listákkal dolgozni, amelyeknek a méretét nem tudhatjuk előre.

A BASIC például csak a DIM kulcsszóban megtestesülő *vektort* vagy *matrixot* adja lehetőségeként. Ezeknek a méretét előre definiálni kell. Nincs annál dühtőbb, amikor a program leáll, mert az egyik táblázatunk betelt, pedig a másik még szinte üres, de hiába, mert a rendszer nem tud "ellopni" az egyik táblázat számára lefoglalt memóriából, hogy odaadja azt a másikkal.

Ha viszont dinamikus adatszerkezetekkel dolgozunk, a különböző táblázatokhoz csak egy-egy mutatót kell előre lefoglalnunk, a menet közben létrehozott dinamikus változók ugyanazt a tárterületet kezdik betölteni. A program csak akkor áll le, ha valóban minden rendelkezésre álló memória elfogyott.

A szabványos Pascalban a NEW eljárás hozza létre a dinamikus változót. Az előző példát folytatva, a

NEW(MUTATO)

eljárás létrehozza a tár erre kijelölt szabad területén az ALAPTIPUS típusú változót, a változó *címét* pedig elhelyezi a MUTATO nevű - statikus - változóban. Magát a dinamikus - tehát az ALAPTIPUS típusú - változót csak a mutatóján, a MUTATO változón keresztül érhetjük el. Ha például az ALAPTIPUS egyszerűen INTEGER, akkor a dinamikus változónak a

MUTATO^ := 0;

utasítással adhatunk értéket; itt a ^ karakter jelzi, hogy nem a MUTATO kap értéket, hanem az általa *mutatott* változó kap értéket. Ha az ALAPTIPUS viszont string volt, akkor az értékadás így nézhet

ki:

MUTATO^ := 'Ez lesz a dinamikus változó tartalma';

Ha újra meghívjuk a NEW eljárást, újabb, azonos típusú dinamikus változó jön létre, és most ennek a címe kerül a MUTATO mutatóba.

Ha a mutatót változó, amely természetesen egy rekord is lehet, tartalmaz egy vagy több mutatót is, határtalan lehetőségek nyílnak meg előttünk. A dinamikus változó "testében" lévő mutató ugyanis egy másik, ugyanilyen dinamikus változóra mutathat, az egy következőre, és így a (majdnem) végtelenségig. A dinamikus változók ekkor az egymásra mutogató mutatók segítségével tulajdonképpen "fel vannak fűzve", a program a változókon "végiglépkedhet", ha ismeri az első elem címét.

A dinamikus változó persze nem csak születhet, hanem meg is halhat a program futtatása során. Ha valahány dinamikus változó feleslegessé vált, és szükségünk van az általa lefoglalt memóriaterületre, a RELEASE eljárással felszabadíthatjuk a memória egy részét. A

RELEASE(MUTATO)

eljárás törli a MUTATO által mutatott dinamikus változót, valamint az összes olyan dinamikus változót, amely az után a változó után hoztuk létre.

Ha a program futása közben valamikor végrehajtjuk a

MARK(MUTATO)

eljárást, a Pascal-rendszer a MUTATO változóban tárolja a dinamikus változók által még éppen *nem* lefoglalt első szabad memóriacímét, így mintegy *megjelöljük*, hogy a későbbi RELEASE eljárás honnantól szabadítsa fel a memóriát.

Törölhetünk egy dinamikus változót a DISPOSE(MUTATO) eljárással is, ekkor azonban csak egyetlen változó helye szabadul fel, azaz a memóriában egy "lyuk" keletkezik. Később, egy újabb NEW eljárással végrehajtásakor az így keletkezett lyuk esetleg "betömődik"; ebből ki következtethető, hogy a MARK-RELEASE páros és a DISPOSE eljárás között választanunk kell, egy programon belül a kettőt egyenesen használva sok kellemetlenséget okozhatunk magunknak.

Az eddig leírtak a szabványos Pascalban érvényesek. A Turbo-Pascalban a dinamikus memóriakezelés további lehetőségekkel bővül. A tárgyalásnál *p* mutatót (pointer) jelöl, *s* pedig memóriaméretet bájttal, ez utóbbi egész kifejezés lehet.

A GETMEM eljárás segítségével a dinamikus változók számára fenntartott memóriaterületen, a *heap*-ben (magyar neve *halom*) tetszőleges méretű memóriaterületet foglalhatunk le, és használhatjuk kedvünk szerint. A

GETMEM(p, s)

eljáráshívás *s* bájttal memóriát foglal le, a kezdőcímet pedig a *p* mutató típusú változóba teszi. Ehhez *p*-t előzőleg az általánosított POINTER típusúnak kell deklarálni:

VAR

P : POINTER;

A GETMEM eljáráshívással lefoglalt memóriaterület felszabadítását végzi a FREEMEM eljárás. A

FREEMEM(p, s)

eljáráshívás felszabadítja a *p* mutató által mutatott, *s* bájttal méretű memóriaszakaszt. A létrehozásakor és a felszabadításakor méretnek értelemszerűen meg kell egyezni egymással.

Ha nem akarjuk, hogy a program várakozás nélkül hibajelzéssel leálljon, ha a *heap* megtelik, érdemes ellenőrizni, van-e elegendő memória a létező dinamikusan változó vagy a lefoglalandó terület számára.

A MEMAVAIL - paraméter nélküli - függvény a *heap* számára még rendelkezésre álló szabad memóriaterület méretét adja vissza, függetlenül attól, hogy az egy darabban vagy szabadalva érhető-e el. Akkor használjuk, ha a MARK-RELEASE párossal szabadítjuk fel a memóriát.

A MAXAVAIL függvény a legnagyobb *egybefüggő* memóriaterület méretét adja vissza. Akkor használjuk, ha a DISPOSE eljárással szabadítjuk fel a memóriát, vagy ha a GETMEM-FREEMEM párossal dolgozunk.

A SIZEOF függvény árulja el egy tetszőleges típusú vagy változó memóriabeli méretét. Ha tehát nem ismerjük dinamikus változónk nagyságát, vagy bonyolult lenne azt kiszámolni, ne esszünk kétségbe.

WRITELN(SIZEOF(INTEGER) : 6);

eredménye 2; ugyanezt az eredményt megkaphatjuk kissé bonyolultabban is:

VAR

N : INTEGER;

BEGIN

WRITELN(SIZEOF(N) : 6); END.

Továbbá, SIZEOF(REAL) eredménye 4, SIZEOF(CHAR) értéke 1, SIZEOF(STRING[80]) eredménye 81 (miért?) SIZEOF(MUTATO) értéke a 8-bites gépeken 2.

(folytatjuk...)

-UL-

A közvetlen lemezkezelés rejtelmerei

2. rész

Az előző részben ismertettük a WD1770-es lemezvezérlő IC felépítését, programozását és az állapotbitek jelenlétét. A parancsok részletes ismertetéséből még kimaradt néhány, most ez következik.

III. Trackműveletek

Ide két parancs tartozik, a track olvasás és a track frás ami nem más, mint a formázás. A szektorműveletekhez hasonlóan itt is be kell pozicionálni a fejet a megfelelő sávra mielőtt kiadjuk az utasítást.

Sáv olvasása szinte teljesen megegyezik a szektor-olvasással, de itt nem kell szektorszámot megadni. A betöltés az indexyuk megtalálásával kezdődik, és egészen addig tart amíg másodsorra elérkezik az indexyukhoz, vagyis amíg a lemez egyszer körbefordul. Ilyenkor minden betöltődik ami a lemezen van, tehát a szektorok adatain kívül a szektorközi területeket (lásd formázásnál) is. A különböző szinkronváltások miatt a Gap területen lévő bájtok hibásak is lehetnek, sőt akár adatvesztés is bekövetkezhet, de az index címjelzés, az index mező, az adat címjelzés, az adatmező és a CRC bájtok helyesek lesznek.

A sáv frása nem más mint a formattálás, vagyis fel kell írni a track-re a szektorazonosítókat, az adatmező helyét és a szektorközi területeket.

Egy sáv felépítése az 1. ábrán látható, az egyes szektorok felépítése pedig a 2. ábrán. Mint az ábrából is kiderül, egy szektor nem csak magából az adatterületből áll, hanem a szektorazonosítóból, különböző jelzőbájtokból és kitöltő területekből (Gap-ekből) is.

A sáv felépítése
Gap I
Szektorok területe
Kitöltés

1. ábra

A szektorok felépítése
Gap II
Szektorfej
Gap III
Adatmező
Gap IV

2. ábra

A Gap-eknek fontos feladatuk van, nem csak a helyet foglalják, hanem időzítnek is. Egy szektor adatainak beolvasása után ugyanis szüksége van a WD-nek egy bizonyos feléledési időre, mire a következő olvasást elkezdheti. Formázáskor a 2. ábrán látható adatokat annyiszor kell felírni a lemezre ahány szektorra akarjuk formázni a lemezt. Egy sáv mérete névlegesen 6250 bájt dupla sűrűségű felírás esetén és 3125 bájt szimpla sűrűségű, formázáskor azonban érdemes még rátenni egy kicsit, vagyis megnövelni a legutolsó kitöltő területet a sebességváltozások miatt, tehát dupla sűrűségűnél 6500 bájt, szimplánál 3250 bájtot kell kiküldeni a WD-nek.

A Gap-ek ajánlott felépítése a 3. ábrán látható, a Gap II kivételével mindegyiktől el lehet térni, de csak az egyes bájtok számának növelésével.

A Gap-ek felépítése	
-FM	-MFM
Gap I 16 bájt 0FFh	32 bájt 4Eh
Gap II 11 bájt 0FFh	22 bájt 04Eh
6 bájt 00	12 bájt 00
	- - 3 bájt 0F5h
Gap III 10 bájt 0FFh	24 bájt 4Eh
4 bájt 00	8 bájt 00
	- - 3 bájt 0F5h

3. ábra

Az adatok felírását a sávra a szektorfráshoz hasonlóan kell elvégezni. Fizikailag az frás az indexyuk elérésénél kezdődik, és egy teljes lemezfordulatig tart. A szektorfrástól eltérően itt nem minden bájt íródik fel egy az egyben a lemezre, hanem vannak speciális bájtok amelyekkel a WD eltérően kezel. FM felírásnál (szimpla sűrűség) a 0F8h és 0FEh közötti bájtok a CRC ellenőrzés inicializálására valók, ugyanazt MFM felírásnál (dupla sűrűség) a 0F5h bájt kiküldésével érhetjük

el. Mindkét felírási módnál 0F7h hatására két CRC bájt íródik a lemezre, 0FEh a szektorfej jelzése és 0FBh az adatmező jelzése. A szektorfej és az adatmező után is fel kell írni a lemezre a két CRC bájtot, tehát egy 0F7h bájtot kell elküldeni a WD-nek. A fentiekből következik, hogy a Gap-ekben, a szektorfejben és az adatmezőben nem használhatók a 0F5h és 0FEh közti értékek.

IV. Megszakítás

Ezzel a parancsal különböző megszakításokat lehet kérni a WD-től, az 10-13 bit beállításától függően, azonban ENTERPRISE-on ezt nem tudjuk kihasználni, mert a megszakítási láncban nem szerepel. Ezért csak egy beállítás mellett van értelme használatának, az 10-13 biteket nullára állítva (0D0h) a folyamatban lévő parancsot tudjuk megszakítani.

Nem esett szó még egy portról ami a kontrollerkártyához tartozik, de nem a WD közvetlen portja. Ez a 18h port amelyen a használni kívánt meghajtót tudjuk kiválasztani. Ezt az alsó négy bit valamelyikének beállításával tehetjük meg. Értelemszerűen a 0-ás bit az A meghajtót, az 1-es bit a B meghajtót stb. jelenti. A portra írt érték 4-es bitjével pedig a meghajtó 0-ás és 1-es oldalát tudunk szelektálni. Ha például 12h értéket küldünk a 18h portra, akkor a következő parancs a B jelű lemezegység 1-es oldalára fog vonatkozni.

További információk

A szektorfej felépítése
0FEh (Szektorfej jelzése)
Sáv száma (0-0F4h)
Lemezoldal száma (0 vagy 1)
Szektor száma (1-0F4h)
Szektor mérete (0,1,2,3)
0F7h (CRC felírása)

Az adatterület felépítése
0FBh (Adatterület jelzése)
Szektorméretnek megfelelő számú adatbájt
0F7h (CRC felírása)

Példák

A táblázatok első oszlopa a bájtok számát, a második oszlop az írandó értékeket jelentik:

1., FM-es formázás	2., MFM-es formázás
40 0FFh	60 4Eh
17-szer:	12 00
6 00	3 0F6h
1 0FEh	1 0FCh
1 Trackszám	60 4Eh
1 Oldalszám	9-szer:
1 Szektorszám	12 00
1 00 (128 bájt/szektor)	3 0F5h
1 0F7h	1 0FEh
11 0FFh	1 Trackszám
6 00	1 Oldalszám
1 0FBh	1 Szektorszám
128 0E5h	1 2 (512 bájt/szektor)
1 0F7h	1 0F7h
10 0FFh	24 4Eh
337 0FFh	12 00
	3 0F5h
	1 0FBh
	512 0E5h
	1 0F7h
	84 4Eh
	460 4Eh

Egy tanács

Ha a közvetlen lemezkezeléssel kísérletezünk, akkor soha ne olyan lemezen próbáljuk ki, amelyen fontos adataink vagy programjaink vannak, mert könnyen tönkremehetnek egy esetleges programhiba miatt.

-DEVIL-

Assembly segédrutinok

A különböző nyelvek közül az assembly programozás igényli a legtöbb időt és energiát. Egyszerű feladatoknál különösen kitűnik, hogy a perifériális INPUT/OUTPUT, főképpen a hibakezelés mennyi munkába kerül.

Vannak gyakran ismétlődő feladatok. Az ilyen rutinokat célszerű uniformizálni, és feladatuk szerint csoportosítani. A fáradtságunk gyorsan megtérül, mert csak egyszer kell őket megírni.

Az első rutin fájl a numerikus és sztringadatok kiírását hivatott könnyíteni. Az írás történhet a memóriába, ahol a kezdőcímet DE regiszterpárban tároljuk, és a visszatérés után a következő szabad bájtra fog mutatni.

Másik lehetőségként a 4. EXOS változóval beállított alapértelmezésű csatornára írhatunk. Visszatéréskor AF tartalmazza az EXOS hívás státuszát. A rutinok megőrzik a HL, az index, és az alter regiszterek tartalmát.

Sztringek írása: A CALL hívást követő DEFB adatokat küldi el blokkírással az első nulla bájtig, majd visszatér az ezt követő címre. A rutinnal nulla bájtot nem lehet kiírni, viszont a hossz nincs korlátozva.

Numerikus adatok írása: A számok mérete 8 vagy 16 bites lehet. Az egy bájtot az akkumulátorba, a két bájtot a HL regiszterpárba kell helyezni. A bináris számokat kiírhatjuk decimális vagy hexadecimális, a BCD számokat pedig a hexadecimális rutinnal. (A karakterek elküldése bájtonként történik.)

A hexa kiírás az egyszerűbb. Minden 4 bitet hexadecimális karakterre kell alakítani. A 8 bit így 2, a 16 bit 4 karakter hosszú lesz.

A decimális kiírásnál először BCD alakra kell konvertálni, ezután már a hexadecimális rutin is elvégezhetné a feladatot. A C regiszterbe helyezett értékkel a kiírt szám formáját is meghatározhatjuk:

C=0: A vezető nullák nem íródnak ki

C=1: A vezető nullák mint szóköz karakterek kerülnek kiírásra

C=2: A vezető nullák is kiíródnak

A C regiszter 7. bitjét beállítva figyelmen kívül marad a legmagasabb decimális jegy, így a 8 bites számot százasként, a 16 bitest tízezresek nélkül tudjuk kiírni.

A szövegfájlt célszerű az ASMON editorával elkészíteni. Szeretném felhívni a figyelmet itt néhány hibalehetőségre: Aki 80 karakteres (ASME) editorral dolgozik, az INCLUDE fájlokat ne mentse ki CTR-1-vel (inteligens gyorsmentés), mert fordításnál hibát okoz. Valamilyen hiba folytán a mentés gyakran csonka. Ezt elkerülhetjük ha az editorból F8-al kilépünk, majd E-vel visszalépünk. Ekkor F4-el választhatunk mentést (TURBO SAVE). Fájlnevének "NUM_STR.ASM"-et adjunk. Ha később valamilyen programunkban használni szeretnénk, INCLUDE NUM_STR.ASM direktívával egyszerűen hozzá-

fordítjuk a kódunk végéhez.

Példaképpen mellékelem a DUMP.ASM listát, mely több rutinszolgáltatást is igénybe vesz. A fordítás előtt [Z]-vel beállítjuk az assembler opciókat:

```
Assembly listing OFF
List conditions NO
Force Pass 2 NO
Memory assembly YES
Memory offset 0
Object file name:
```

[A]-val fordítuk. Hibátlan esetben [G]-vel megadjuk a 3000-s kezdőcímet, majd Enter-rel lefuttatjuk a kódot.

Végül a lényegesebb rutinnevek listája:

DECHLPRINT	16 bites decimális
DECAPRINT	8 bites decimális
DECHLDE	16 bites dec. memóriairás
DECADE	8 bites dec. memóriairás
HEXHLPRINT	16 bites hexadecimális
HEXAPRINT	8 bites hexadecimális
HEXHLDE	16 bites hex. memóriairás
HEXADE	8 bites hex. memóriairás
STRPRINT	sztring
STRDE	sztring memóriairás
APRINT	akkumulátor
BPRINT	B-regiszter
SPPRINT	szóköz kiírása
CRLFPRINT	újsorjel kiírása
JPHL	vezérlésátadás
JPIX	vezérlésátadás
JPIY	vezérlésátadás

(Hsoft)

```

SOROK:      ORG 3000H      ;A KOD KEZDOCIME
SORHOSSZ:   EQU 10
KEZDOCIM:   EQU 8
DUMP:       CALL STRPRINT ;SZOVEG KIIRATAS
            DEFB "MEMORIADUMP:",13,10,0
            LD HL,KEZDOCIM ;A DUMP KEZDOCIME
            LD B,SOROK      ;A DUMPSOROK SZAMA
DUMP10:     PUSH BC       ;SORSZAMLAO ELMENTES
            LD C,1         ;VEZETO SZOKOZOK
            CALL DECHLPRINT ;SORCIM DECIMALISAN
            CALL SPPRINT    ;SZOKOZ IRAS
            CALL HEXHLPRINT ;SORCIM KIIRATAS
            CALL SPPRINT    ;SZOKOZ IRAS
            PUSH HL        ;SORCIM ELMENTES
DUMP20:     LD B,SORHOSSZ  ;SORBAJTOK SZAMA
            PUSH BC       ;SZAMLALO MENTESE
            LD A,(HL)     ;BAJT BEOLVASAS
            INC HL        ;POINTER NOVELES
            CALL HEXAPRINT ;BAJT HEXA KIIRASA
            CALL SPPRINT  ;SZOKOZ KIIRAS
            POP BC        ;SZAMLALO VISSZATOLTES
            DJNZ DUMP20   ;CIKLUS SZAMLALASA
            POP HL        ;SORCIM VISSZATOLTES
DUMP30:     LD B,SORHOSSZ  ;SORBAJTOK SZAMA
            PUSH BC       ;SORSZAMLAO ELMENTES
            LD A,(HL)     ;BAJT BEOLVASAS
            INC HL        ;POINTER NOVELES
            SUB 32        ;A TARTALMAT 32-159...
            RES 7,A       ;...KOZOTTI ERTEKRE...
            ADD A,32      ;...KONVERTALJUK.
            CALL APRINT   ;A KARAKTER KIIRASA
            POP BC        ;SZAMLALO VISSZATOLTES
            DJNZ DUMP30   ;CIKLUS SZAMLALAS
            CALL CRLFPRINT ;UJSORJEL KIIRAS
            POP BC        ;SZAMLALO VISSZATOLTES
            DJNZ DUMP10   ;SORCIKLUS SZAMLALAS
            RET           ;VEGE...VISSZATERES
            INCLUDE NUM_STR.ASM
            ;RUTINFAJL CSATOLASA
            ;A PROGRAMHOZ.
```

Totózők figyelem!!!

A TOTOTIPP programmal ezres nagyságrendű kulcsbázis is kezelhető egy lemezen! 300-at meghaladó kulcsot tartalmaz, melyet tetszés szerint lehet bővíteni.

Segítséget nyújt a bontás készítésénél és a találatok értékelésénél is.

Ára 5.25-ös lemezen 300Ft+postaköltség.
A formal igényt (DS/DD, DS/SD, SS/DD, SS/SD) kérem megadni!

Megrendelhető: Haluska László,
1086 Budapest, Karácsony S.u.18.3.41.


```

NUM STR.ASM          1992. HSOFT:
; HL,IX,IY ES AZ ALTER REGISZTEREK MEGORZODNEK:
; CSATORNA HIVAS UTAN, A STATUSZ AZ AF-BEN LESZ.
; MEMORIA IRASNAL DE A KOVETKEZO CIMRE MUTAT.
; NUMERIKUS RUTINOK
; A, HL SZAM KIIRASA MEMORIABA VAGY DEF CSATORNARA
HEXHLDE:            LD A,H           ;HL-T 4 SZAMJEGGYEL
                   CALL HEXADE      ;DE CIMRE IRJA.
                   LD A,L
HEXADE:             LD A,L
                   PUSH AF           ;A-T 2 SZAMJEGGYEL
                   RRCA              ;DE CIMRE IRJA.
                   RRCA
                   RRCA
                   RRCA
                   CALL HEXADE10
                   POP AF
HEXADE10:          CALL HEXA         ;A ALSO NEGY BITJET
ADEWR:             LD (DE),A        ;1 SZAMJEGGYEL
                   INC DE           ;DE CIMRE IRJA.
                   RET
HEXHLPRINT:        LD A,H           ;HL-T 4 SZAMJEGGYEL
                   CALL HEXAPRINT   ;A DEF. CSATORNARA
                   LD A,L           ;IRJA.
HEXAPRINT:         PUSH AF         ;A-T 2 SZAMJEGGYEL
                   RRCA            ;A DEF. CSATORNARA
                   RRCA            ;IRJA.
                   RRCA
                   RRCA
                   CALL HEXAPRINT10
                   POP AF
HEXAPRINT10:      CALL HEXA         ;A ALSO 4 BITJET KIIRJ/
APRINT:           LD B,A           ;A TARTALMAT KIIRJA.
BPRINT:           LD A,255        ;B TARTALMAT KIIRJA.
                   EXOS 7
                   RET
SPPRINT:          LD B," "        ;SZOKOZT IR.
                   JR BPRINT
CRLFPRINT:        CALL STRPRINT   ;UJSORJELET IR.
                   DEFB 13,10,0
                   RET
DECHLPRINT:       PUSH HL         ;HL-T KIIRJA
                   CALL HL BCD     ;DEF CSATORNARA
                   JR DECAPRINT10
DECAPRINT:        PUSH HL         ;A=T KIIRJA
                   CALL A BCD      ;DEF CSATORNARA
DECAPRINT10:     PUSH IX
                   LD IX,APRINT
                   JR DECOU0
DECHLDE:          PUSH HL         ;HL=T KIIRJA
                   CALL HL BCD     ;DE CIMTOL
                   JR DECADE10
DECADE:           PUSH HL         ;A=T KIIRJA
                   CALL A BCD      ;DE CIMTOL
DECADE10:         PUSH IX
                   LD IX,ADEWR
DECOU0:           CALL DECOU
                   POP IX
                   POP HL
                   RET
DECOU:            BIT 7,C
                   JR Z,DECOU10
                   DEC B
                   JR DECOU20
DECOU10:          CALL DECOU40
DECOU20:          LD A,H
                   CALL DECOU30
                   LD A,L
DECOU30:          PUSH AF
                   RRCA
                   RRCA
                   RRCA
                   RRCA
                   CALL DECOU40
                   POP AF
DECOU40:          CALL HEXA
                   DEC B
                   JR Z,DECOU60
                   BIT 1,C
                   JR NZ,DECOU60
                   CP "0"
                   JR Z,DECOU50
                   SET 1,C
                   JR DECOU60
DECOU50:          BIT 0,C
                   RET Z
                   LD A," "
DECOU60:          BIT 7,B
                   RET NZ
                   PUSH BC
                   CALL JPIX
                   POP BC
                   RET

```

```

A_BCD:            PUSH BC         ;A=T ATVALTJA
                   PUSH DE        ;BCD KODRA
                   LD H,A
                   LD B,8
                   LD DE,0
A_BCD10:          ADD HL,HL
                   LD A,D
                   ADC A,A
                   DAA
                   LD D,A
                   LD A,E
                   ADC A,A
                   LD E,A
                   DJNZ A,BCD10
                   EX DE,HL
                   POP DE
                   POP BC
                   LD B,3
                   RET
HL_BCD:           PUSH BC         ;HL-T ATVALTJA
                   PUSH DE        ;BCD KODRA
                   LD DE,0
                   LD C,E
                   LD B,16
HL_BCD10:         ADD HL,HL
                   LD A,E
                   ADC A,A
                   DAA
                   LD E,A
                   LD A,D
                   ADC A,A
                   DAA
                   LD D,A
                   LD A,C
                   ADC A,A
                   LD C,A
                   DJNZ HL,BCD10
                   EX DE,HL
                   POP DE
                   POP BC
                   LD B,5
                   RET
HEXA:             AND 15         ;A ALSO 4 BITJET
                   ADD A,"0"      ;ASCII KARAKTERRE
                   CP "9"+1       ;KONVERTALJA.
                   RET C
                   ADD A,7
                   RET
;SZTRINGKIIRO RUTINOK.
; A HIVAST KOVETO DEFB ADATOK KIIRASA NULLAIG.
; IA SZTRINGET NULLA BAJTTAL FELTETLENUL ZARJUK LEI
; IA RUTINOKKAL NULLA BAJTOT NEM LEHET KEZELNI!
STRDE:            EX (SP),HL      ;A HIVAST KOVETO
                   PUSH HL        ;SZTRINGET A DE
                   CALL BCKOMP    ;MEMORIACIMTOL
                   EX (SP),HL     ;KEZDODOEN ATMASOLJA.
                   OR B
                   JR Z,STRDE10
                   LDIR
STRDE10:          POP HL
                   EX (SP),HL
                   RET
STRPRINT:         EX (SP),HL      ;A HIVAST KOVETO
                   LD D,H         ;SZTRINGET A DEFAULT
                   LD E,L         ;CSATORNARA KIIRJA.
                   CALL BCKOMP
                   LD A,255
                   EXOS 8
                   EX (SP),HL
                   RET
BCKOMP:           XOR A         ;BAJTOK SZAMLALASAT
                   LD B,A         ;VEGZO ALRUTIN.
                   LD C,A
                   CPIR
                   LD A,B
                   CPL
                   LD B,A
                   LD A,C
                   CPL
                   LD C,A
                   RET
JPHL:             JP (HL)
JPIX:             JP (IX)
JPIY:             JP (IY)

```


Jó az egér a háznál, ha jó...

Mármost az egér. Van ugyanis rossz egér is, itt van mindjárt az ENTERPRISE eredeti egere (rengeteg e eme helyen), amit a népnyelv némi jóindulatú túlzással egyszerűen csak "patkányoknak" nevez. Ennél a monstrumnál több nagyságrenddel finomabb akármelyik IBM PC-re tervezett társa, ráadásul az olcsóbbak az EP-egér eredeti árának felébe sem kerülnek. Csupán egy a gond: hiába keres a reménykedő géptulajdonos, egyetlen olyan csatlakozót sem talál hón szeretett számítógépén, amelybe a csodaegeret bedughatná.

-Ez most már mindig így lesz? - kérdezi hangjában nem kevés csaldottsággal. Válaszunk:

-Nem, ezentúl minden másképp lesz!

Elkészült ugyanis egy soros vonali illesztőkártya, amely elsősorban éppen az ENTERPRISE számítógép és a PC-egér mindkét beteljesíthetetlen vágyai beteljesülésének enged szabad folyást, de ezen kívül még másra is jó. (Lapzártakor még kutatjuk, mire is jó ezen kívül a kártya, minden ötletet szívesen fogadunk...)

A tájékozottabb felhasználó megkérdezheti:

-Vajh nem lehet-e az ENTERPRISE meglévő soros portjához csatlakoztatni ama fránya pécé-egért?

Mire mi:

-Naná, hogy lehet! Csak nem érdemes! Ugyanis ez a "soros" port programmal alakítja át a beérkező soros információt párhuzamossá, és ekközben a processzor mással nem is törődik, ami igen nagy hiba, így nem lehet pl. egy rajzolópogramban emberi sebességgel és folytonossággal mozgatni a kurzort. (Ezt onnan tudjuk, hogy elképesztően okosak vagyunk, másrészt kipróbáltuk.)

Ezért aztán megszületett a mellékelt rajzon látható igen egyszerű felépítésű kártya. Az eszköz lelke az Intel 8251A (U1) típusú soros vonali adó-vevő áramkör. U2 illeszti az adatvonalakat a számítógéphez, U6 és U7 pedig a soros ki- és bemeneteket a külvilághoz. A csatlakozópontokban látható számok a szabványos 25-pontos CANNON csatlakozó lábszámai, a felettük látható szövegek a vonalak RS-232 megnevezései.

U3 a címekódor: a 00 I/O címhez rendeli U1 adatregiszterét, a 01 I/O címhez pedig a kontroll-regiszterét. U4 fele egy kvarcoszcillátort alkot, amely egyrészt közvetlenül szolgáltatja U1 órajelét, másrészt U5-tel 2 hatványai szerint leosztva a soros vonal ütemfrekvenciáját (Baud rate) határozza meg. JP1-gyel állíthatjuk be ezt a frekvenciát, amit U1 belül még 16-tal leoszt, így jönnek ki a jelölt ütemértékek. PC-egér illesztéséhez az 1200-as beállítás kell alkalmazni.

U4 maradék része szinteket fordít. Külön figyelmet érdemel az U4D - D1 vonal, amelyen keresztül U1 megszakítást tud adni a gép processzorának, lehetővé téve ezáltal, hogy az két bajt beérkezése kö-

zött egy csomó hasznos dolgot is csináljon.

Szerepel még a rajzon egy feszültség-stabilizátor (U8), amely a +5 voltot állítja elő az IC-knek, nem szerepel viszont (bár a kártyán természetesen rajta van) a +/- 12 voltos előállító áramkör, melyből főleg a -12 voltos rész eszenciális az egérke illesztése szempontjából, de a rajtot főlélegesen bonyolította volna.

Most már tudjuk, hogyan működik a kártya, csak azt nem tudjuk, hogy minek, azaz hogy jön ez az egész össze az egérrel?

Először is a PC-egerek egy nagy (és olcsóbbik) része soros vonalon csatlakozik a géphez. Két jelentősebb csoport terjedt el: a Microsoft-kompatibilis vagy "két gombos", és a Mouse Systems-kompatibilis vagy "három gombos" egerek. Az egyszerűség kedvéért csak a második társulattal foglalkozunk, az első is meglepően hasonló működésű.

Eme jószágokat működési módjuk szerint is kategorizálhatjuk, úgy mint optikai egerek, mágneses egerek, mechanikus működésű, avagy golyós egerek, egyéb egerek. A megfizethető árkategóriában a mechanikus egerek dőnek, név szerint kiemelhetjük a legeslegolcsóbb, de azért remekül használható Genius GM-6 típust, a továbbiak is elsősorban reá, és árban is közeli rokonaira vonatkoznak.

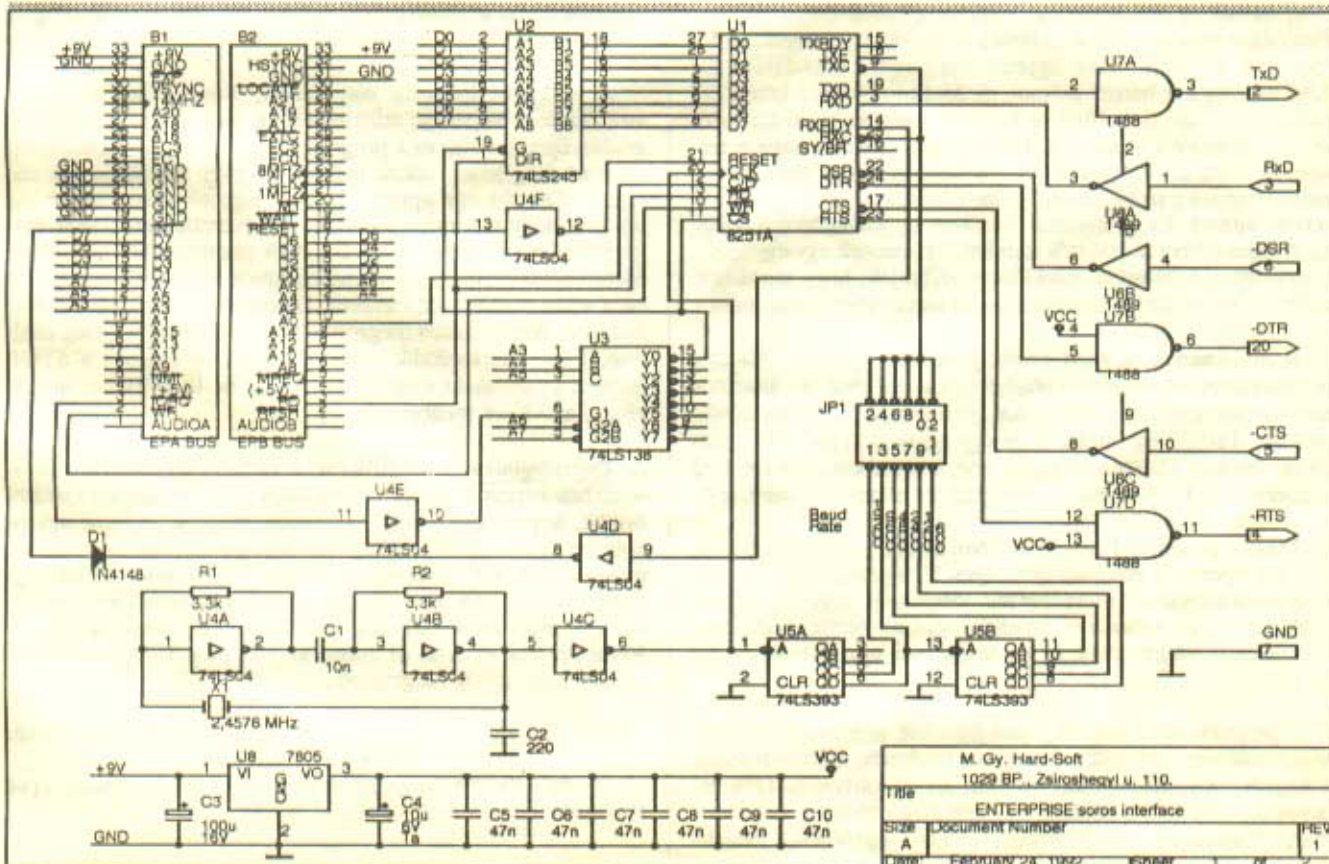
Mindenekelőtt az egérnek némi tápfeszültségre van szüksége a törhető működéshez, amelyet a soros port -DTR lábán megjelenő -12 voltból nyer. Ha ez megvan, már érdemes mozgatni is a jószágot. Ilyenkor az egér hasábnak lakó golyó két, egymásra merőleges tengelyű lyukacsos tárcsát forgat meg, melyek elmozdulásának mértékét és irányát két-két optokapu detektálja. Ezek jelei az egér külön bejárattal mikroprocesszorába jutnak, amely kiszámolja, hogy mekkora elmozdulások történtek X, ill. Y irányban, és ezeket előjeles számokká alakítja. Hozzácsapja még a három gomb állapotát (i. hogy le vannak-e nyomva, avagy nem), és az egész kupaot (öt bajt) elküldi a soros vonalon, 1200 bit/s (Baud) sebességgel minden esetben, ha az egér állapotában bármely változás áll be (odébblokkolták és/vagy megnyomták egy vagy több gombját).

Kissé részletesebben: az első bajt alsó három biteje a gombok állapota invertálva. A második és a negyedik bajt összege jelenti az X-irányú, a harmadik és ötödik bajt összege pedig az Y-irányú előjeles elmozdulást. Innen kezdve a többi már a számítógépben futó programon múlik...

Utóirat: Ha valakit érdekel a dolog, az alanti címen rendelheti meg a kártyát, akár egerestől. Egy órdögös program is készült a kártyához Devilsoft műhelyében. Vagy a kártya készült a programhoz, ha Devilsoftot kérdezik, ez nézőpont kérdése.

M. Gy. Hard-Soft

1029 Bp., Zsárosszegyi út 110.



PCBPRISE 1.2: Komoly alkalmazás

Mire jó az ENTERPRISE? Mindenre! A PCBPRISE programmal például nyomtatottáramkör-tervezésre. S nem is akár-hogy...

A program természetesen nem alkalmas ipari termelésre, mert a fejlesztők (a program *Dr. Kövesdi István* munkája, a külső alkatrészkönyvtárat *Kozma Miklós* készítette) az amatőrök igényeit igyekeztek kielégíteni, tűrhető sikerrel. *20x20 cm-es*, vagy kisebb panelra *legfeljebb 200 alkatrészből álló* kapcsolást vihetünk fel háromféle - digitális, analóg, és rádiófrekvenciás - stílusban. *A rétegek száma 256 lehet*, de szerencsére a megcélzott felhasználói kör lehetőségei aligha mutatnak túl a kétrétegű NYÁK-lemezen. A felbontás *1/8-ad*, vagy *1/16-od* raszter (a raszter *1/10-ed inch*, pl. két IC-láb távolsága), ami több, mint megfelelő. Két IC-láb között egy vezeték vihető át, ennél többet nem is igen engednek a szélesebb körben hozzáférhető NYÁK-gyártási lehetőségek.

Egyetlen, de nem apró probléma a tervből gyártófilm előállítása. Első ránézésre ez legalább akkora erőfeszítést igényel, mint maga a tervezés. A program csupán nyomtatót kezel, ami tulajdonképpen érthető: ritka az olyan felhasználó, aki a tizenvalahányezer forintos ENTERPRISE-ához beszerez egy csekély kétszázézer forintos HP rajzgépet. Másrészt az a benyomásom, hogy a program a vezetősávok helyét pontonként tartja nyilván, és ebből önmagában szép feladat egy plotterhez illeszkedő vonalas adatszerkezet létrehozása. *1 szó, mint 47, a program csak nyomtatni tud*, ill. lemeze is tud írni olyan fájlt (dot matrix file), amit aztán egy másik, a saját nyomtatónkhoz illeszkedő programmal kinyomtathatunk. Tapasztalataim szerint mátrixnyomtatóval készült rajzról meglehetősen rossz minőségű gyártófilm készíthető, amely még sok retusálást igényel. De sebjaj, egy kelően elszánt felhasználó előtt úgy sincs akadály...

Nézzük ezek után, mit kap a felhasználó, ha nem habozik, és vásárol egy ilyen programot. Kapja természetesen a programlemez a fájlokkal, egy gondosan készített leírást, és egy kis könyvecskét, az ún. kódkönyvet (jelen esetben *Babits Mihály: Hatholdas rózsakeri című* kötetét). Ez utóbbi a programvédelem célját szolgálja. Néha ugyanis a program a menüablak bal felső sarkába kír három számot, pl. 35 12 9. Ekkor a könyv 35. oldalán a 12. sor 9. betűjét be kell írni a gépbe, ha tovább szeretnénk haladni a munkával. Elmés, ugye? Nekem nagyon tetszik, és mint az EP-piacon forgolódo személy, a legteljesebb mértékben meg tudom érteni a szerzőt. A szabad programcserebe kitűnő dolog, egyúttal azonban megakadályozza, hogy tehetséges emberek jelentős munkát fektessenek egy-egy szoftvertermék előállításába, mivel nem várhatják, hogy munkájuk valaha is megtérül. A hírnevtől pedig simán éhen lehet halni.

A dokumentáció, mint írtam, gondosan készített. Minden programfunkciót elegendő részletességgel tárgyal, az általános részben pedig hasznos tanácsokat ad a tervek hatékony előállításához. Tartok azért tőle, hogy egy egészen kezdő felhasználónak túlzottan tömör a tárgyalásmód, lehetne kissé bővebb lére eresztetni mind a parancsok ismertetését, mind a jótanácsokat.

Maga a program Turbo Pascalban fródott, a futtatásához az IS-DOS operációs rendszer szükséges. Ez egyúttal azt is jelenti, hogy a felhasználható memória mindössze 64K, ezért a program nyolc részre lett szétosztva, amelyek egymást hívják, némi diszkpörgetéssel, várakozással és körömrágással egybekötve, de hát ez van...

A program lefed minden olyan funkciót, ami a NYÁK-tervezéshez szükséges. Megadhatjuk a panel méretét, módosíthatunk, elforgathatunk, átrendezhetünk már létező kártyát is. Igen érdekes (és hasznos) lehetőség téglalap alakú blokkok önálló elmenthetősége és behozhatósága. Ennek segítségével "bevált"

áramköri részleteket ültethetünk át egyik panelról a másikra. Elhelyezhetünk alkatrészeket a panel mindkét oldalára, hibaelenőrzéssel (hogy beleér-e már létező alkatrészbe), vagy anélkül, forgathatjuk, tologathatjuk azokat. Ha készen vagyunk, nyomtathatunk beültetési rajzot is.

Az összeköttetéseket megadhatjuk grafikusán, vagy karakterekkel, ill. csomópontokkal, vagy kötéscenként. (Ha valaki nem értendő, mi ez, ne ijedjen meg, a program használatakor rövid úton rájön majd.) Kétféle vonalvastagság között választhatunk, de lehetőség van a súlyosabb vezeték (föld, tápfesz) automatikus kiterjesztésére is. Kifarthatók a kötések és a szabadon maradt lábak is a képernyőre vagy a nyomtatóra. Ha sok ráérő időnk van, optimalizálhatjuk az alkatrészek és a kötések elhelyezkedését a legrövidebb kötéshossz, a legkevesebb kereszteződés, vagy egyszerre mindkettő szerint. A huzalozást (leg-alábbis részben) rábízhatjuk a számítógépre is. Ez ugyan jelentősebb kártyánál elég sokáig tart, viszont közben megfőzhetjük a vacsorát, kivasalhatjuk az ingeket, megnézhetjük a meccset...

Természetesen nem drótoz meg mindent a gép, a hiányzó vezetékkel kézzel berajzolhatjuk, és a már meglévő huzalozást is szerkeszthetjük. (A program védelmében meg kell jegyezmem, hogy a közkezen forgó IBM PC-s NYÁK-tervezők automatikus huzalozója is csak kb. 80 %-os, leszámítva a cég által mellékelte minta NYÁK-ot. Vannak tökéletesen huzalozó programok is, csak azok hosszú órákat futnak egy 33 MHz-es 486-os gépen, és vagy 1 millió forintba kerülnek.) Végül, ha már mindennel elkészültünk, csak a bevezetőben tárgyalt nyomtatás marad hátra, meg ami még azután jön...

A programrendszer lehetőséget ad Turbo Pascalban írt felhasználói modul illesztésére is, ami jó dolog, csak még nem világos, hogy mire. Létezik a program IBM-kompatibilis gépekre is (PCBCGA stb.), melynek megvásárlása esetén a szerző beszámítja a PCBPRISE-ért kifizetett összeget. Hurrá! (Azt csak mellékesen jegyzem meg, hogy a PC-s változatnak sokkal többet kell tudnia, ha versenyre akar kelni a már létező profi programokkal.)

A program általános megjelenése, használhatósága jónak mondható. A képernyő áttekinthető, a funkciók követhetők. A grafika újrarajzolása és a programváltás embertelenül lassú, ezt egyszerűt meg lehet szokni, másrészt amúgy sem lehet mit csinálni... Csupán egy apróbb és egy nagyobb dolog zavar: Az apróbb, hogy szerintem lehetnének a menüfeliratok akár magyarul is, a súlyosabb, hogy bizonyos parancsokból nem lehet ésszerűen kilépni. Pl. az összeköttetések képernyőre listázását nem lehet megállítani, ráadásul képernyőnként egy gombot le kell ütni, ez, ha közben meggondolta magát az ember, elég unalmas. Ehhez kapcsolódik még, hogy némely esetben a STOP gombot gyanútlanul lenyomva egyenest az IS-DOS-hoz jutunk vissza, ami kissé mellbevágó.

Összefoglalva: A PCBPRISE a kitűzött feladatot a lehetőségekhez képest korrekten oldja meg. A magam részéről örülök, hogy ilyen komoly alkalmazás született kedvező mikro-számítógépekre! (Bár azért ha NYÁK-ot óhajtok tervezni, megmaradok a P-CAD-nél, a jó öreg 386-os AT-n.) Persze jó lenne még egy kapcsolásrajz-szerkesztő, különböző tervezési ellenőrző programok, HP-GL interfész, meg toronyóra láncsal. Remélhetőleg Kövesdi úr magas kort ér meg közöttünk, és aktivitása még sokáig megmarad...

Mészáros Gyula

(A PCBPRISE megrendelhető: Dr. Kövesdi István, 1144 Budapest, Csertő út 18-20/134)

Ismét a CLICK-ről

Lukács Árpád olvasótársam már elpanaszolta, hogy az új CLICK (OUT 191, 34) esetén nem lehet betölteni egy fájlt sem. Ez azonban nem mindig igaz.

Ha a fájl kimentése előtt kiadjuk a SET FAST SAVE OFF parancsot, azt minden további nélkül vissza tudjuk tölteni így is.

A másik tapasztalatom talán érdekesebb.

Az OUT 191, 34 kiadása után jelentősen megváltozik a kimentés sebessége. (A hangmagasság megváltozásából is következtetünk erre.)

Táblázatban összefoglalva:

CLICK	FAST SAVE	BETÖLTÉSI IDŐ	VISSZATÖLTÉS
NORM.	ON	9 SEC.	CSAK NORM. CLICK ESETÉN
NORM.	OFF	18 SEC.	MINDKÉT ESETBEN LEHETSÉGES
ÚJ	ON	14 SEC.	MINDKÉT ESETBEN LEHETSÉGES
ÚJ	OFF	28 SEC.	NEM SIKERÜLT VISSZATÖLTENI

Butuza Tamás
- TOMSOFT -

Ilyen még a neppereknél sincs...

SPRED release 1.5

Felhasználóbarát Entersprite kompatibilis sprite editor

- Tömértelen funkció
- Pull-down menürendszer
- Esztétikus kivétel
- Exdos használat
- Beépített help
- Magyar nyelvű .WP leírás

Mindez 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 álland.



AEROCARITAS ALAPÍTVÁNY

Tisztelet Hozzájárulásaink!

Hazánk négy tájegységében működtetünk helikoptert. A regionális központok kialakításához még további 7-8 helikopter beállítása szükséges. A szolgálat fenntartása, az emberi életek mentése nem olcsó, de kisebb áldozatvállalással egyre több életet tudunk megmenteni, és ez mindent megér. Az Aerocaritas Mentőszolgálat embereinek akaratából az emberért munkálkodik.

Számlaszám: POSTABANK RT. 289-98943/026-00321

Fizessen elő a Computerworld-Számítástechnikára! Csak nyerhet!

Informatikai Iparunk vezető lapja a hetente megjelenő Computerworld-Számítástechnika. Hírelt, információit, elemző írásait és tesztjeit csaknem mindenki olvassa, aki - akár fejlesztőként, akár kereskedőként - e területen tevékenykedik.

De mint a csúcstechnológia mértékadó hírlapja, nem csak a szó szoros értelmében vett szakemberek számára ad fogódzót a számítástechnika (számítógépek, számítógépekre írt programok), a számítógépes hálózatok, a távközlés és egyéb informatikai alkalmazások világában. Üzleti információit a vállalkozóknak és beruházásokkal foglalkozó vezetőknek is adhatnak busásan kamatozó ötleteket. Műszaki kérdésekkel foglalkozó cikkel a legfrissebb információkkal szolgálnak a hazánkban és a nagyvilágban megjelenő újdonságokról.

Gyorsan változó hazai piacunk trendjeiről árulkodnak a lapban hétről hétre megfrissülő hirdetések. A beruházások tervezésekor segítséget nyújtanak a megfelelő számítástechnikai eszközök kiválasztásában. Egy közelmúltban készített közvélemény-kutatás eredményeivel szerint a Computerworld-Számítástechnika olvasóinak háromnegyede vette figyelembe döntés-előkészítéskor a lapban közzétett hirdetéseket.

A Computerworld-Számítástechnika előfizetési díja

fél évre: 1098 Ft

egy évre: 2196 Ft

Legyen az előfizetőnk!

A Computerworld-Számítástechnika kiadója:

IDG Lapkiadó Kft. 1072 Budapest, Rákóczi út 16.

Telefon: 111-7917, 122-3293 Fax: 142-3965



LITTLE PUFF IN DRAGONLAND

A Dizzy tojásból kikelt egy kis sárkány. (Bizonyára sokáig ültek CODEMASTERS-ék a tojásukban.-EPY) Így született meg a kis Puff, aki az iskolaév végeztével befizetett egy olcsó DRAGIBUSZ útra, Skóciába. Az egész nyarat itt töltötte, már csak két nap volt az iskolakezdésig. Ahogy pakolta össze a ruháit észrevette, hogy az útlevelet elvesztette. Leült, gondolkodott egy picit, és eszébe jutott, hogy már otthon is kiestek a lapok a passzportjából. Eldöntötte, hogy végigjár minden utat, amerre a két hónap folyamán sétált, talán megtalálja az útlevelet.

Itt kapcsolódunk be a játékba. Az irányítás történhet *internal* (3. SINCLAIR), *external* (2. KEMPSTON) *joystickkal*, illetve *billentyűzettel*, amit be is állíthatunk (4. DEFINE). Egyébként beállítás nélkül a billentyűzetkiosztás: [O] - balra, [P] - jobbra, [Q] - fel, [A] - le, [SPACE] - tűz. Tárgyat felvenni a joy lefele húzásával tudunk (Tiszta Dizzy.-EPY), és egyszerre három tárgy lehet nálunk. (Ez is tiszta Dizzy.-EPY) Ha felvesszünk egy tárgyat, akkor a legelső tárgy kihullik a kezünkéből. A játékot a tűzgomb megnyomásával indíthatjuk.

A startpályán vegyük fel a cseresznyét, amely plusz 200 pontot jelent. (Egyébként minden étel megemésztésekor 200 ponttal gazdagodunk.) Láthatunk itt egy kapcsolót is, amire ugorjunk fel. Ha leejöünk róla, a kapcsoló újra felemelkedik. Ez így sehogyan se jó. Keresni kell valamit, amit rá tudunk tenni, és így lent marad. Menjünk hát jobbra. Na, itt van a határőrsg. A túlóldalon már *Dragonland* földjét láthatjuk. De a határőr hangos *Pass!* kiáltással kéri az útlevelet. Hiába is próbálunk magyarázkodni, a határőr nem enged tovább. Vegyük fel az itt lévő üveget, melynek tartalmát igyuk ki. Valami kőkemény házipálinka lehetett, mert elfogyasztása után a tűzgomb hatására a kis Puff tüzet okád. Marad az egyetlen út, balra. Itt egy ház találatunk, de az ajtaja sajnos zárva van. Menjünk tovább! Most egy farönk állja az utunkat. De mire való legújabb fegyverünk, a tűz. Álljunk közvetlenül a rönk mellé, és fújjunk rá tüzet. Hoppá! A fa leégett, most már továbbmehetünk. A lyukat ugorjuk át, és menjünk végig ezen a szinten. Itt van az útlevel első darabja. Menjünk oda, és vegyük fel. Aja-jaja-jaj! A fejünkre pottyant egy kókuszdió! Inkább ne próbálkozzunk ezzel, hanem vegyük fel az itt heverő üveget. Huh! Ez bizonyára kávé volt, mert nem tudunk több tüzet okádni. Sebaj, most már úgysincs mit tenni. Menjünk visszafelé, és pottyanjunk le az előzőleg átugrott lyukon. Itt is egy lyuk. Essünk tovább! Ez egy rossz ötlet volt, hiszen *RICK DANGEROUS* példáját követve mi is néhány karó hegyén találtuk magunkat. Akkor merre? Menjünk talán balra. Itt van egy teknőc. Ugorjunk fel gyorsan a ládára (*BOX*). Ha a teknős elment alattunk, essünk le, menjünk a doboz elé, és vegyük fel. (Na végre, legalább egy tárgy van már a kezünkben!-EPY) Most menjünk tovább balra, de a teknőst ugorjuk át. (Akinék már sokadszorra sem sikerült, az próbálkozhat úgy is, hogy amikor felvette a ládikót, akkor menjen balra. Mikor a teknőc szembe jön velünk, akkor tegye le gyorsan, ugorjon fel rá, majd ha a teknős ismét átment alattunk, akkor essünk le ismét, vegyük vissza, és már át is jutottunk a képernyőn.) Na, itt meg a vizipók mászkál, alatta pedig van valami úszógumi. Menjünk oda. De utunkat állja egy meredek fal. Tegyük le a dobozt, ugorjunk fel rá, és így már felvehetjük a mentőövet (*LIFE BELT*), de a csodapókra vigyázzunk!

Menjünk vissza jobbra, de ne hagyjuk ott a ládát sem. (Ez kivételesen nem frigidáda.-EPY). Most már leeshetünk a lyukon. Hát ez nem jött be! Akkor az úszógumi talán valahol máshol kell. Talán vízhez... Ugorjunk át a lyukat.

A következő pályán egy pisztolyt (*GUN*) találatunk, de ezt hagyjuk itt. Azután pedig egy dugóhúzó (*DECORKER*) látunk. Ezt vigyük magunkkal, majd továbbhaladva vegyük fel az útlevel következő darabját is. Ezekután egy újabb lyukat látunk. Itt ne essünk le, hanem ugorjunk át rajta! Egy lábpumpát (*FOOT-PUMP*) találatunk. Vegyük fel! Jaj, de már több, mint három tárgyunk van! Hagyjuk itt a dugóhúzót! Ezek után leeshetünk a lyukon keresztül. Itt egy bányászisakot (*TINHAT*) találatunk. Ne vegyük fel, hiszen az úszógumi kell a vízhez, a pumpa kell

az úszógumihoz, a ládikó pedig azért kell, hátha megint lesz egy magas fal. Viszont a hamburgert, mit hamburgert, DUPLA hamburgit zabáljuk meg.

Most menjünk át balra a vízben, de vigyázzunk, nehogy a felettünk szálldosó madár letojjon minket egy Dizzy tojással. (Mert ha lepottyant minket, akkor azt mondja: Le vagy tojva, és kezdetjük előlrol az egészet!-EPY) Ha átértünk, akkor két kukacot látunk mászkálni. Menjünk tovább szép lassan, nehogy hozzáérjünk valamelyikhez. Az itt lévő villával (*FORK*) ne törődjünk, hagyjuk itt. A következő pályán együk meg a szőlőt (Ilyen egy fzlést! Sárkány létere szőlőt?!-EPY) A lyukat átugorva egy házban találjuk magunkat. Itt egy kalapács (*HAMMER*). Vegyük fel úgy, hogy nálunk legyen a láda is. A kalapáccsal széttörtük a ládát, és egy kulcsot találtunk benne, rajta egy 22-es számmal. Van itt egy ajtó is, próbáljuk meg kinyitni. Sikerült!!! A kulcsra már nincs szükség, de a pumpát és a mentőövet ne hagyjuk itt.

Ez a ház ismerős. Ez volt zárva a játék elején. A startpályától balra vagyunk. Most tegyük meg ismét az előző utat vissza az ajtóig, de most hozzuk el a sisakot is. (Balra, le, jobbra, le, sisakot felvenni, balra a házig. Azért mentünk körbe, mert a gilisztákat elég nehéz átugrani, de aki ügyes, az mehet a középső szinten is jobbra!) Most menjünk ismét át az ajtón, tegyük le az úszógumit és a pumpát, és menjünk teljesen jobbra. Most már bátran felvehetjük a kókuszdió alatti útlevelet, a sisakot megvéd. Valamint vegyük fel a kókuszdiót is. A sisakot hagyjuk itt. Majd menjünk a kókuszdióval a startpályára. Tegyük rá a kapcsolóra, amely most már lent marad. Vegyük vissza az ajtónál hagyott tárgyakat, és a már kétszer megtett utat (a vízben át) sétáljuk végig harmadszorra is, de a középső szint jobb oldalán hagyott dugóhúzó vegyük fel! Ha visszaértünk az ajtóhoz, akkor menjünk tovább balra, és vegyük fel a banánt is útközben. (Na, most ez majom, vagy sárkány?-EPY)

Ha elérkeztünk a következő házba is, akkor tegyük le a pumpát és az úszógumit. Vegyük fel az itt lévő lámpaernyőt (*SHADE*), valamint az alatta rejtőző üveget (*BOTTLE*) is. A dugóhúzó segítségével kinyitottuk az üveget, és abból elő is került az útlevel harmadik darabja. (A kis sárkány biztos kiitta a tartalmát valamikor, és nem éppen józan állapotában beledugta az útlevelet!-EPY) A lámpaernyőt el is dobhatjuk, majd haladjunk tovább balra.

Az út végén feltehetjük rá a útlevel utolsó darabjára. Most menjünk vissza az ajtón keresztül a skót határörhöz, de vegyük fel a kókuszdiót is. A határőr átenged, de ezzel még nincs vége a megpróbáltatásoknak! Ugyanis a *Dragonland*-i határőrt is meg kell győzni arról, hogy mi át fogunk menni ezen a vasfüggönyön.

Vegyük fel a fűrészt! A kókuszdiót szétfűrészelve egy arany pénzérmét (*GOLD COIN*) tartunk a markunkban. A másik határőr nem elégszik meg az útlevellel, hanem még vámot is kell fizetnünk. Hiába magyarázkodunk, hogy nincs nálunk semmi elvámolnivaló, akkor is le kell fizetnünk.

Továbbmenve ugorjunk fel a hídra! A következő pályán régi ellenségünk, egy szörny próbálja hazatérésünket megakadályozni. Ugráljunk át, de vigyázzunk a fejünk felett köröző madárra is! Ha itt is átjutottunk, akkor végre megláthatjuk otthonunkat, és a mellette már kidülöfőiben lévő táblát: "HOME, SWEET HOME" (Othon, édes othon. Fordította: EPY, az angol nyelv doktora!-EPY) Végre hazaértünk. De az örömben egy kis öröm is vegyül, mert holnap reggel kezdődik a sulii! Nem baj! Jövöre Puff a világ másik végére, Új-Zélandba megy nyaralni!

CONGRATULATIONS, YOU'VE SUCCESSFULLY GUIDED LITTLE PUFF SAFELY HOME!!! (Gratulátók, sikeresen és szerencsésen hazavezettél Puff-ot!!!)

-JOVI-

LITTLE PUFF IN DRAGONLAND	
Grafika:	6
Zene/FX:	1
Játszhatóság:	5
Összhatás:	5

THE UNTOUCHABLES

Manapság egyre több olyan játék jelenik meg a piacon, melyek egy-egy filmet dolgoznak fel (pl. *Indy*, *Back To The Future*, *Ghostbusters*, *Tenage Mutant Ninja Turtles*, *Platoon* és még folytathatnám a sort). Ilyen a most ismertetésre kerülő **THE UNTOUCHABLES** című játék is, amelynek alapötletét a hazai mozikban "Aki legyőzte Al Caponét" címmel vetített film szolgálta.

1930-at frunk. Az USA egész területén szesztilalom van, így Chicagóban is. *Al Capone* és társai veszik át az irányítást a város felett. Ezt megtehetik, hiszen a szeszcsempészet óriási vagyont hoz számukra. Az utcákon napról-napra, sőt óráról-órára újabb gépfegyveres harcok dúlnak a szeszcsempész bandák között. A rendőrség tehetetlen a hihetetlen túlerővel szemben. Nem marad más lehetőség a csempészbandák megfékezésére: a gengszterek ellen a saját eszközeikkel, azaz fegyverrel lehet csak szembeállítani. De ki lehet erre alkalmas?

Ekkor érkezik a városba egy titkosügynök, aki nem más, mint *Elliot Ness*. Néhány nap múlva meg is kezdődik a véres lövöldözés, a kegyetlen harc a gengszterek bandái ellen. A fő cél pedig: *Al Capone* uralmának letörése, és jobbkezeit, *Frank Nitty*-t megölni.

A játék, mely teljesen a film történetéhez igazodik összesen hat szinten keresztül játszódik. Ezek az alábbiak:

* **Áruház (WAREHOUSE)**: Chicago legnagyobb áruházának a raktárában vagyunk. A napokban feltűnt a rendőrségnek, hogy az áruház polcai elkezdtek üresedni, pedig a raktár - állítólag - tele van. No de mivel? Néhány nappal később egy rendőrségi információból megtudjuk, hogy a raktárba nem tartanak mást, csak a gengszterek által csempésztett italt. Ekkor lép először akcióba *Ness*. A helyszínt oldalról látjuk, és a célunk, hogy minél több gengsztert elfogjunk. Ha már elegendőt tartunk a markunk között, akkor máris a következő szinten találjuk magunkat.

* **A híd (THE BRIDGE)**: A rendőrség újabb információval szolgál: Az USA-Kanada határon ma délután többszáz doboz szeszt próbálnak átjuttatni. *Ness* kapásból a helyszínre siet AK jelű szovjet géppityujával, ahol ellenállásba ütközik. Kénytelen

hátrálni, de egy mélyedés kiváló lövészárókként szolgál. Itt hasalunk, s lövöldözzük a rosszfúkat. Jobbra-balra gurulva a híd újabb és újabb pontjait láthatjuk. A kép alsó részén egy távcsőben kinagyítva láthatjuk azt a részt, ahová lövünk.

* **A sikátor (THE ALLEYS)**: A vasútállomásra szeretnének eljutni, de természetesen az eddigi tetteink után a csempészek ezt nem nézik jó szemmel. Egy szők sikátoron kell átmennünk, de az ablakokon kihajoló gengsztereket is le kell lőnünk, természetesen.

* **A pályaudvar (RAILWAY STATION)**: Szegény *Ness* barátunkra újabb feladat vár. Az állomáson egy babakocsit kell végigvinnie, természetesen benne a babával. Igen ám, de *Al Capone* emberei mindenhol ott vannak, így itt is szorgalmasan lövöldöznek a lépcsőn leguruló babakocsira, amit az idő gyors múlása miatt néha-néha toljunk meg egy picit.

* **Mentsük meg a túszt (RESCUING THE ACCOUNTANT)**: *Al Capone* jobbkeze, *Frank Nitty* látva a vég közeledtét, elrabolta *Ness* legjobb barátját, bajtársát. A célunk természetesen a túszt megmentése, azaz *Nitty* letöltése. Erre egy pisztoly áll a rendelkezésére, melyben sajnos csak egyetlen egy golyó van.

* **Vadászat a tetőn (THE ROOFTOP CHASE)**: Az utolsó pálya hasonlít a sikátorok nevéhez, ugyanis itt is a kihajló illetve előttünk elfutó embereket kell lelőnünk, viszont itt egy hatlövétű winchestert szorít *Ness* a kezében. Ha sikerült ez a pálya is, akkor láthatjuk, ahogy *Al Capone* lezuhan a tetőről egyenesen *Ness* autójának a tetejére.

-JOVI-

THE UNTOUCHABLES	
Grafika:	7
Zene/FX:	1
Játszhatóság:	6
Összhatás:	5

THE CURSE OF SHERWOOD

Nehezen veszem rá magam, hogy erről a digitális szörnyűségről egyáltalán említést tegyek. Végül mégis rávettem magam a leírás elkészítésére, ugyanis rájöttem, hogy idegesítő, nehéz hétköznapi ez a program mégis hoz egy kis megnyugvást, és nem is olyan szörnyű, de csak addig, amíg meg nem fejtjük. Miután megfejtettük, jobb lesz a programot tartalmazó adathordozót (lemez vagy kazetta) jól elrakni szem elől, mert ha ezután még egyszer betöltjük, a jó sors kímélje meg szeretett gépünket a rapityára zúzástól!

Elhatároztam hát, hogy megkínzom az ENTERPRISE-okat Magyarországon, ezért papírra vettem e programnak nevezett nemtudoménmicsodának a megfejtését, s ennek hatására a géptulajdonosok saját maguk teszik tönkre a gépüket - vagy nem.

Úgy emlékszem, a program elején a gyártó nem reklámozza magát nevének a feltüntetésével (érthető okokból), vagy mégis kifizéri a nevét a program elejére, csak már nem emlékszem rá, ugyanis már több, mint 3 hónapja nem töltöttem be. (Tán ezért működik még a gépem.)

A program tulajdonképpen arról szól, hogy a *sherwoodi erdőt* egy szép nyári napon különböző alvilági rosszcson teremtések lepik el, és a hajdan nyugodt éjszakák folyamán az erdőből kísértávként fokozatosan uralmuk alá akarják hajtani Angliát, hogy a Gonosz uralkodjon az ottani emberek fölött. Az egykor nyugodt erdőt, *Robin Hood* lakhelyét ekkor hirtelen átok iltette meg. Ezt az átkot kell megtörnie egy szentéletű embernek, egy szerzetesnek, úgy, hogy a veszéllyel teli erdőn egy éjjel keresztül-vágva be kell zárni a *Gonoszok Kapuját*, mely az erdő másik

végében levő erdőben található egy veszélyes tűzokádó sárkánytól őriztetve. Ezt a kaput egy szent ezüstkereszttel lehet bezárni, de előbb azt meg kell szerezni egy gonosz főpaptól.

A szerzetes természetesen a Tisztelt Játékos. Ennyit a történetről, és most következzen a lényeg!

Ami a program irányítását illeti, nem scrolloz, hanem változtatja a képernyőket.

A START-képernyőről induljunk jobbra, a következő képernyőn az ott szambázó elég rossz kinézetű bácsit löjük le. Ezen nemes cselekedet után indulunk megint jobbra, a következő képernyőn létrejövő kisebb katonai támadást (hármán egy ellen) visszaverve az egyik katona által elejtett pajzsot nem fölvéve (mert semmi hasznát nem tudjuk venni, és fölösleges dolgokat meg minek cipeljünk magunkkal?) sétáljunk el a következő képernyőre. Ott találunk egy püspököt. Mikor épp megszólítanánk, hogy: "Helló, kolléga!", észrevesszük, hogy bánt. Tehát, levonjuk a logikus következtetést, hogy: ellenség, álpap, azaz egy püspöknek öltözött gonosz alvilági teremtmény. Viszonzásul löjük le... A bunkósbotot, amit leejtett, vegyük fel, ennek segítségével kardok helyett bunkókat hajigálunk.

Az ezután jobbra levő képernyőn a csontvázakat törjük össze, majd sétálunk jobbra. Ott a két csontvázal az előbb ismerttetett módon járunk el, majd beszélünk a házba, az ott levő csúnya néni leütjük, az általa elejtett varázsgömböt felvesszük, majd szép angolosan távozzunk. Innen csak balra lehet menni. (Micsoda felfedezés!) Akkor legyünk kedvesek egy képernyőnyit az említett irányba menni, majd ott le. Az ott levő

pár darab madarat lecsapva jobbra, és ott egyenesen a gombák által alkotott körbe be. Ez - bármennyire hihetetlen -, teleportál minket egy másik helyszínre. Az ott levő kaput törjük be egy bunkóval, menjünk át rajta, majd a baromfik kilövésével le. Ott van egy álpap, ez őrzi a keresztet. Diszkrétan verjük fejbe, vegyük fel a keresztet, bunkóinkat cseréljük le az ezüstkére, amit a továbbiakban utasítás nélkül ne cseréljünk le semmire, a már ismert (és egyetlen) útvonalon menjünk vissza egész a START képernyőig. Onnan induljunk el balra minden utunkban levő teremtményt kilőve addig, míg el nem jutunk egy háziköig. Ebbe még ne menjünk be, hanem egy képernyőnyit balra menvén öljük meg az ott tartózkodó farkasembert, amit csak az ezüst késsel tudunk csak megtenni. Ennek következtében halála közben kiejti a vadonatúj műfogsorát, amit vegyünk fel, aztán menjünk be az előző képernyőn levő házba.

Ott egy boszorkány lakozik. Ahogy bemegyünk, meglátja nálunk a protkót és a varázsgömböt, melyet mi oda szeretnénk neki adni, és ő pedig szeretné elfogadni. Ezt tegyük meg úgy, hogy menjünk oda hozzá, és érintkezzünk vele (persze nem úgy). Ezután eltűnik készletünkéből a gömb és a protkó, de a birtokunkba kerül egy tűzoltókészülék, mely immunissá tesz minket bizonyos tűzek ellen. A csere után kiderül, hogy a farkasember a boszi egyszülött fia volt, és az anyuci megpróbál tettünk miatt kinyírni minket. Előzzük meg a bajt, és az üzlet megkötése után öljük meg a "hölgyet", hogy ne szenvedjen... Utána menjünk jobbra, majd a legközelebbi elágazásnál fel. Főnt aztán jobbra, a következő képernyőn levő álpapot löjük le, majd az általa elejtett "Ice wand" (jégfal) nevű fegyvert vegyük fel. Ezután menjünk jobbra, mindent kilőve, de a katonák által elszórt nyílpuskát még ne vegyük fel. Tovább jobbra a méheket kilőve, majd az előttünk elterülő folyót némi jéggel megszórvva és ezáltal befagyasztva még mindig jobbra, egészen addig, amíg az út véget nem ér. Itt van egy sárkány (a várban ennek az ikertestvére van), amit végezzünk ki, majd az ott levő pénzeszacskót vegyük fel. Utána menjünk balra három képernyőnyit, majd ott a jégfalat cseréljük le a nyílpuskára. Ezután menjünk balra, a következő képernyőn fel. Ott van egy cukrosbácsi, akit azonnal ölünk meg. Ekkor elejt egy kulcsot, amivel a fölöttünk levő kaput célszerű lenne kinyitni. Ezután a kulcsot

cseréljük vissza a helyette letett tárgyra. Utána menjünk fel, majd jobbra egy házig, ahová menjünk be. Ott van egy apóka, aki pénzünkért cserébe ad egy térképet az erdőben levő mocsár járható útvonaláról, amit nekünk nagyon át kell tanulmányoznunk, hogy rájövünk a járható útra. Kössünk vele üzletet (lásd: boszorkány), aminek módját még egyszer nem kívánom részletezni. A csere után ha akarjuk le is löhetjük, de nem muszáj, mert nem bánt. Menjünk ki a házból, utána balra, a csontvázakat löjük ki, majd felfelé, és lőn: elérkeztünk a mocsárhoz!

Ezen vergődjünk át. (Csak úgy mellékesen: ha a mocsárban az egyik képernyőről a másikra való átkelésnél rossz utat választunk, akkor a program lefagy, amit ezután újból be szabad tölteni...) A mocsár után menjünk jobbra, itt a tűzekeken minden baj nélkül átmehetünk. Ezután menjünk arra, amerre az út vezet (másképp nem is lehet), mindent kilőve, amíg el nem jutunk az erdőig. Ennek a kapuján menjünk be, majd bent a sárkányt löjük le. Menjünk rá a szoba közepén levő ötágú csillagra, mire a kereszt eltűnik iszonyatos zörej kíséretében, és ezzel megnyertük a játékot.

Nos, ennyit a játékról, röviden, dióhéjban. A program még jó is lenne, ha volna zenéje, nem lenne unalmas és a mocsárnál nem fagyana be. Még az a kevés hageffekt is ocsmány, ami van benne. De azért mégis érdemes megfajteni egyszer (többször nem ajánlott), de azután a lehető leggyorsabban meg kell tőle válni. A grafika viszont elég jó, a színek szépek, és jól meg vannak válogatva. Ez az egyetlen pozitívum az egész programban. Szerintem azért egyszer ki lehet próbálni. A megfajtest igyekezzem érthetően írni, de aki mégsem ért valamit, nyugodtan írhat az újság címére.

Lola

THE CURSE OF SHERWOOD	
Grafika:	7
Zene/FX:	1
Játszhatóság:	5
Összhatás:	4

A szfv az élet motorja . . .



SZIV



1992. április 15 — 16 — 17. naponta 10 — 18 óráig
Megyei Művelődési Központ (Székesfehérvár, Liszt F. u. 1.)

Székesfehérvári Informatikai Vásár

Számítógépek, nyomtatók, másolók, irodatechnikai eszközök, telefonok, telefaxok, üzenetrögzítők, telefon alközpontok, biztonságtechnikai eszközök, szoftverbörze, hálózati adatbank, egyéb informatikai eszközök kiállítása és vására.

Kiállítók:

Aktív Rekord Bt., Alku Kft., AR-Net Bt., ITV Albatech Kft., COLAST Kft., Comenius Társaság, Glória Kft., Inforing Kft., INFOSYS Alapítvány, K+F Kft., MÁTRIX Kft., MERKANT Kft., Modem Kft., OPTIMUM-JUNIOR Kft., Orfanik Kft., SO-TECH Kft., Számtex Bt., Tel-Vill GM., WESTEL Rádiótelefon Kft., . . .

(A kiállításon az újság szerkesztőivel is találkozhat!)

Ön is megnyerheti a színes televíziót, ha ellátogat hozzánk!

Az ENTERPRESS Alapítvány Alapító Okirata

- 1./ Az Alapítvány neve: **ENTERPRESS Alapítvány**
- 2./ Az Alapítvány székhelye: **Székesfehérvár, Zichy liget 10.**
- 3./ Az alapítvány létrehozója: a **MATRIX Számítástechnikai Kereskedelmi és Szolgáltató Kft.**
Képviselője a mindenkorli ügyvezető.
- 4./ Az Alapítvány célja: Az ENTERPRISE személyi számítógép felhasználók magyarországi kiadványának az **ENTERPRESS c. újság kiadásának** személyi és tárgyi feltételeinek megteremtése és folyamatos **fejlesztése.**
- 5./ Az Alapítvány **vagyon**a: Induló vagyon: A **MÁTRIX Számítástechnikai Kereskedelmi és Szolgáltató Kft.** által az Alapítványnak átadott, **10.000.-Ft azaz tízezer forint készpénz.**
- 5/1. **Az Alapítvány nyitott.** Hazai és külföldi magán- és jogi személyek, társadalmi szervezetek és ezek társaságai az alapítványi cél megvalósítása érdekében csatlakozhatnak, az Alapítványt anyagi, vagy bármilyen más eszközzel akár rendszeresen is támogatják. A feltételeken adott támogatás elfogadásáról a Kuratórium dönt.
- 6./ Az Alapítvány számlavezetője: Postabank és Takarékpénztár RT Székesfehérvári Fiókja.
- 7./ Az Alapítvány számlaszáma:
Postabank 299-98922/042-01966/4004
- 8./ **Az alapítvány jogi személy.** Az alapítók az alapítványi célok megvalósítása érdekében 3 fős Kuratóriumot hoznak létre, amely az alapítvány képviselőjét is ellátja.
Tagjai:
1. **Ujlaki László** elnök.
2. **Trattnerné Tőke Zsuzsa**
3. **Juhász István**
- 8/1. A Kuratóriumi tagok megbízatása halálozás, lemondás, közügyektől történő jogerős eltiltás, a kuratóriumi tagsági jogokkal járó kötelezettségek elmulasztása miatt szűnik meg. Megüresedés esetén a kuratóriumi hely betöltéséről a kuratórium dönt.

8/2. A Kuratórium feladatai:

- Képviseli az Alapítványt illetve az Alapítvány képviselőjére megbízást ad.
 - Felelős és elszámolási kötelezettséggel tartozik az alapítványnak, az általa kezelt alapítványi vagyonért.
 - Az alapítványnak évente beszámol az általa végzett tevékenységről.
 - Dönt az Alapítványhoz történő csatlakozások feltételeinek elfogadásáról.
 - Meghatározza ügyrendjét.
 - A kuratóriumi ügyrenddel önmagának kijelöli feladatait.
- A kuratórium **évente legalább egy alkalommal ülésezik**, de szükség esetén rendkívüli ülés hívható össze. A kuratórium határozatképes, ha az ülésen tagjainak több mint fele részt vesz. A kuratórium határozatait egyszerű szótöbbséggel, nyílt szavazással hozza meg. Szavazategyenlőség esetén az elnök szavazata dönt.
- A kuratórium üléseiről jegyzőkönyvet kell vezetni.
- 9./ **A Kuratórium az Alapítvány vagyonával szabadon gazdálkodik**, azzal vállalkozási tevékenységet is folytathat, úgy, hogy az abból befolyt jövedelmet csak az Alapítvány céljaira használhatja fel.
 - 10./ **Az Alapítvány megszűnése esetén** az alapítványi vagyon az alapítóra száll vissza.
 - 11./ **Az Alapítvány érvényességéhez az Alapító Okirat bírósági nyilvántartásba vétele szükséges.** Az Alapítvány a nyilvántartásba vétellel nyeri el jogi személyiségét és kezdi meg működését.

Székesfehérvár, 1992. március 30.

Aláírók:

*Ujlaki László
Trattnerné Tőke Zsuzsa
Juhász István*

Postafiók 334

Tóth József csereszegtomaji olvasónk először is elnézést kér, hiszen, mint írja, tulajdonképpen nem is nekünk kellene elneveznie a levelét: "Hogy mégis írok, annak az az oka, hogy van egy ENTERPRISE gépem, és szeretném egy lemezzel együtt bővíteni. Sajnos, erről a gépről kideríteni valamilyen, főleg ha tartozékokról van szó, szinte lehetetlen. Az egyik Mikrovilágban megtaláltam az Enterprise képviselőt című - gondoltam, ők a legilletékesebbek -, és felhívtam őket, hogy némi tájékoztatást adjanak. Szépen alakultak a dolgok, mint reméltem, mert kiderült, hogy ők forgalmazták a lemezzel együtt. Meg is egyeztünk egy 3,5 hüvelykes 720 KB-os lemezzel együtt, 15 000 Ft-ért; készülékük ugyan nem volt, de éppen vártak szállítást. Az egész úgy január közepén történhetett. Abban egyeztünk meg, hogy ha megkapják az árut, táviratoznak, és ha még szükségem lesz rá, utánvétellel elküldik.

Január 27-én meg is jött a távirat. Nagyon megörültünk, másnap reggel már telefonáltam is, de sajnos, csak az üzenetrögzítővel tudtam beszélni. Azóta úgy tűnik, hogy az egész egy rossz álom. Az üzenetrögzítőn hagyott üzenetre senki sem reagál. Az újságban megadott címről a levelem "címet ismeretlen" jellel visszajött. Hát ez van: se cím, se telefon, se lemezzel együtt. Nagyon kérjük Önöket, ha tudnak, segítsenek!"

Eddig a panaszos levél. Mit is lehetne mondani? Szegény Tóth József bizony nagy trágnyadombra bukkant. Ezt már csak így, pusztán levelet olvasva is meg lehetne állapítani (egy nálunk kevésbé vadkeleti ország kereskedelmi kamarája, cégbírósága talán le is vonná a megfelelő következtetéseket). Olvasónk által ismert probléma számunkra nem ismeretlen.

1990. augusztus 1-én jegyeztettük be a lapot, akkor az első szám már nyomdakész volt, amiből a legelső példányt a müncheni ENTERPRISE GmbH. vezetője kapta. Kb. egy héttel rá fax érkezett egy bizonyos VTGe Kft-től - akkor nagy-nagy örömmünkre -; kiderült, hogy az ENTERPRISE Computers magyarországi képviselője. Rövid időn belül sikerült találkoznunk a képviselő vezetőjével, *Kopácsy Vilmos*sal

(az ő jellemét és cégének valódi üzletpolitikáját akkor még nem ismertük), s mi el voltunk ragadtatva ettől a tempótól. A megbeszélésen sok mindenre ígéretet tett a képviselő, kaptunk címet, telefonszámot, rövid határidőket, aztán elbúcsúztunk.

Teltek a hetek, s nem történt semmi. Sebaj! - gondoltuk, biztosan nagyon elfoglaltak, keressük mi őket! Kerestük is, levélben, telefonon (helyesebben üzenetrögzítőn). De semmi, a találkozó pedig szükséges lett volna, hiszen a húszezer géptulajdonost jó érzéssel tölthette volna el, ha az újság révén értesülnek a képviselőről (karácsonyi ajándékként). Mivel nem kaptunk semmiféle visszajelzést, háromszor a következő tettük: javasoltunk egy találkozási időpontot (két héttel előre az üzenetrögzítőre), és pontosan megjelentünk. De csak mi, szerkesztők...

Aztán végre-alahára sikerült: a képviselő vezetője fél év eltelte után (1991. február hó) fogadott bennünket! Újabb ígéretek, újabb egyezkedések, újabb remények, újabb várakozás. (Kaptunk egy Lisp, egy Forth és egy Exdos leírást, no meg rengeteg ENTERPRISE-VTGe prospektust, amiket azóta elhasználtunk.)

Azóta sem történt semmi, több, mint egy éve! Mi persze próbálkoztunk, de 1991 végén úgy döntöttünk, hogy komolytalan *vállalkozásokkal* nem foglalkozunk.

A történetek után megvan (meglehető!) a véleményünk a VTGe-ről, a mögöttes lévő emberekről, s mindarról, ami 1987 óta történik az ENTERPRISE felhasználókkal, és ez csak a trágnyadomb csúcsa. (Nem vagyunk elfogultak: a VTGe-t dicsérem, elégedett hangú levelet még sohasem kaptunk.)

Be kell ismernünk, tisztelt Tóth József, nem tudunk tanácsot adni. Ha úgy dönt, hogy folytatja a VTGe felkutatását és sikerrel jár, kíváncsiak leszünk: hogyan csinálta?

*Ujlaki László,
Hajnal Csaba*

Az eddig beérkezett készámú pályamunka miatt folytatódik a . . .

PROGRAMOZÁSI VERSENY!!!

Szerkesztőségünk programozási versenyt hirdet az alábbi két kategóriában:

1., DEMO

Olyan programokat várunk, amelyek
- minél jobban kihasználják a Nick és a Dave lehetőségeit,
- az Enterpress-t propagálják: szerepel az újság neve, a postacímünk (1399 Budapest, Pf. 701/334), a kiadó neve (Mátrix KFT.), címe (8000 Székesfehérvár, Honvéd utca 8.) és telefonszáma ((06-22) 29-888
- szellemes (magyar illetve angol nyelvű) mondatokat úsztatnak.

Demo programot egyénileg és csoportosan is lehet készíteni. A programnyelvet mindenki maga választhatja meg, és olyan segítséget használ, amelyet csak akar. A lényeg tehát a minél látványosabb program!

2., BASIC JÁTÉKPROGRAM

Olyan Basic nyelvű játékprogramokat várunk, amelyek
- a "PROGRAM 0"-ba beférnek,,
- minél gyorsabbak,

- alapötletük, történetük (lehetőleg) újszerű.

A programot egyének és csoportok is készíthetik. Nem nevezési feltétel, de nem baj, ha a program Zzzip-pel fordítható. A játékprogram fajtája bármilyen lehet: kalandjáték, logikai játék stb. Azok a programok számíthatnak sikerre, amelyek grafikát és zenét is használnak. A program kritikus részein gépi kódú rutinokat (CODE sorok, ROM hívás) is szabad használni, ezt azonban nem szabad túlzásba vinni! A legjobb programot közölni fogjuk.

Az új beküldési határidő: 1992. július 1.

A Mátrix KFT. szerkesztőségünkkel együtt kategóriánként az alábbi díjakkal jutalmazza az első három helyezettet:

- 1. díj: 3000 Ft
- 2. díj: 2000 Ft
- 3. díj: 1000 Ft

A programokat postacímünkre küldjék:
ENTERPRESS, 1399 Budapest, Pf. 701/334.

mikrovilág

Az ENTERPRESS előző számai korlátozott példányszámban még megrendelhetők a kiadó címén (MÁTRIX Kft. 8000 Székesfehérvár, Honvéd u. 8.), vagy megvásárolhatók a Műszaki Könyvruházban (Bp. VI. ker. Liszt F. tér 9.) és a Fókusz Könyvruházban (Bp. VII. ker. Rákóczi út 14.).

Tisztelt Olvasóink!

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

A szerkesztőség

A NEPPEREKNEK NEM TALÁLHATÓ!

Az előző számban meghirdetett AKCIÓ határideje a nagy érdeklődés miatt, Május 31!

Azonban a programcsomag ára május 15-től 280,-Ft helyett, 405,-Ft!

NE FELEDJE, EGY IBM!

Ha nem akar rendelni, akkor is KÉRJEN TÁJÉKOZTATÓT EGYÉB SZOLGÁLTATÁSAINKRÓL! HIGGYE EL MEGÉRI!

EREDMÉNY: a III/4. ENTERPRESSBEN!

APOLLO Studió, Bóta István,
Tiszafüred Pf. 87. 5350

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 hirdeteit 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, Ányos u. 1/3. Tel.: (80) 29-493
Ifj. Márffy Béla, 4400 Nyíregyháza, Új u. 28.
Tóth István, 7305 Mecsekpölöske, Szegfű u. 1.

KLUB

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

*** ENTERPRESS Kéthavilag az Enterprise számítógépek felhasználóinak * II. évfolyam 2. szám * 1992. március-április *** Kiadja a MÁTRIX Kft., Székesfehérvár * Felelős kiadó: Juhász István György * A kiadó címe: MÁTRIX Kft., 8000 Székesfehérvár, Zichy liget 10. Telefon:(22) 29-888 Telefax:(22) 11-565 *** Felelős szerkesztő: Ujjkai László * A szerkesztőség tagjai: Hajnal Csaba Ízszerkesztő, Devlisoft, JOM, Art Bándor, Bozal 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 0906-1820 *** Terjeszti a Magyar Posta * Előfizethető a HELIÚ, 1900 Budapest, vagy a MÁTRIX Kft. címén * Előfizetési díj egy évre 294 Ft, két évre 147 Ft. * ** Következő számunk májusban jelenik meg. *** Az ENTERPRESS-ben közreadott információk célja az, hogy segítsék, tudnivalókat lássák el a gép felhasználóit. A közölt programokat, kapcsolási rajzokat, leírásokat mindenki szabadon felhasználhatja, de tilos azokat a kiadó írásbeli engedélye nélkül másolni, terjeszteni. * A szerkesztőség kizárólag nem őrzi meg, és nem küld vissza; továbbá külön levelezésre nincs módja. *** ENTERPRESS © 1992 MÁTRIX Kft. ***