

ENTERPRISE

KÉTHAVILAP AZ ENTERPRISE SZÁMÍTÓGÉPEK FELHASZNÁLÓINAK

Jó hír!

A szerkesztőség és a kiadó nagy örömmel tudatja a Tisztelt Olvasókkal, hogy az *Enterprise* az 1992-es évben a megszokott módon kéthavonta, 16 oldalon továbbra is megjelenik.

Szeretnénk, ha a következő évben is legalább ilyen formában megjelenhetne a lap. Az újság népszerűségében olvasóink közvetlen módon segíthetnek nekünk. Kérünk mindenkit, hogy a lap közepén található *Enterprise* plakátot ragasszák ki a lakóhelyükön olyan engedélyezett helyre, ahol azt sok ember láthatja.

A szerkesztőség és a kiadó

A külső is számít

Nem kell megjedni, nem arról lesz szó, hogy farmernadrág és pólóring helyett szmokingban járjunk! Viszont nem árt, ha összefoglaljuk azokat a tudnivalókat, amelyek a programok megjelenési formájával kapcsolatosak.

1976-ban született meg a "home computer" kifejezés az Altair kitzszámítógép (zacskóban lehetett kapni) megjelenésekor. Az akkori időkben a maroknyi felhasználó számára ropant fontos volt, hogy minél több programhoz juthassanak. A programozók pedig igyekeztek ezt teljesíteni, erejüket azonban a programok "belsejének" megírásánál összpontosították. Mit jelentett ez? Ha például valaki egy szövegszerkesztő megírásán fáradozott, akkor leginkább azokkal az algoritmusokkal, rutinokkal foglalkozott, amelyek a felhasználó számára nem láthatóak (memóriamozgatás, háttértárak kezelése, memóriaallokálás; a bevitt szöveg darabjainak törlése, beszúrása, másolása stb.) Csak ezután következett a program "külsőjének" elkészítése. Pedig a felhasználó számára ez ugyanolyan fontos, mint a program igazi lényege: egy, a magjában jó program rossz megjelenésben, nehéz, logikátlan kezelésben hasznavehetetlen. Olyan ez, mintha valaki készítené egy motorikusan igen jó autót, de karosszériát, üléseket már nem adna hozzá.

Sajnos ránk, *Enterprise* felhasználókra, mind a mai napig jellemző a programéhség, a programokra pedig az igénytelen megjelenés. Szeretnénk, ha gyorsan megszűnne ez az irányzat, és inkább a most ismertetendő elveket használnák majd a programozók programjaikban.

Mindenki számára világos, hogy a programoknak menüvezérléssel kell működniük. Menü alatt ne azt a primitív módszert értsük, amelynek egymás alá sorokba kirjuk a menüpontokat, mindegyik elé egy számot teszünk, és ezt a számot kérjük be (az input paranccsal...) választásként.

Az igazán szép megoldás az, amikor a menüpontokat egy külön kis ablakban jelenítjük meg, az ablak fejlécében feltüntetve a menüpontcsoport nevét (az ábrán a GENERAL felirat). Az ablakkezelő rutint úgy kell elkészíteni, hogy az azt a képernyőrészt, amelyre rátelepszik elmentse, és amikor bezárjuk az ablakot, akkor az eredetileg ott álló képernyőtartalom jelenjen meg ismét. Mindez csak a videomemória közvetlen írásával igazán gyors! Az ablak ak-

kor mutató, ha szép, rámás kerete van (az ábrán duplakeret), és ha a belsejében még az egyes menücsoportokat is el tudjuk választani (az ábrán szimpla vonalak). Ha még ez sem elég, akkor az ablak még árnyékot is vehet a letakart részre (az ábrán sáfrányzott rész), így háromdimenziós hatást érhetünk el. Az pedig már tényleg luxus, hogy az ablak különféle effektekkel jelenik meg, és tűnik el.

Az egyes menüpontok között a botkormánnyal (külsővel, belsővel egyaránt) mozoghatunk, és természetesen lehetőség van egyetlen

mozdulattal a menü elejére-végére ugorni (pl. shift+joy_le, shift+joy_fel). Azt a lehetőséget is biztosítanunk kell, hogy ha például a felhasználó a legutolsó sorban van, és lefelé akar mozogni, akkor a kiválasztócsík a legelső sorra térjen át (wrap).

Ha még azt is belevisszük, hogy az egyes menüsorokban kiemelünk egy-egy betűt (az ábrán ilyen pl. New, Open stb. kövérített betű), és egyszerűen csak a betű lenyomásával kiválaszthatja a felhasználó a kívánt sort, akkor már kezdetünk elégedettek lenni. Az éppen "megcélzott" menüpontnak értelemszerűen eltérő színűnek kell lennie a többi ponttól.

Ha a felhasználónak több információra lenne szüksége az adott menüpontból, akkor erről segítséget is kérhet. Ezt úgy éri el, hogy rááll a menüpontra, majd pedig leüti az F1-et. (A számítógépek világában ez az elterjedt segélykérő-billentyű.) Ekkor egy újabb ablak jelenik meg, amelyben a segítséget adó szöveg olvasható.

A kiválasztás végrehajtásához az Enter-t kell leütni. Ekkor vagy eltűnik az ablak, vagy újabb jelenik meg egy másik menünek. Ha több ablak van egymáson, akkor érvényesül igazán az árnyékos, háromdimenziós megjelenés.

A felhasználónak nagy segítséget jelent, ha egy-egy gyakran használt funkcióhoz ún. hot-key-t rendelünk. Így nem kell mindig lehfnia azt a menüt, amelyikben ez a menüpont szerepel, hanem egyszerűen csak leüti a billentyűkombinációt (az ábrán ilyen a Save CTRL-S).

Könnyen előfordulhat, hogy semmit sem kívánunk kiválasztani. A programot erről az Esc (Escape-szökés, menekülés) lenyomásával értesítjük (az Esc-nek is ez az egyik (egyetlen) felhasználása). Ekkor bezáródik az ablak.

Ez előző sorok tulajdonképpen almenükre igazak. A főmenüt úgy szokás kialakítani, hogy az a képernyő tetején legyen, a menüpontok pedig egymás mellett legyenek. A fenti megállapítások itt is igazak, néhány kiegészítéssel: a főmenühöz szokás egy külön billentyűt kapcsolni (általában az F10-et), amellyel aktivizálható. (Ha tesszem azt a felhasználó egy szövegszerkesztővel dolgozik, tehát éppen szöveget ír, de valamilyen funkciót el akar érni, akkor megnyomja az F10-et.) Az aktívra vált főmenüben jobbra-balra, előre-hátra mozoghatunk, és itt is van wrap, bár itt teljesebben. Ha ugyanis a főmenüből megnyitunk egy almenüt, és ha ekkor oldalra mozgatjuk a botkormánnyal, akkor a főmenüből kiválasztott menüpont mellett álló másik menüpontokhoz tartozó almenü gördül le.

A főmenü egy adott pontjának elérése úgy is megoldható, hogy valamilyen billentyűkombinációval érjük el. (Az ábrán látható GENERAL almenüt, amely közvetlenül a főmenüből nyílik le, mondjuk a CTRL-G-t.)

(Ábra a 15. oldalon található)

(folyt. köv.)

TARTALOM 92/1

KURZUS

Assembly 9.	2-3
A Pascal 9.	4-5
A közvetlen lemezkezelés rejtelvei 1.	6

PROGRAMOZÁSTECHNIKA

Az Életjáték (3)	7
Lakat alatt	8

HARDVER

Hat férőhelyes EPROM-bővítő kártya	9
------------------------------------	---

TESZT

EPDOS v1.3	10
------------	----

TIPPEK-TRÜKKÖK

Írás a státuszosorba	11
----------------------	----

KÖNNYED MŰFAJ

Dizzy III., VENDETTA	12-14
----------------------	-------

MINDENFÉLE

Postafiók 334	15
Hírdetések, felhívások	16

Assembly

9. rész

A nyolcadik részben elrejtett felhívás hatására valóságos levélzón érkezett hozzám, így lelkiemben megerősödve folytatam a sorozat írását. Köszönöm Tisztelt Olvasóim sorait. Igazából persze semmivel sem lettem tájékozottabb elvárásokról, nagyon kevés konkrét kérdést érkezett hozzám. Nem akarok a levelekre külön-külön válaszolni, de szeretnék néhány dolgot leszögezni: Meglehetősen nehez (képtelenség) az assemblyról úgy sorozatot írni, hogy a teljesen kezdők és a profik számára egyaránt érdekes, tartalmas legyen. Igyekezem (igyekeztem) az alapoktól elindulva ismertetni az assembly rejtelmét, és szeretnék a sprite-kezelésig, a grafikus-képernyőmeghajtásig, biztonságosan működő hibakezelésig stb. eljutni. Elárulom azt is, hogy ilyenek megírására szeretnék ígálni "szakértőket" bevonnak. Céloom tehát az, hogy azok, akik rendszeresen olvassák a sorozatot, előbb-utóbb képesek legyenek önállóan programokat, rutinokat írni.

A sorozat nyolcadik részében két program jelent meg. Az első program működése és tanulása gondolom mindenki számára világos. Az ott közölt második programhoz még hiányzik a magyarázat. A magyarázat után a grafikáról esik néhány szó.

Így lesz gyorsabb

A program eleje hasonlít az elsőre. Az első érdekesség a videolap megjelenítése (35-38). Ebben a megoldásban azt szerettem volna illusztrálni, hogy hogyan lehet helyet sprórolni: nem regiszterenként, hanem regiszterpáronként adjuk meg a szükséges paramétereket, így néhány bajtípi helyet nyerhetünk. Persze az EP nem ZX81, van helyünk bőven (?) a memóriában.

Programunk közvetlenül a videomemóriába fog dolgozni, szükségünk van a megnyitott videolap kezdőcíme. Ez azt a Nick címet jelenti, amely a lap "bal-felső" pozíciója. Az EXOS 11-et B=3-mal (@ADDR) hívva, BC-ben visszaadja a videolap kezdőcímét. Szeretném hangsúlyozni, hogy ez a cím mindig Nick-címet jelenti! A kapott értékből nekünk kell eldönteni, hogy mi a szegmensszám.

A szegmensszám meghatározásához egy ügyes trükköt használunk fel. A Nick 64KB-os videomemóriát kezel, az első szegmens a 252-es, az utolsó a 255-ös. Ha binárisan ábrázoljuk azt a Nick címtartományt, amely egy-egy szegmenshez tartozik, a felső két (b15 és b14) bit viselkedésében a következőket figyelhetjük meg (a b13-b0 biteket nem ábrázolva):

szegmens	b15	b14
252	0	0
253	0	1
254	1	0
255	1	1

A b15 és a b14 tehát egyértelműen azonosítja az illető videoszegmenst. A két bitet kétszer balra mozgatva egy regiszterben előállítjuk a 0,1,2 vagy 3 értéket, amire már csak 252-t kell hozzáadni, és megkapjuk a szegmensszámot.

Egy konkrét példa: Legyen az EXOS 11 után BC=39413. Erről a Nick címről kell tehát megtudnunk, hogy milyen szegmensre esik. Határozzuk meg a high bajt értékét. Mivel $39413 = 245 + 153 \cdot 256$, ezért a high bajt 153-mal egyenlő.

A 153 binárisan: 10011001. Számunkra csak a legfelső két bit érdekes: 10b. Ha az alsóbb 6 bitet nullázzuk ("összeeszeljük", azaz maszkoljuk az eredményt 11000000b-sal), akkor az előbbi bináris értékből csak ez marad: 10000000b. Most ezt az értéket kétszer balra rotáljuk, így az 00000010b (decimális 2) lesz. Ha most ehhez 252-t (a "báziszegmens" értékét) adunk, akkor megkapjuk a címhez tartozó szegmensszámot, ami $252 + 2 = 254$ lesz. (Tényleg jól okoskodtunk, hiszen a 39413 nagyobb, mint 32768 és kisebb, mint 49152, tehát a Nick 254-es szegmensén van.)

HL-be töltjük (45-47) BC-t, ezt a szegmensszám meghatározása követi. Elsőként (50) A-ba töltjük B-tartalmát, azaz a high bajtot. A kapott tartalmat kimaszkoljuk (52) a 11000000b értékkel, így csak a felső két bit marad meg, az alsó hat nulla lesz. Ezt követi a kétszeres balra rotálás (53-54). A kapott eredményhez nem hozzáadjuk a 252-t, hanem "belevagolyjuk" (55). Így A-ba kerül a szegmensszám, amit B-be töltünk (57).

A következő feladat, hogy a szegmenst a Z80-ra lapozzuk. Első kérdés, hogy melyikre? A program a CPU P1 és P2 lapjára. De mit keres itt két szegmensnek megfelelő tartomány? Képzeliük el, hogy az EXOS Nick címet mondjuk 16383-at ad vissza. Ez a 252-es szegmenst jelenti, de azon csak egy bajtot. Az általunk megnyitott videolap további része már a következő 253-as szegmens lesz, tehát a rutin megbízhatósága érdekében a következő szegmenst is belapozzuk majd. Ezzel a megoldással nem kell ellenőriznünk azt, hogy a kapott szegmens ténylegesen hány bajt érhető el a videolap által meghatározott mérethez képest. Felmerülhet újabb kérdésként, hogy mi van akkor, ha

a 255-öt kapjuk szegmensszámként? Semmi különös. P1-re lapozzuk a 255-ös szegmenst, P2-re pedig a 0-ás (egyik EXOS) szegmenst.

A következő feladat, hogy a Z80-ra lapozott videoszegmenst valamilyen Z80-as címen elérhessük (Nick-Z80 címtranszformáció), ennek a címnek a kiszámításához a következő újabb trükköt használjuk: A 16 bites bináris aritmetika szabályai szerint $65535 + 1 = 0$, $30321 + 45763 = 10549$ stb. Ez számunkra azt jelenti, hogy ha egy adott Nick címhez egy kellő öszihez hozzáadunk, akkor az egy Z80 címet jelent. Az érthetőségért számítsuk ki, hogy a 39413-as Nick címet hogyan kell átalakítani Z80-as P1 tartományba eső címmé. (A 252-es szegmenst a Z80 P1-re lapozzuk.) A 39413-as Nick cím a $32768 - 49151$ Nick tartományba esik (ezt akár "Nick-P2"-nek is nevezhetnénk), ezt kell átalakítanunk a Z80 P1-es lapjához, P1-hez a $16384 - 32767$ Z80 tartományba esik. Ha egy kicsit gondolkodunk, rájövünk, hogy nyilván $49152 - 1$ kell ehhez a Nick címhez hozzáadnunk, hiszen 3 szegmensnyi tartományt kell átugranunk. (Megoldható persze ugyanez kivonással is, de véleményem szerint így olvashatóbb a lista.) Így: $39413 + 49152 = 23030$, tehát ha a a 252-es videoszegmenst a Z80 P1 lapjára lapozzuk, akkor az adott Nick cím a 23030-as Z80 címet jelenti.

Ezeket a címtranszformációkat külön rész végzi (60-74). A-ban a szegmensszám van. Először ellenőrizzük, hogy egyenlő-e 255-el? Ha nem (61), akkor ráugrunk a 254-es, 253-as, 252-es szegmensre jelentő sorokra. Ha egyenlő 255-el (62), akkor DE-be töltjük az ofsztérteket, majd "rácorgunk" a további részekre, de azok nem módosítják DE értéket. (Javasolom mindenkinek, hogy tekintse jól át ennek a pár sornak (60-74) a logikáját! Gondolják át úgy, hogy agyuk A regiszterébe sorra a 255, 254, 253, 252 értékeket töltsék.)

Ha megvan az ofsztértek, akkor már nincs más dolgunk, minthogy a HL-ben levő Nick címhez hozzáadjuk azt, így végül HL-ben előáll a várva várt Z80-as cím.

A nehezen túljutottunk. Be kell lapoznunk a videoszegmenseket, de előbb őr ember módjára elmentjük a pillanatnyi szegmenskiosztást P1-en, P2-n (77-84). Gondolom ezt nem szükséges részletezni.

Most jön a lényeg, a megjelenítő rész (87-99). (Itt jól látható a programozási munka egyik sajátossága: sok-sok fontos körítés után és előtt egy pár bajtos rutinrészlet jelenti az eredményt.) Az A regiszter kettős életet él a rutinban: egyrészt a megjelenítendő karakter kódját hordozza, másrészt a cycl 0 ciklust tartja kezében. Először (87) az akkumulátorba nullát töltünk (XOR A)? Hogy is lehet ez? Tessék rájőnni! Ezután HL tartalmát (Z80-as lapkezdőcím) a veremre tesszük (88), ez egyben a legkülső (cycl 0) ciklus első utasítása.

D-be az Y-méretet (89), E-be az X-méretet (90) töltjük, ez utóbbi a "középső" ciklus (cycl 1) belépési pontja. Tulajdonképpen E darab karaktert teszünk ki D-szer.

A belső ciklus (cycl 2, 91) első utasítása letölti a karakter kódját a HL által címzett címre. Ezután növeljük HL-t (92), így az már a következő pozíciót címzi. Csökkenteni kell a ciklusszámlálót (93). Ha még nem érte el a nullát (94), akkor folytatódik a karakterek kirakása.

Ha a belső ciklus lefutott, akkor a "középső" ciklus számlálójának értékét csökkenti (95). Nem nulla érték esetén visszamegy cycl 1-hez, feltölti E-t, majd kirak egy újabb sort. Természetesen kezében HL tartalmának folyamatosan kell növekednie. Ha a cycl 1 is lefutott, akkor egy adott karakterrel feltöltöttük a teljes képernyőt. Előlről kell kezdeni, így a kezdőcímet felvesszük a veremről HL-be (97), majd növeljük a karakterkódot, azaz a külső ciklus számlálóját. Ha ez nem érte el a nullát, akkor a külső ciklus újra lefut, az összes belső ciklussal együtt. Ha az akkumulátor eléri a nullát, akkor vége a ciklusoknak, így a megjelenítés végrehajtásának.

Már csak rendet kell raknunk magunk után (102-105): visszaállítjuk az eredeti szegmenskiosztást, majd pedig lezárjuk a videocsatornát, és kész.

Fordítsuk le és indítsuk el a programot. *Apropó! Egyik olvasóm írta, hogy a sorozatban közölt rutinokat nem tudja Basic-ből futtatni. Nos, ezeket a programokat az ASMON-ból kell elindítani, azt hittem, hogy ez magától értődik. Később szeretnék majd az "ánütetésről" is beszélni.*

Ha elindult a program, akkor az elsőnél lényegesen nagyobb sebességgel jeleníti meg a teljes karakterkészletet.

Az első grafika

Az 1. listán látható program jelenti első grafikus próbálkozásunkat. Az előző példára hivatkozva szeretném felhívni a figyelmet arra, hogy ott igazi, hardver-szöveges üzemmódban ("TEXT 40"), karakteres képernyőn jelentek meg a karakterek. Ennek kezelése rendkívül egyszerű, hiszen a videomemória egy bajtja egyértelműen meghatároz egy karakter. Sajnos nincs ez így a szoftver-szöveges képernyőn ("TEXT 80"). Ennek az a lényege, hogy ebben a módban egy EXOS rutinocská jelenít meg grafikus képernyőn a karaktereket (egy karakter egy 8*9-es pontmátrixban jelenik meg). Szöveges képernyőnek azért nevezhetjük, mert editorcsatornát rendelhetünk hozzá. Bizonyára mindenki megfigyelte már, hogy 80 karakteres képernyőre szinte nincs is alkalmazás (kivéve az IS-DOS-t). Ennek pontosan a képernyőtípus sajátossága az

oka: ha szöveges képernyőnek szeretnénk használni, akkor nem tudjuk gyorsan meghajtani, mert úgy kellene "megprogramozni", mintha grafikus képernyő lenne. Ezért aztán a programozók vagy a hardver-szöveges, vagy pedig igazi grafikus képernyőt választanak.

A listán olyan programot láthatunk, amely megnyit egy nagyfelbontású, kétszínű grafikus videolapot, amelyre felrajzol egy kifestett kört, majd ezt a kört 200 ponttal balra scrollozza. Sok dolgot kell ezzel kapcsolatban elmondanom, ezért ennek értelmezése a következő részre marad, addig gondolkozzon mindenki a működésén (home work). Néhány dolgot el kell árulnom, az Asmon hibájából: a programot csak úgy lehet használni, ha 5-ös fejléccel programmá fordítjuk, és új alkalmazói programként futtatjuk.

(folytatjuk...)
-HCs-

```

1      org 0100h
2
3  evar  macro @exvar,@val
4      ld bc,256+@exvar
5      ld d,@val
6      exos 16
7      endm
8
9  x_size equ 40
10 y_size equ 27
11 vchan equ 10
12
13      di
14      ld sp,stack
15      ei
16
17      evar 22,1
18      evar 23,0
19      evar 24,x_size
20      evar 25,y_size
21
22      ld a,vchan
23      ld de,vid
24      exos 1
25
26      ld a,vchan
27      ld bc,seqlen
28      ld de,seq
29      exos 8
30
31      ld a,vchan
32      ld bc,256+1
33      ld de,y_size*256+1
34      exos 11
35
36      ld a,vchan
37      ld b,3
38      exos 11
39
40      push bc
41      pop hl
42
43      ld a,b
44      and 11000000b
45      rlc a
46      rlc a
47      or 1111100b
48      ld b,a          ;B-ben a szegm.
49
50      cp 255
51      jr nz,seg254
52      ld de,32768
53  seg254 cp 254
54          jr nz,seg253
55          ld de,49152
56  seg253 cp 253
57          jr nz,seg252
58          ld de,0
59  seg252 cp 252
60          jr nz,sepage
61          ld de,16384

```

```

62  sepage add hl,de          ;HL-ben P2-es cím
63
64          ld de,x_size*2+y_size*9-1
65          add hl,de
66
67          in a,(0B3h)
68          push af
69          in a,(0B2h)
70          push af
71          in a,(0B1h)          ;mentes
72          push af
73          ld a,b          ;B-ben a szegm.volt
74          out (0B1h),a
75          inc a
76          out (0B2h),a
77          inc a
78          out (0B3h),a
79
80          ld b,200
81  cycl_0  push bc
82          push hl
83
84          ld b,y_size*9
85  cycl_1  or a
86          ld c,x_size*2
87  cycl_2  rl (hl)
88          dec hl
89          dec c
90          jr nz,cycl_2
91          call c,r_side
92  point  djnz cycl_1
93          jr cend
94
95  r_side  ld de,x_size*2
96          add hl,de
97          set 0,(hl)
98          or a
99          sbc hl,de
100         ret
101
102  cend    pop hl
103         pop bc
104         djnz cycl_0
105
106         pop af
107         out (0B1h),a
108         pop af
109         out (0B2h),a
110         pop af
111         out (0B3h),a
112
113  loop   jr loop
114
115  vid    db 6
116         defm "VIDEO:"
117  seq    db 27,"C",255,32,32,255,0,0,0,0
118         db 27,"A"
119         dw 640,486
120         db 27,"E"
121         dw 480,480
122         db 27,"F"
123  seqlen equ $-seq
124
125  stack  defs 64
126         defb 0
127         end

```

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

9. rész

A beépített függvények és eljárások

A függvények egy része a bemenő paraméterével azonos típusú eredményt ad, sok függvény különböző típusú paramétereket is elfogad. Más függvények típusa eltér a bemenő paraméter típusától; sőt, egyes függvények célja kimondottan a konverzió egyik típusból a másikba.

Ez megnehezíti a tárgyalásukat, osztályozásukat. Megpróbáljuk a lehető leglogikusabb csoportosítást.

Jelölések

A függvények és (eljárások) paramétere általában konstans, változó vagy kifejezés lehet. Egyes esetekben a paraméter csak változó lehet, ezt mindig jelezzük. A paramétert helyettesítő kisbetű jelzi a paraméter típusát:

- i : egész (integer)
- r : valós (real)
- c : karakter (char)
- s : string
- b : logikai (boolean)
- f : file (vagy text)
- x : a paraméter többféle típusú lehet.

Egész eredményt szolgáltató függvények:

ABS(i) : az i egész paraméter abszolút értéke
 SQR(i) : az i egész paraméter négyzete (nem négyzetgyöke!)
 TRUNC(r) : az r valós paraméter egész része, azaz a törtrész elhagyásával keletkező egész szám
 ROUND(r) : az r valós paraméter kerekítésével keletkező egész szám (0-tól 0.5-ig lefelé, onnantól felfelé kerekít)
 PI:
 ABS(1) eredménye 1; ABS(-2) eredménye 2
 SQR(3) eredménye 9
 TRUNC(3.9) eredménye 3; TRUNC(-3.9) eredménye -3
 ROUND(3.9) eredménye 4; ROUND(-3.9) eredménye -4

E két függvény jelentőségét az adja, hogy míg minden egész számnak megfeleltethető egy vele azonos matematikai értéket képviselő valós szám, addig ugyanez visszafelé már nem igaz: a valós számok csak kis töredékének felel meg valamilyen egész szám, még akkor is, ha a valós érték egyébként az ábrázolható egész számok tartományán belül van. Ez azt jelenti, hogy egy egész kifejezést mindig értékül adhatunk egy valós változónak (az ehhez szükséges típuskonverziót a fordítóprogram önmagától előállítja). Viszont a fordított esetben, tehát ha egy valós kifejezést akarunk értékül adni egész változónak, abban az esetben, ha egyáltalán az értékadás végrehajtható, minden esetben meg kell mondanunk, hogy milyen konverziót kívánunk végrehajtani: a kifejezés egész (vagy csonkított) értékére, vagy pedig kerekített értékére van-e szükségünk. Tipikus példa lehet egy rajzoló program, ahol a görbe pontjainak koordinátáit nagy valószínűséggel valós értékekkel számoljuk, a rajzolásához viszont egész képpont-koordinátákat kell megadni. Ha itt csonkított értéket használunk, a görbe jobban torzul, mint ha kerekítést alkalmaznánk.

Egész típusú paraméterrel is használható a következő két függvény, amelynek bemenő paramétere bármilyen sorszámozott (azaz egész, résztartomány vagy felsorolási, vagy karakter) is lehet:

SUCC(x) eredménye a sorozat következő eleme
 PRED(x) eredménye a sorozat előző eleme

PI:
 SUCC(3) eredménye 4; PRED(3) eredménye 2
 SUCC('A') eredménye 'B'

és, feltételezve a

TYPE

NAP = (hétfő, kedd, szerda, csütörtök, péntek, szombat, vasárnap)

deklarációt,
 SUCC(kedd) eredménye szerda
 PRED(kedd) eredménye hétfő
 viszont SUCC(vasárnap) és PRED(hétfő) eredménye meghatározatlan.

Tetszőleges felsorolási típusú, illetve karakteres típusú érték típuson belüli sorszámát adja vissza az ORD függvény. Karakter típusú paraméterrel így a BASIC ASC függvényével egyenértékű, de sokkal

általánosabban használható.

ASCII kódkészletet feltételezve,
 ORD('A') értéke 65,
 ORD('a') értéke 97,
 és az előző deklaráció érvényességén belül
 ORD(hétfő) értéke 0,
 ORD(vasárnap) értéke 6.

Kizárólag karakter típusú eredmény esetén az ORD függvény inverze a CHR függvény:

CHR('65') eredménye 'A', és így
 CHR(ORD('A')) eredménye 'A'.

Az ODD függvény eredménye logikai (BOOLEAN) típusú: ha egész paramétere páratlan, a függvény értéke TRUE (igaz), páros szám vagy nulla esetén pedig FALSE (hamis).

A függvények egy külön csoportját alkotják a matematikai függvények. Ezek paramétere és értéke is valós.

Már ismerjük az egész típusú ABS függvényt. Nos, az ABS(r) függvény valós paraméter esetén valós értéket szolgáltat. Ugyanez érvényes az SQR függvényre is. A valós paraméter négyzetgyökét az SQRT(r) függvény adja.

A SIN(r) függvény a valós paraméter szinuszt,
 a COS(r) függvény a valós paraméter koszinuszt,
 az ARCTAN(r) függvény a valós paraméter arkusz tangensét adja meg. Ezek a leggyakrabban használt trigonometrikus függvények; a többi értékét szükség esetén ezekből már kiszámíthatjuk.

Az EXP(r) függvény az e alapú exponenciális függvény ("e az r-edik hatványon") megfelelője.

Az LN(r) függvény ennek inverze, tehát az e alapú logaritmus függvénye.

A bevitellel és kivitellel kapcsolatos eljárások és függvények

A most következők, sajnos, nem vonatkoznak a HISOFT-PASCAL implementációra, mivel az - a SPECTRUM-tól örököve - nem tartalmaz file-kezelést. A Turbo-Pascal viszont teljes mértékben megfelel a leírásnak, sőt, további szolgáltatásokkal teszi még könnyebbé a file-kezelést.

A bevitel és kivetel tárgyalásakor röviden áttekintettük a két alapvető (és igen ritkán használt) beviteli-kiviteli eljárást, a GET(f) és a PUT(f) eljárást. Emelttük az inkább használatos READ és WRITE eljárást is. Ezek már változó számú paramétert is elfogadnak, így használhatunk igen kényelmes. Általános alakjuk:

READ(f, p1, p2 ... pn) és
 WRITE(f, p1, p2 ... pn).

A file-kezeléshez tartozik még néhány eljárás és függvény:

Az ASSIGN(f, s) eljárás az f file-hoz az s stringben megadott file-nevet rendel hozzá. A file ezután már megnyitható frásra:

REWRITE(f)
 vagy - feltéve, hogy a file létezik - olvasásra:
 RESET(f).

Az olvasásra megnyitott file-t nem kell lezárni, de az frásra megnyitott file lezárását nem szabad elmulasztani, hiszen ekkor kerül ki ténylegesen a háttértárra vagy perifériára a file utolsó darabja, illetve ekkor tródik felül a file katalógusbejegyzése a merevlemezen vagy a floppyra. A lezárást - Turbo-Pascalban - a CLOSE(f) eljárás végzi.

Olvasáskor a file végét jelzi a logikai (boolean) típusú EOF(f) függvény, ez igaz (TRUE) értéket ad, ha a következő elem már nem beolvasható, mert elértük a file végét. Ennek alapján egy file másolását a következő programválat mutatja:

```
TYPE
  ALAPTIPUS : CHAR { vagy tetszőleges más típus };
  F1, F2 : FILE OF ALAPTIPUS;
VAR
  PUFFER : ALAPTIPUS;
  FNAME1,
  FNAME2 : STRING[ 40 ];
```

```
...
FNAME1 := 'F1.DAT'; FNAME2 := 'F2.DAT'; { vagy más módon adunk nevet }
```

```
ASSIGN( F1, FNAME1 ); RESET( F1 ); { megnyitjuk olvasásra }
```

```
ASSIGN( F2, FNAME2 ); REWRITE( F2 ); { megnyitjuk frásra }
WHILE NOT EOF( F1 ) DO BEGIN
  READ( F1, PUFFER ) WRITE( F2, PUFFER );
```


END (WHILE);
CLOSE(F2);

Bár a szöveges információ a FILE OF CHAR deklaráció segítségével is kezelhető lenne, a Pascal bevezet egy teljesen új típust a szöveges file-ok feldolgozásához. Ez a típus a TEXT. Például:

TYPE
T : TEXT;

A TEXT típus nem azonos a FILE OF CHAR típusal. A különbség az, hogy a TEXT típus, amellyel, hogy természetesen karakterekből áll, nagyobb egységekre, szövegsorokra osztható. Tudjuk, hogy fizikailag a szövegsorok végét CR és LF karakterpáros jelzi. A TEXT típus használatával azonban ezt a CR-LF karakterpárost a programozó nem látja, a nyelv más módon, sokkal kényelmesebben jelzi a sorok végét.

Egyrészt a TEXT típus esetén a READ és a WRITE eljárás mellett használható a csak erre a típusra érvényes READLN és WRITELN eljárás is. Ezek teljes alakja

READLN(f, p1, p2 ... pn)
WRITELN(f, p1, p2 ... pn)

A READLN eljárás beolvassa a paraméterlistájában megadott elemeket, majd az aktuális sor végét átugorva, új sor elejére áll (azaz átugorja a következő CR-LF karakterpárost).

A WRITELN eljárás kiírja a paraméterlistájában szereplő elemeket, majd új sorra ugrik (azaz a file-ba ír egy CR-LF karakterpárost).

A sorok kezelését könnyíti az EOLN(f) függvény, ez igaz (TRUE) értéket ad, ha az olvasott file-ban éppen sor végéhez értünk.

A szövegfile tagolását segíti a PAGE(f) eljárás, amely új lapot kezd a file-ban, azaz a további kiírás új lapra kerül. Ennek elsősorban nyilván nyomtatásra szánt szöveg esetén van jelentősége. Fizikailag ilyenkor a file-ba általában egy FF (form feed) karakter kerül, ez a nyomtatónál lapdobást vált ki.

A Pascal, a többi programozási nyelvhez hasonlóan, a perifériás készülékeket - a billentyűzetet, a képernyőt, a nyomtatót stb. - szintén file-ként kezeli. A Turbo-Pascal például a következő készülékeket fogadja el:

CON: A rendszerkonzol, általában a képernyő és a billentyűzet.
KBD: A billentyűzet, csak bemeneti eszköz.
TRM: Terminál (képernyő), csak kimeneti eszköz.
LST: Valamilyen listázó eszköz, általában nyomtató.
AUX: Valamilyen külső eszköz, pl. lyukszalagolvasó vagy -lyuksz-

tó.

USR: A felhasználó által definiált eszköz.

A CON: eszköz puffertelt és echózott be- és kivittelt meg, azaz a billentyűzetet beírt karakterek megjelennek a képernyőn, és mindaddig nem kerülnek be a befogadásukra rendelt változóba vagy változóba, amíg meg nem nyomjuk az ENTER billentyűt. Ez azt jelenti, hogy addig javíthatjuk az esetleg hibásan bevitt szöveget.

A KBD: eszközről ténylegesen karakterről karakterre tudjuk az információt bevinni, és a beírt karakterek nem jelennek meg a képernyőn.

Ezeket a készülékeket ugyanúgy használhatjuk, mint bármilyen file-nevet:

ASSIGN(F1, 'TRM:'); REWRITE(F1);
ASSIGN(F2, 'KBD:'); RESET(F2);

Van azonban egy még kényelmesebb lehetőség a készülékek használatára. A Turbo-Pascal mindegyik készülékhez hozzárendel egy előre definiált - TEXT típusú - változót. Ha ezeket használjuk, nincs szükség sem az ASSIGN, sem pedig a REWRITE vagy a RESET eljárás használatára, ezek a file-ok automatikusan megnyitódnak a program indításakor. Ezek a file-azonosítók a következők:

CON, KBD, TRM, LST, AUX, USR

és értelemszerűen mindegyik a hasonló nevű készülékhez van hozzárendelve.

A szabványos Pascalban - és a Turbo-Pascalban is - létezik még két előre definiált file, az INPUT és az OUTPUT file. Mindekkettő általában a rendszerkonzolhoz, tehát a billentyűzet-képernyő együttesen van hozzárendelve. Ez a két file jelenti a rendszer szabványos bemenetét és kimenetét; ez annyira így van, hogy ezt a két file-t használva a nevek maguk elhagyhatók. Ennek értelmében

READ(X) megfelelője READ(INPUT, X)
READLN(X) megfelelője READLN(INPUT, X)
WRITE(X) megfelelője WRITE(OUTPUT, X)
WRITELN(X) megfelelője WRITELN(OUTPUT, X)

Ez rendkívül leegyszerűsíti az alapszintű be- és kivittelt.

A TEXT típus használata esetén a Pascal egy óriási segítséget ad, ez az automatikus konverzió. A READ, READLN, WRITE és WRITELN eljárás ugyanis nemcsak karaktereket tud olvasni vagy írni, hanem numerikus (egész és valós) értékeket, sőt, kiírásakor logikai (boolean) típust is használhatunk. A numerikus-karakter konverzió és elmentettje olyan természetes számunkra, hogy meg sem fordul a fejünk-

ben, hogy ez bizony valóban konverzió: a 123 egész érték kiírásakor a file-ba a '123' karaktersorozat kerül; a file-ban található '1.2' karaktersorozat beolvasásakor a READ (vagy READLN) eljárásban megadott valós változóba az 1,2 valós érték (pontosabban, egy ehhez igen közeli érték) kerül.

A WRITE és a WRITELN eljárás további lehetőséget ad ennek a konverzióknak a befolyásolásához, ez a formátumvezérlés. A Pascal formátumvezérlése rendkívül egyszerű, ugyanakkor hatékony. A kiírandó kifejezést (változót, konstant) kettősponttal elválasztva egy pozitív egész eredményt adó másik kifejezés (változó, konstans) követi. A második érték neve mezőszélesség, és nevéhez híven a kiírási mező szélességét adja meg, azaz azt, hogy a kiírás hány karakter szélességű legyen. Ez a formátumvezérlés alkalmazható a numerikus egész, a karakter, a logikai és a string értékek kiírásához, valamint a valós értékekhez is, ha megfelel a lebegőpontos (exponenciális, normálalakú) megjelenítési forma.

Ha nem adunk meg formátumvezérlő információt, a Pascal minden értéket annyi karakterrel ír ki, amennyi ehhez minimálisan szükséges. A valós értékeket ekkor lebegőpontos alakban írja ki, az értékes jegyek számát az adott gépi reprezentáció határozza meg. Az ilyen "szabodon hagyott" kiírásnak az a hibája, hogy az egyes értékek egymásba folynak, nem lehet azokat egymástól megkülönböztetni. Például a

WRITELN(1, '2', 3, TRUE)

eljáráshívás eredménye

123TRUE

ami nem túl szerencsés. Javíthatunk az eredményen így:

WRITELN(1, ' ', '2', ' ', 3, ' ', TRUE)

vagy pedig így:

WRITELN(1, '2', 3, ' ', TRUE)

mindkét megoldás eredménye

1 2 3 TRUE

A formátumvezérlést alkalmazva azonban a különböző hosszúságú eredmények is oszlopokba rendezhetők.

WRITELN(1 : 4, '2' : 4, 3 : 4, TRUE : 6);

WRITELN(11 : 4, '22' : 4, 33 : 4, FALSE : 6);

WRITELN(111 : 4, '222' : 4, 333 : 4, TRUE : 6);

hatására a három számjegy mindegyike egy-egy négykarakteres mező jobb szélére igazítva jelenik meg:

1	2	3	TRUE
11	22	33	FALSE
111	222	333	TRUE

A valós értékek kiírása, mint már említettük, exponenciális formában történik, így

WRITELN(3.14159265358 : 8)

eredménye

3.14E+00

Ha inkább fixpontos formában szeretnénk látni a valós értéket, akkor alkalmazzuk a formátumvezérlés másik alakját, amelyik egy további értéket ad meg, a mezőszélességtől ugyancsak kettősponttal elválasztva. Ez az érték a kiírandó tört jegyek számát adja meg:

WRITELN(3.14159265358 : 8 : 4)

eredménye

3.1415

Ha csak az egész jegyekre vagyunk kíváncsiak, egyszerűen 0-t adunk meg.

Ha a bármelyik érték nem fér el a számára ily módon előírt helyen, különböző Pascal implementációk eltérően viselkedhetnek. Míg a nyelv eredeti leírása szerint ilyenkor az értékből csak annyi jelenik meg, amennyi a számára előírt helyen elfér, nem törődve azzal, hogy így esetleg téves értéket kapunk, a Turbo-Pascal ilyen esetben kitor a korlátok közül, és önkényesen annyi pozíciót használ fel a kiírásra, amennyi ehhez minimálisan szükséges; azaz, mintha nem is adunk volna meg formátumspecifikációt.

(folytatjuk...)

-UL-



Tisztelt Honfitársaink!

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 Aerocartas Mentőszolgálat emberek akaratából az emberért munkálkodik.

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

A közvetlen lemezkezelés rejtelmerei

1.rész

Az LPT sorozat befejezése óta nem sok újdonságot olvashattak az igazán profi ENTERPRISE programozók. Most azonban ismét itt van egy olyan téma amelyben az ENTERPRISE legnépszerűbb lelkivilágába próbáljuk elvezetni a bátrabb olvasókat.

Azt általában minden lemezegységgel rendelkező tudja, hogy a controller-kártya agyát egy WD1770-es vagy újabb változatokban egy WD1772-es floppy disk controller chip alkotja. De ezen IC közvetlen programozását már kevesen ismerik, ezért most erről lesz szó.

A WD1770 egy 28 lábú DIP tokozású NMOS alapú integrált áramkör, amely az egy- és kétoldalas lemezegységek meghajtására egyaránt alkalmas, de csak szimpla és dupla sűrűségű felírásra képes. Tehát normál esetben maximum 800 Kbájtos lemezeket képes kezelni. A WD1772 csak annyiban különbözik a 1770-estől, hogy gyorsabb az író/olvasó fej léptetése. Ennyi bevezető után rá is térhetünk a lényegre, vagyis a programozásra.

A controllerrel négy írható/olvasható porton keresztül lehet kommunikálni, ezek sorban a következők:

Olvasásnál:

- 10h — Állapot regiszter.
- 11h — Track regiszter.
- 12h — Szektor regiszter.
- 13h — Adat regiszter.

Írásnál:

- 10h — Parancs regiszter.
- 11h — Track regiszter.
- 12h — Szektor regiszter.
- 13h — Adat regiszter.

Ezekon kívül van még egy port amely a controller-kártyához tartozik, a 18h port, ennek használatáról később lesz szó, ugyanis ez nem a WD1770-es közvetlen portja. A kiadható parancsok a következők:

7	0	6	5	4	3	2	1	0	-	bit
0	0	0	0	0	h	v	r1	r0	-	Fej alaphelyzetbe állítása.
0	0	0	1	h	v	r1	r0	-	-	Fej pozicionálása.
0	0	1	u	h	v	r1	r0	-	-	Fej léptetése.
0	1	0	u	h	v	r1	r0	-	-	Fej léptetése befelé.
0	1	1	u	h	v	r1	r0	-	-	Fej léptetése kifelé.
1	0	0	m	h	E	0	0	-	-	Szektor olvasása.
1	0	1	m	h	E	P	a	-	-	Szektor írása.
1	1	0	0	h	E	0	0	-	-	Cím olvasása.
1	1	1	0	h	E	0	0	-	-	Track olvasása.
1	1	1	1	h	E	P	0	-	-	Track írása.
1	1	0	1	I3	I2	I1	I0	-	-	Parancs megszakítás.

A táblázatban használt betűjelzések jelentése:

- h = Motor bekapcsolás jelző.
- h=0, Felpörgetési folyamat engedélyezve.
- h=1, Felpörgetési folyamat tiltva.
- v = Ellenőrzés jelzőbit.
- v=0, Nincs ellenőrzés.
- v=1, Ellenőrzés bekapcsolva.
- r1,r0 = Fejléptetési sebesség.
- r1 r0 WD1770 WD1772
- 0 0 6ms 2 ms
- 0 1 12ms 3ms
- 1 0 20ms 5ms
- 1 1 30ms 6ms
- u = Track regiszter beállítás jelzőbit.
- u=0, Nincs beállítás.
- u=1, Fejléptetés után beállítja a track-regisztert.
- m = Folyamatos szektorművelet jelzőbit.
- m=0, Csak egy szektor írása/olvasása.
- m=1, Egymást követő szektorok írása/olvasása.
- a = Adatjelzés típusa.
- a=0, Normál adatjelzést ír.
- a=1, Törölt adatjelzést ír.
- E = Beállási késleltetés.
- E=0, Nincs késleltetés.
- E=1, 30ms késleltetés a művelet megkezdése előtt.
- P = Írás előkompenzáció.
- P=0, Előkompenzáció engedélyezve.
- p=1, Előkompenzáció tiltva.
- I3-I0 = Megszakítási mód.

Az állapotregiszter biteinek jelentése:

- b7 --- Motor bekapcsolt állapotának jelzése. Ha a bit 1-es, a drive motorja be van kapcsolva.
- b6 --- Írsvédelem jelzése. A bit 1-es állapotú ha írsvédett lemezre próbáltunk írni.
- b5 --- Fejmozgatási parancsoknál a motor felpörgetési folyamat befejezését jelzi 1-es értékkel, olvasási parancsoknál a rekord típusát

mutatja. 0=Normál adatjelzés, 1=törölt adatjelzés.

b4 --- Szektor nem található. 1-es értéket vesz fel ha a keresett track-et, szektort vagy oldalt nem találja.

b3 --- CRC hiba. Beállított állapotú ha adathibát talált.

b2 --- Adatvesztés/Track 0. Fejléptetési parancsoknál jelzi, hogy a fej a 0-ás track-en van (=1) vagy nem (=0). Íráskor és olvasáskor 1-es ha adatvesztés történt (nem lett kiolvasva az adat mire a következő megérkezett vagy nem lett beírva az adatregiszterbe időben).

b1 --- Adat kérés. Olvasáskor jelzi (=1), hogy kiolvasható a bájt az adatregiszterből, íráskor beállítva, ha beírható a következő bájt.

b0 --- Foglaltság jelző. 1-es értékű ha a parancs végrehajtás alatt van.

A lemezműveletek használata a következő, a megfelelő parancs kódját be kell írni a WD parancsregiszterbe a 10h porton keresztül. Ezt csak akkor tehetjük meg, ha az állapotjelző bájt 0. bitje, vagyis a foglaltság jelző törölt állapotú. A művelet befejezése után a státusz bájt kiolvasásával ellenőrizhetjük, hogy a folyamat hibamentesen zajlott-e le vagy, hogy milyen hiba történt. Most pedig következzen az egyes parancsok részletes leírása.

I. Fejmozgató parancsok

Ezen utasítások mindegyikénél a parancsbájt alsó két bitje adja meg a fejléptetés sebességét, vagyis, hogy két léptető impulzus kiadása között mennyi legyen a várakozási idő.

Az alaphelyzetbe állító parancsnak semmilyen más adatra nincs szüksége, hatására az író/olvasó fej a 0-ás sávra áll be. A művelet befejeztével a track-regiszter nullázódik.

Fej pozicionáló parancs kiadása előtt a track regiszterbe az aktuális track számát kell beírni, az adatregiszterbe pedig a beállítandó sáv számát. Az utasítás befejeztével a fej a kívánt sávra fog állni és a track-regiszter is ezt az értéket fogja tartalmazni.

A léptetés parancsal egy pozícióval léptethetjük a fejet abba az irányba amelyet utólagra használtunk. Az u-val jelzett bitől függően a WD a track-regisztert is beállítja.

A fej befelé léptetésével a nagyobb számú sávok felé, a kifelé léptetéssel a 0-ás sáv felé lép a fej. Itt szintén az u bit állapotától függ, hogy a track-regisztert is növeli/csökkeníti vagy sem.

II. Szektorműveletek

Szektor írási és olvasási utasításoknál a parancskód kiadása előtt be kell állítani a fejet a kívánt sávra, majd be kell írni a szektor-regiszterbe a megfelelő szektor számát. Az E bit állapotától függően a parancs kiadása után vagy azonnal megkezdődik a végrehajtás vagy egy 30ms-os késleltetés után. Ha nem találja a megcélzott szektort, akkor szektor nem található hibával befejeződik a végrehajtás. Ha az m bit beállított állapotú, akkor az írás/olvasás nem csak egy szektorra vonatkozik, hanem egészen addig tart amíg a sávra található legnagyobb számú szektor is sorra kerül.

Szektor olvasásakor az egyes bájtokat az adatregiszterből lehet kiolvasni. Azt, hogy készen áll-e a bájt a kiolvasásra az állapot bájt 1-es bitjéből lehet megállapítani. Ha ez a bit egyes akkor, kiolvasható az adat, kiolvasás után a bit törődik, és az is marad még a következő bájt készen nem lesz. Ha egy adatbájt nem kerül kiolvasásra a következő bájt megérkezéséig, akkor az állapot bájt 2-es bitje beállított állapotúvá válik, ami az adatvesztést jelzi, de a beolvasás ekkor sem szakad meg.

Szektor írásakor a parancs kiadása után az adatregiszterbe kell sorban beírni az adatokat amit a lemezre akarunk írni. Az állapot bájt 1-es bitjének beállított állapota jelzi, ha küldhetjük a következő adatot. Ha nem írjuk be a bájtot az adatregiszterbe időben akkor beállított lesz az adatvesztés jelzőbit és a lemezre 0-ás bájt kerül kírásra.

A cím olvasási parancs a szektor olvasáshoz hasonlóan működik, de itt nem lehet megadni szektorszámot, hanem a legelső megtalált szektorazonosító blokkot lehet beolvasni. Egy ilyen blokk 6 bájt hosszúságú, és a következő adatokat tartalmazza:

1. bájt — Track száma. (0-79)
2. bájt — Lemezoldal száma. (0-1)
3. bájt — Szektor száma. (0-255)
4. bájt — Szektor mérete. (0-3)
5. bájt — CRC 1
6. bájt — CRC 2

A szektorméretek az alábbiak lehetnek:

- 00 = 128 bájtos szektor
- 01 = 256 bájtos szektor
- 02 = 512 bájtos szektor
- 03 = 1024 bájtos szektor

IBM formátumú lemez mindig 512 bájtos szektorokat tartalmaz. Ez a funkció a fejpozicionálás ellenőrzésére használható.

(folytatjuk...)
-DEVIL-

Az Életjáték (3)

(Folytatás az előző számból)

Már majdnem készen vagyunk, csak azt kell eldöntenünk, hogy mit jelentsen a SOROKON-VÉGIG és az OSZLOPOKON VÉGIG. Mivel a táblázat legszélein a SZOMSZÉDOK függvény nem működik (ki-lóg a lóláb, akarom mondani, az index az indextárból), a LÉPÉS célszerűen a második sortól és oszloptól az utolsó előtti sorig és oszlopig megy majd. Mielőtt elhamarkodnánk, definiálunk néhány konstanst:

```
CONST
SORSZÁM = 24;
OSZLOPSZÁM = 38;
```

Első próbálkozásként alkalmazkodunk a megjeleníthető mérethez, nem elhanyagolva keserű tapasztalatunkat, hogy az utolsó két oszlopot az EDITOR „nem szereti”, esetleg átvizsi kedves sejtjeinket a következő sorba. Később úgyis rájövünk, hogy ez az élettér komolyabb konfiguráció esetén kevés, akkor csak ezt a két konstanst kell majd módosítani. Akkor viszont majd ügyelni kell arra, hogy mi látszik az ábrából.

Erről eszembe jut, hogy jelenleg semmi sem látszik. Kacérkodunk a gondolattal, hogy esetleg egy kettős ciklussal minden LÉPÉS után végigsöprünk az aktuális táblán, és kirajzoljuk a képernyőre a tartalmát. Ez persze rettenetesen lassú lenne. Azután eszünkbe jut, hogy egy-egy lépés során a konfiguráció nem nagyon változhat, hiszen minden sejt csak a közvetlen környezetére gyakorol hatást. Ezért úgy döntünk, hogy mindig csak a változást rajzoljuk, ehhez pedig (bár programozói énünk — helyesen — berzenkedik ellene) a rajzolást becsépesszük a FÜGGVÉNY függvénybe (ott ugyanis mindig tudjuk, van-e változás). Emiatt a FÜGGVÉNY paramétereit kibővílnék a sor- és az oszlopkordinátával.

Ha az ELŐKÉSZÍTÉS eljárás KEZDŐKONFIGURÁCIÓ részbe beleügyeskedünk egy egyszerű kezdő ábrát, nem feledkezve meg annak kirajolásáról sem, akkor egy Pascal-szerű nyelvet használva szinte változtatás nélkül írhatnánk a programot; ha beziklábosan (bocsánat, BASIC-ben) dolgozunk, akkor viszont egy kevés kódolási munkával az alábbi programhoz jutunk:

```
100 PROGRAM "LIFEGAME.BAS"
110 NUMERIC SORSZ,OSZLSZ
120 LET SORSZ=24:LET OSZLSZ=38
130 NUMERIC E(1 TO 24,1 TO 38)
140 NUMERIC F(1 TO 24,1 TO 38)
150 NUMERIC SOR1,SOR2,OSZL1,OSZL2
160 LET SOR1=1:LET SOR2=SORSZ:LET OSZL1=1:
LET OSZL2=OSZLSZ
170 NUMERIC Q
180 LET Q=1
190 DEF KEZD
200 NUMERIC II,JJ
210 FOR II=1 TO SORSZ
220 FOR JJ=1 TO OSZLSZ
230 LET E(II,JJ)=0
240 LET F(II,JJ)=0
250 NEXT JJ
260 NEXT II
270 LET E(3,3)=1:LET E(3,4)=1:LET E(3,5)=1
280 PRINT AT 3,3:"***";
290 END DEF
300 DEF SZOMSZ(S,O)
310 IF Q>0 THEN
320 LET SZOMSZ=(E(S-1,O-1)+E(S-1,O)+
E(S-1,O+1)+E(S,O-1)+E(S,O+1)+
E(S+1,O-1)+E(S+1,O)+E(S+1,O+1))
330 ELSE
340 LET SZOMSZ=(F(S-1,O-1)+F(S-1,O)+
F(S-1,O+1)+F(S,O-1)+F(S,O+1)+
F(S+1,O-1)+F(S+1,O)+F(S+1,O+1))
350 END IF
360 END DEF
370 DEF FUGGV(SZ,P,K,L)
380 IF P=0 THEN
390 IF SZ=3 THEN
400 LET FUGGV=1
410 PRINT AT K,L:"***";
420 ELSE
430 LET FUGGV=0
440 END IF
450 ELSE
460 IF SZ<2 OR SZ>3 THEN
470 LET FUGGV=0
480 PRINT AT K,L:" ";
490 ELSE
500 LET FUGGV=1
510 END IF
```

Ne indítsuk el! Borzalmasan lassú lesz. Az előkészítés mintegy fél percig megy — ezt még kibírnánk — viszont egy-egy generáció generálása (sic) két és fél percig tart, ami tarthatatlan (sic). Az egyetlen gyors megoldás, ha a programot azonnali lefordítjuk a ZZZZZZZZZZZIP-el (itt egy kicsit megszaladtak a Z betűk, elég lesz belőlük három is). (Ez megmagyarázza a lokális változók suta megválasztását: a ZZZIP nem fogad el azonos nevű változókat. A ZZZIP által ránk erőszakolt megkötések nélkül egyébként még egyszerűbben és szebben meg lehetett volna csinálni a programot.)

Így már elfogadható a sebesség, de még nem mondtuk ki az utolsó szót. Akinek kedve van, megpróbálhatja gyorsítani azt. Természetesen nem bitfaragásra gondolunk, hanem az algoritmus javítására.

Egy optimalizálást már „elkövettünk” menet közben: nem hagytuk magunkat rábeszélteni a teljes képmező újrajrészolására. Enélkül a program csak elméleti érdekességgel bírna, használni nem lehetne. Ezzel érjük el egyébként a legnagyobb javulást a sebességben, és nemcsak BASIC-ben, hanem más, gyorsabb programnyelveken is.

Egy további lehetőség a SZOMSZÉDOK függvény optimalizálása: a megoldás jelenleg nem vesz tudomást arról, hogy a kiszámolandó összeg egy része az előző cellánál kiszámolt összegben már „benne van”. Kérdés, hogy ezzel lehet-e időt megtakarítani.

Egy másik (nagyon hatásos) beavatkozás, ha nem „faltól-falig” meggyünk, hiszen az ábra általában csak a sejtternek egy kis részét foglalja el. Itt azt lehet kihasználni, hogy az ábra alsó és felső, illetve bal és jobb szélső eleme minden lépésben legfeljebb egy mezőnyivel terjeszkedhet. Elegendő tehát egy minden lépésben újra meghatározandó korlátpár-pár között vizsgálni a táblázatot.

Természetesen csak akkor jutunk használható programhoz, ha megoldjuk a kiinduló konfiguráció berajzolását. Nem árt figyelni, hogy a sejtalmaz nem terjed-e túl a rendelkezésre álló területen. (Ilyenkor továbbmenve hamis eredményt kapunk.) Ha pedig az optimalizálást túlzásba vittük, nem árt a lépésenkénti végrehajtásra felkészülni, hogy egyáltalán lássunk is valamit. (Bár, lehet, hogy ez még odébb van...)

Ha eluntuk az Életjátékot, a SZOMSZÉDOK és a FÜGGVÉNY függvény átírásával egyéb sejt-szimulációt is előállíthatunk. Nagyon érdekes pl. az az eset, amikor csak az élszomszédokat vesszük figyelembe, és a szám paritását vizsgáljuk (páros számú szomszéd esetén lesz sejt a cellában, páratlan számúnál nem lesz). Ilyenkor a sejtalakzat néhány lépésben megnégyszerezzi önmagát. („Kész az önreprodukáló sejtautomata! — kiáltunk fel, de korai az öröm, a szakemberek szerint ez nem az!).

„A mű kész, az alkotó pihen” — mondta egy fehér szakállú programozó, amikor egy kicsit más jellegű életjátéka végre úgy-ahogy elindult (a program azóta sem hibátlan, pedig alkotója már minimum a 6.2 verzióánál tart vele). Csak azt akartuk megmutatni, hogy átgondolt, korszerű programtervezési módszerrel nem kell hat nap egy ekkora program létrehozásához. Körülbelül egy órai munkával (az eszközök: papír, ceruza, fej) és tizegynéhány percnyi begépeléssel azonnal működő, toldozást-foldozást nem igénylő program készíthető.

-UL-

```
520 END IF
530 END DEF
540 DEF LEP
550 NUMERIC I,J
560 FOR I=1+1 TO SORSZ-1
570 FOR J=1+1 TO OSZLSZ-1
580 IF Q>0 THEN
590 LET F(I,J)=FUGGV(SZOMSZ(I,J),E(I,J),I,J)
600 I PRINT AT 23,1:I,J,SZOMSZ(I,J),F(I,J);
610 ELSE
620 LET E(I,J)=FUGGV(SZOMSZ(I,J),F(I,J),I,J)
630 I PRINT AT 23,1:I,J,SZOMSZ(I,J),E(I,J);
640 END IF
650 I DO
660 I LOOP UNTIL INKEY$<"
670 NEXT J
680 NEXT I
690 END DEF
700 CLEAR SCREEN
710 CALL KEZD
720 DO
730 CALL LEP
740 LET Q=-Q
750 LOOP
```


Lakat alatt

Sok programozó szeretné a kisebb-nagyobb méretű alkotásait a külvilágtól elrejtetni, és a programrészek módosításait lehetőleg elkerülni. A feladat ellátására már születtek különböző módszerek és programok, ilyenek már az Enterpressben is jelentek meg. Az *NLIST3* segítségével elkészített program szokatlan változásokat okoz a basic interpreter működésében.

1. Betöltés után automatikusan elindul.
2. Letiltja a melegezetet és a stop megszakítást.
3. Ha a futás valamilyen hiba miatt megszakad, akkor sem lehet a programhoz hozzáférni, ugyanis számos utasítás le van tiltva. A LIST, LLIST, MERGE, LOAD, SAVE, VERIFY, CHAIN, INFO, NEW, DELETE parancsmódban "NOT UNDERSTOOD", programmódban pedig hidegresetet fog okozni.

Cél szerű a forrásprogramot is megőrizni, hogy a későbbiekben szükségessé váló módosításokat el lehessen végezni. Az *NLIST3* a fordításhoz meghajtót vagy 2 magnót igényel. Az utóbbi esetben:

```
inpFILE tape1:forrásprogram neve
outFILE tape2:védettprogram neve
neveket kell megadni.
```

(Hsoft)

```
100 PROGRAM "NLIST3.PRG"
110 ALLOCATE 215
120 CODE COPY=HEX$( "3E 01 F7 05 DF 3E 02 F7 07 DF
2B 7C 85 20 F1 3E 02 01 80 00 11" ) & WORD$( COPY +
26 ) & HEX$( "F7 08 C9 00 02 99 00 00 00 00 00 00
00 00 00 00 00 00 00 79 86 06 40 81 F3" )
130 CODE =HEX$( "FD A6 82 10 80 00 02 27 C2 FC 64
CD 5F 8C 99 CB F5 78 C9 94 BF 19 37 07 E2 20
10 08 78 32 02 00 84 20 E4 00 88 08 01 10 8D
C0 02 20 30 04 02 14 00 04 57 CB E1 0C 30 00 4" )
140 CODE =HEX$( "76 AC 01 00 35 90 46 13 00 0A 86
40 20 00 40 6E 28 08 01 80 04 F0 48 24 12 36
02 08 C6 C0 11 18 08 00 18 74 0F 8D 4F 70 18
E9 8A 00 69 AC D8 67 E6 70 AC A6 4A 27 0B 05 4" )
150 CODE =HEX$( "C4 42 80 BC 20 18 24 85 42 80 02
49 4C 5A 42 20 94 C9 24 31 01 38 98 49 29 95
06 62 02 81 48 8E 2E 24 18 40 87 53 99 AC C2
20 26 25 4E 67 A3 61 BC 40 31 1C 8E 46 19 4" )
160 POKE 540,190:POKE 541,190
170 CODE =" NLIST3 HALUSKA LASZLO 1990."
180 INPUT PROMPT "inpFILE>":AS
190 OPEN #1:AS
200 INPUT PROMPT "outFILE>":AS
210 OPEN #2:AS ACCESS OUTPUT
220 LET B$=""
230 FOR X=1 TO 16
240 GET #1:AS
250 LET B$=B$&AS:PRINT #2:AS;
260 NEXT
270 CALL USR(COPY,ORD(B$(3))+ORD(B$(4))*256)
280 CLOSE #1
290 CLOSE #2
```

... Ha én egyszer elkezdek beszélni!

Nem véletlen a cím, a *SPEAK* bővítés segítségével, minden külső eszköz nélkül, szavakat, mondatokat lehet a gépünkkel elővarázsolni, persze a *SpeakE*-nél gyengébb minőségben. Felhasználásával, a programjainkat egyszerű módon elegánsabbá tehetjük. A *SPEAKBAS* program, a begépelést és a futtatást követően, elkészíti szalagra vagy lemezre a *SPEAK* programot. A későbbiekben már csak ezt kell betölteni. Mivel bővítéről van szó, a betöltés után létszólag nem történik semmi. *HELP*, és *HELP SPEAK* parancsokkal meggyőződhetünk arról, hogy a gépünk új bővítést kapott, melyet kikapcsolásig nem is fog elfelejteni. Az *INFO* utasítás után megállapítható, hogy 16 kilobájttal csökkent a szabad memória.

A megszóllatáshoz rendszerparancsot kell kiadni. Bár ezt más rendszer alól is megtehetjük, most csak a Basic alatti használatot fogom elmagyarázni. Parancsmódban, az új sor elején, kettőspont, dollárjel és szóköz után következhet a kimondásra elküldendő szöveg. Az [ENTER] lenyomására a gép beszélni kezd. Programmódból:

```
EXT "$ szöveg"
vagy
EXT "$ *AS
```

formában is használhatjuk, ahol az AS-ba előzőleg tetszőleges szöveget tehetünk. A szöveg tartalmazhat kis és nagybetűket egyaránt, valamint szünetjeleket, melyekkel a szavak közötti vázkozást jelezzük. Ezek hosszúsági sorrendben a következők: *chr\$(64)*, szóköz, vessző, pont.

Mindegyik jel dupla idejű mint az előző. Megváltozik a kijelzésük azoknak a betűknek, melyeket *chr\$(39)* karakter követ. Az egyéb karakterek figyelmen kívül maradnak. A magánhangzókat duplázással, vagy *chr\$(39)*-el hangfűzhető. Az Y-t tartalmazó kettős mássalhangzóknál, Y helyett használjuk *chr\$(39)*-et, vagy *chr\$(64)*-et. Kettőzött T betű között hasznos lehet a *chr\$(64)*.

Lehetőségünk van a hangerő, hangmagasság, hangszín és a szavak közötti szünetek idejének módosítására is, sorrendben a 150-153-as EXOS változók átírásával.

- 150 - hangerő
- 151 - hangmagasság
- 152 - hangszín
- 153 - szünetek

Kísérletezhetünk, hogy a számunkra legmegfelelőbb hangzást hozzuk ki a gépből. Akkor sem kell kétségbe esni, ha teljesen elrontjuk az érthetőséget, mert

:\$\$ (kettőzött dollárjel)

parancs kiadásával alaphelyzetre állíthatjuk a programot. Két példát mutatok a módosításra:

```
SET 150,15 /a hangerőt halkabbra állítja
SET 151,18 /a gép magasabb hangon, gyorsabban fog beszélni.
(Hsoft)
```

```
100 PROGRAM "SPEAK.BAS"
110 OPEN #1:"SPEAK" 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 6204 0000 0000 0000 0000 0000 0000
210 DATA 793D C83D CAD7 C03D CAA9 C03D C07B FE9A
220 DATA D0D6 96D8 2168 C485 6F8C 9567 0528 0756
230 DATA 0520 037A 2F57 72AF 4FC9 5350 4541 4820
240 DATA 7665 7273 696F 6E20 312E 300D 0A20 2020
250 DATA 2020 2028 4329 2048 534F 4654 2031 3939
260 DATA 302E 0D0A 436F 6D60 616E 6473 3A20 2420
270 DATA 7374 723D 5370 6561 682C 2024 2430 496E
280 DATA 6974 6961 6C0D 0A56 6172 6961 626C 653A
290 DATA 2031 3530 2D31 3533 3D56 6F6C 2C53 7065
300 DATA 6564 2C54 6F6E 652C 5061 6753 650D 0A80
310 DATA C5D5 2134 C020 0A01 1300 EB3D F708 D1C1
320 DATA C9FE 0520 F913 1A96 20F4 2310 F8E1 E111
330 DATA 34C0 0175 003D F708 2129 C118 3AC5 D51A
340 DATA 134F 1AFE 2420 D705 2818 0520 D113 1AFE
350 DATA 2420 CBE1 E121 64C4 1168 C401 0400 ED80
360 DATA AFC9 E1E1 EB23 AF0D C83E 208E 28F7 116C
370 DATA C4D5 ED80 AF12 E1F3 AFD3 A8D3 AC3E 39D3
380 DATA A706 FFC0 9CC1 AFD3 A8D3 ACFB AF4F C945
390 DATA 4E54 4552 5052 4141 4A53 5A2C 4245 535A
400 DATA 4545 4C4F 4F20 5052 4F47 5243 4D42 4F27
410 DATA 5649 4954 4545 532E 4845 4553 5A49 4954
420 DATA 4554 4054 452C 4841 4C55 5348 4120 4C41
430 DATA 4153 5A4C 4F27 2C42 5544 4150 4553 542C
440 DATA 455A 4552 4849 4C45 4E43 535A 4141 5A48
450 DATA 494C 454E 4356 454E 00C8 7FC0 CB77 C8C8
460 DATA AFC9 7E0D 93C1 4847 7987 C8FE 4138 18FE
470 DATA 5830 1478 FE27 792D 04C6 1A06 FFC5 E5CD
480 DATA F3C1 E1C1 2318 DBED 586A C41E 00FE 2E2D
490 DATA DDC1 FE2C CDD0 C1FE 20CD DDC1 FE40 CDD0
500 DATA C118 E128 05CB 3ACB 18C9 F1CD EAC1 18D4
510 DATA 1318 E3E3 7A83 20F9 C921 21C2 4F06 0009
520 DATA 4E21 96C2 097E E60F 4F23 7EE6 8081 4F28
530 DATA 7E07 0707 E607 E528 3411 5FC4 6F26 0019
540 DATA 46E1 237E 2BE5 1711 58C3 6F26 0019 3E80
550 DATA F5A6 2803 3A68 C4D3 A8D3 AC3A 6AC4 3C3D
560 DATA 20FD F105 2807 B70F 30E6 2318 E32A 69C4
570 DATA 2600 2929 29E8 AFCD EAC1 0D79 E60F E120
580 DATA AF79 17D8 2323 1890 0002 060A 0E10 1216
590 DATA 1A1C 2226 2A2E 3234 3842 484A 4E50 5056
600 DATA 1A5C 6466 7074 7A7C C284 86C2 C2C2 888C
610 DATA 9294 C29E A6A8 AE80 C2C2 8E8C 3681 3419
620 DATA 31AB 1819 91C3 3419 31E0 3684 92E3 3519
630 DATA 519C 3131 3496 3687 333A 323D 32C0 1819
640 DATA 519C 3322 3181 3131 36A5 3131 36A8 368A
650 DATA 1819 31AB 1819 511C 3431 3234 3287 2210
660 DATA 1319 21AE 92C3 1819 31E0 368D 3431 3234
670 DATA 3287 1819 711C 92C3 3231 3243 3244 32C5
680 DATA 3F81 3419 3128 333A 323D 32C0 1819 91D3
690 DATA 3319 716D 3293 3E84 9263 333A 323D 32C0
700 DATA 92F3 3E87 3131 3625 3131 3525 3293 3E8A
710 DATA 1819 3128 333A 323D 32C0 1319 3260 1319
720 DATA 71DD 92D3 1819 716D 3293 3E8D 3431 3234
730 DATA 3237 333A 323D 32C0 3253 3254 32D5 1A99
740 DATA E1C3 E1C7 8F0F F803 0F07 C1E3 FF40 17FF
750 DATA 0003 F87C C1F1 F803 FE00 7FFC 0003 F80F
760 DATA 09F1 FE03 EF40 17FF 0003 E15C 35C5 AA35
770 DATA 0000 0000 0000 3E8E 3873 CFF8 78C3 DF1C
780 DATA F1CF FE03 C0FF 0000 FFFB 007F F803 FFF0
790 DATA 01FF E003 AACA 5A05 213D FE1F F800 001F
800 DATA FFFC 2000 0003 FFF7 0879 0002 FFE1 C71F
810 DATA E003 FFD0 01FF F003 7F01 FA5F C007 F80F
820 DATA C0FF 0042 AAA5 555A AAAA 5AA5 5AAA 5555
830 DATA AAAA A555 AA5A AAA5 55AA AAA5 55AA AA55
840 DATA A5A5 AAA5 B766 6CDB F9B3 6CAD 3737 66FC
850 DATA 9887 F6C0 0386 60F7 F73E 4DFB FE5D B7DE
860 DATA 46F6 96B4 4FAA A955 AAAA A569 599A 6A95
870 DATA 5595 556A A555 A94D 666A 92EC A555 D296
880 DATA 55A2 BACD 0066 99CC 6731 BE66 39A6 6819
890 DATA 6659 C671 0967 19CB 0171 CC73 1999 CCC6
900 DATA 6719 9AC6 5900 2E5A 5EFE 3F18 1850 3F18
910 DATA 1850
```


Hat férőhelyes EPROM-bővítő kártya

Az előző számban tárgyalt buszkiterjesztő egységbe illeszthető kártyák közül ismertettem most a legegyszerűbbet. Ezzel reményeim szerint újratervezhetően demonstrálhatom az abban a cikkben leírt tervezési elveket, másrészt a téma kapcsolódik a hardver-rovat nyitó cikkéhez is (Mi lakik a cartridge-ben?).

Mint az közismert, a cartridge lehetőségei bővítőprogramok befordítására több értelemben is végesek. Egyrészt a kártyán csak egy vagy két EPROM (Erasable Programmable Read-Only Memory = törölhető és programozható csak olvasható memória) helyezhető el, másrészt a rendelkezésre álló címtartomány is csupán 64K, azaz négy szegmens. Nem kell azért pánikba esnünk, a jobboldali csatlakozón keresztül több, mint 3 megabájt várja, hogy teletöltsük, csupán a megfelelő hardvert kell megépítenünk hozzá.

A RAM-bővítésekre is gondolva a 4 MB címtartományt középen elfeleztem, így a 08hex - 7Fhex szegmenstartomány juttott az EPROM-oknak. Ez csaknem 2 MB. A szomorú a dologban, hogy bekapcsoláskor az EXOS a cartridge-on kívül csupán a 0-ra végződő hexa sorszámú (10hex, 20hex, ...) szegmenseket vizsgálja, így az azokon kívül elhelyezkedő programokat nem tudja beláncolni, azok észrevétlenül maradnak. Ha egy program több szegmensben terül el, a program felelőssége megtalálni a "függelékait", így tesz például a HiSoft Pascal. Ezek szerint maradt hét helyünk (a 10h, 20h, 30h, 40h, 50h, 60h és 70h sorszámú szegmensek) amelyeken az EXOS méltóztatik a programok jelenlétét észrevenni. Ebből a 20h szegmenst felejtjük is gyorsan el, mivel általában itt található a lemezegység kezelőprogramja, az EXDOS. (Hogy miért csak általában, az a következő szám témája lehetne, de ahhoz még sok előfizetőnek kellene lefolynia a Dunán...)

Maradt hat szegmens, ez már tényleg a miénk, azt csinálhatunk velük, amit csak akarunk! Mondjuk, akarjunk oda EPROM-okat elhelyezni, méghozzá szinte akármilyen típusú! Az ésszerűség határain belül maradva megragadhatunk a 28-lábú EPROM-oknál, ez memóriában a 8 K - 64 K, árban a 300 Ft - 700 Ft tartományt fedi le, így minden szempontból megfelelőnek tűnik. Négy típusról van szó - 2764, 27128, 27256 és 27512, így, mivel ezek lábkiosztását bölcsen a lehetőségeken belül egyformára terveztük, 2 db jumper elegendő lesz egy foglalathál az ott helyet foglaló EPROM fajtájának beállítására.

Egy foglalathalnak legegyszerűbb az x0h - xFh szegmenstartományt kijelölni, igaz, hogy így "pazarlunk", de a veszendőbe menő szegmensekkel nemigen tudnánk úgysem mit kezdeni. A cím kiválasztáshoz fel kell tehát használnunk az A21, A20, A19 és A18 címvezetéseket, továbbá a -MREQ memóriakérés és a -RD olvasás vonalat is. Ez hat vonal, és ugyancsak hat EPROM közül kell választanunk. A feladatot legegyszerűbben egy alkalmasan beégett PROMmal (PROM - Prog-

ammable Read-Only Memory = programozható csak olvasható memória) oldhatjuk meg. A 74S188-as típus (U3) éppen megfelel: az engedélyezéssel (-CE) együtt összesen hat bemenete és nyolc kimenete van. Az utóbbiakból az EPROM-ok kiválasztásához csak hatra van szükségünk, így egy további (U3/1. láb) használhatunk az adatmeghajtó áramkör (U1) engedélyezésére. Az RPI ellenállás-létra a cím kiválasztó nyitott kollektoros kimeneteinek "felhúzására", inaktív állapotban magas szintre hozására szükségesek.

Nézzük részletesebben, hogyan működik a cím kiválasztás! Ha a processzor nem a memóriához fordul, a -MREQ jel magas, így U3 letiltódik. Ha frunk a memóriába, a -RD jel magas marad, azaz a PROM (U3) A4-es lábára logikai 1 kerül. Ez a 10h-1Fh közötti tartományt jelenti, ahol a PROM összes kimenete magas, tehát nem választ ki egyetlen EPROM-ot sem. EPROM-ba frni amúgy sem lehet, viszont ha valaki megpróbálná, enélkül a védelem nélkül kárt tehetne a kártya áramkörében, hiszen a kétirányú buszmeghajtó kimenete kerülne szembe az adott EPROM adatkimenetével, ami az áramkönyvek szerint hosszú, fáradtságos szervizelést ígér. (Tulajdonképpen elég lenne az egyirányú meghajtó is, de így könnyebb volt a nyomtatott áramkört megtervezni, másrészt vannak még egyéb megfontolások is, pl. CMOS sztatikus RAM behelyezhetősége.)

Próbáljunk most egy 7Fh fölötti, mondjuk a 9Bh szegmenst olvasni. Ebből csak a legmagasabb négy cím, azaz a 9-es számú, a -RD jel pedig alacsony, így az U3-ra kerülő cím 09. Itt a PROM tartalma ismét csupa 1-es, azaz nem történik EPROM-kiválasztás. Olvassuk most a 43h szegmenst! Ekkor a fentiek alapján U3-ra 04 kerül, aholis bináris 11101110 található. Ebből a középső 0 kiválasztja az U7-es EPROM-ot, az utolsó 0 pedig engedélyezi az adatmeghajtó áramkört (U1).

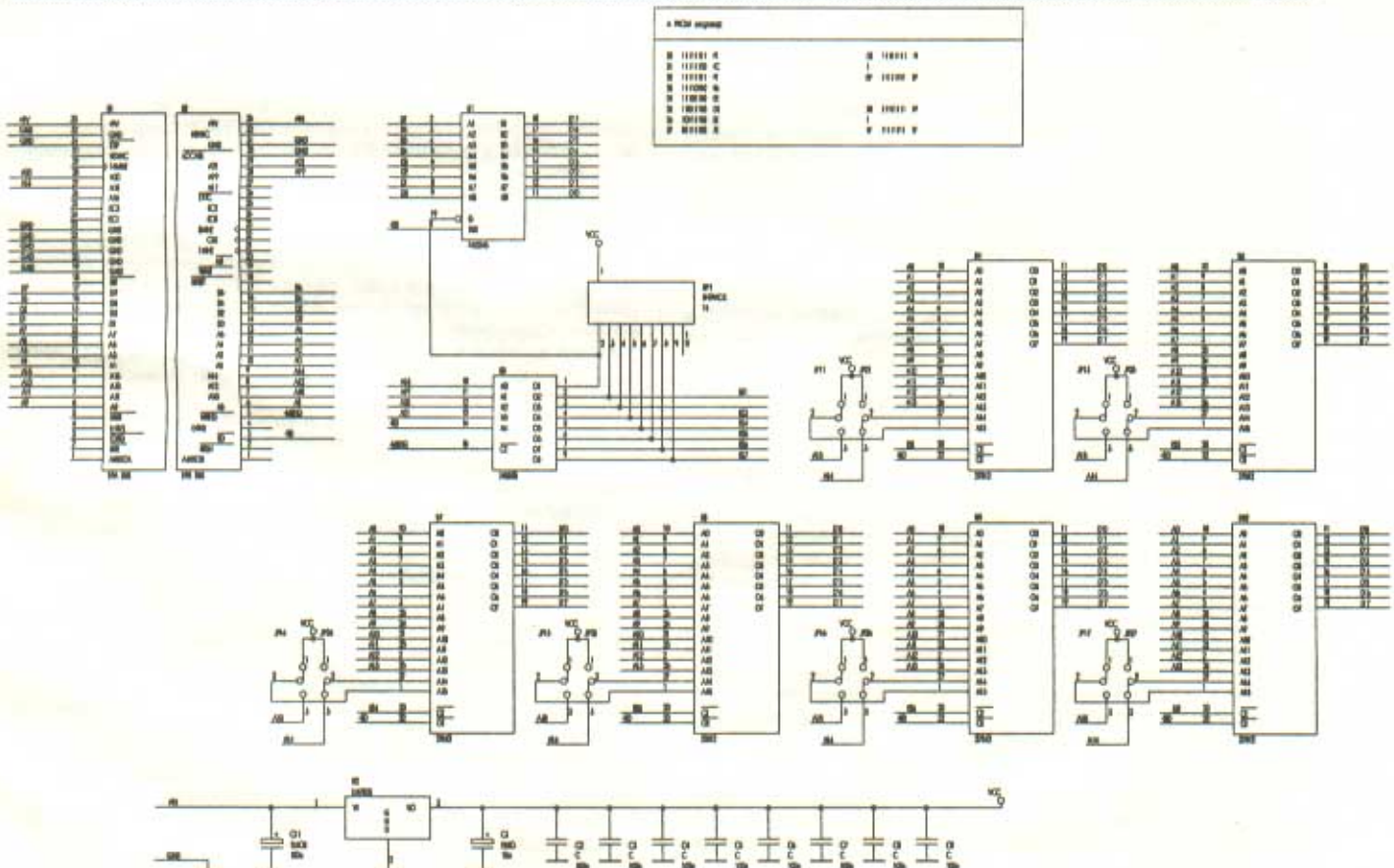
Az EPROM-ok mellett található jumperek hivatottak az adott foglalathalban lévő EPROM típusát kiválasztani. A 27512-es IC-hez mindkét jumpernek lefelé kell állnia. A 27256-oshoz JP1x-nek fölfelé, JP2x-nek lefelé kell állnia (ez szerepel a rajzon). A 27128-as és 2764-es típusok beállítása egyforma: mindkét jumper fölfelé.

Végül, de utolsósorban a kártyán, ahogy azt az elmúlt számban megtárgyaltuk, szerepel egy önálló tápfeszültség-stabilizátor (U2), meg persze az elmaradhatatlan hűdegítő kondenzátorok (C11, C1-C9) is.

Szakértő szemnek feltűnhet, hogy az EPROM-ok címvezetékei csak úgy össze vannak kötve, amely jelentős terhelést okoz. De nincs semmi baj, hiszen a kártya nem közvetlenül a géphez, hanem a buszkiterjesztő egységhez csatlakozik, amelyen a megfelelő meghajtóáramkörök szerepelnek. Ugyanezért nem kell a -RD jelet sem féltetni, amelyet pedig nyolc bemenet terhel.

Mészáros Gyula

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



EPDOS v1.3: Meglátni és megszeretni...

Nagy örömmel szolgálta, hogy a HSoft diadala szerkesztőségünknek tesztelésre legújabb termékét, az EPDOS 1.3-at. Nem csalódtunk a programban...

Az első benyomások

Az EPDOS fizikailag két darabból áll. Az egyik az operációs rendszert tartalmazó 27C256 típusú EPROM, gondosan megcímkézve, Copyright jelzéssel ellátva. A másik az EPDOS 1.2 RENDSZER DISK címkét viselő lemez, amely feltételezésünkkel (és saját címkéjével) el- lentében nem az IS-DOS-nál megismert rendszerlemmezhez hasonló valaminek bizonyult, hanem elsősorban HELP lemezeknek, azaz az operációs rendszer használatát segítő HELP fájlokat tartalmazza.

Üzembehelyezés

A rendszert tartalmazó EPROM CMOS technológiájú, ez azt jelenti, hogy elvileg *gondosan kell vele bánni*. Ezt azt jelentené, hogy szállítani, tárolni csak a lábait elektromosan összekötve szabad (elektromosan vezetőkialakítású szivacsba szúrva; házilag megteszi az ezüstpapír is), foglalatba behelyezni pedig különleges fogóval, amely mindaddig fémesen összeköti az összes lábat, amíg azok a foglalatban a helyükre nem kerülnek. Mivel ilyenkor nem rendelkezünk, maradt a "puszta kéz" technikája. Az EPROM eddig még túlélt...

A kikapcsolt ENTERPRISE gépből kivett BASIC ROM-bővítőt szétszedjük, és mivel az csak gyári, egy foglalatot, óvatosan, két óráscsavarhúzóval használva, kiemeljük a foglalatból az eredeti BASIC ROM-ot. Az EPDOS EPROM kerül a helyére, óvatosan, hogy a lábai meg ne sérüljenek, és ami ugyanolyan fontos, az IC-tok kis kivágásának a foglalat hasonló kivágása irányába kell néznie (ellenkező esetben, volt egy nagyon drága, pár másodpercig világító zselblámpaizzók). Az összeszerelt bővítődoboz visszamegy a helyére, és már be is kapcsolhatjuk a gépet.

Az indulás

Az első kellemes meglepetés: a RAM-teszt villámgyorsan lezajlik - később majd megtudjuk, hogy rendelkezésre áll az EXOS "normál", azaz lassú (viszont teljes!) RAM-tesztje is, és még sok más opció. Ezután az EPDOS kirajzolja körvonalait a képernyőn. Kissé zavaró, hogy a rossz minőségű monitoron nem látszik a 42 ill. 84 karakteres kép két szélé, esetleg a legalsó két-három sor. Legfelül a rendszer azonosító üzenete, alatta sok-sok parancs, egy - egyelőre - üres állapotsor, egy szintén üres hely, nyilván a katalógusnak, vagy ahogy - elég illogikusan - nálunk mondják: a könyvtárnak, újabb állapotsorok. Legalul egy ismerős hibaüzenet:

Not ready - drive A:
Retry, abort (R/A):

Beteszünk egy lemezt a meghajtóba, majd ENTER-t nyomunk. Amint vártuk, a képernyő legnagyobb ablakában megjelenik a lemez katalógusa, a szokásos adatok mellett a fájlok attribútumai is láthatók. Egy sor eltérő színnel jelenik meg, nyilván ez az "aktuális" fájl-bejegyzés, amellyel ezt-azt tehetünk. Mint később kiderül, akárhány fájl kijelölhetünk, és ezekkel csoportos műveleteket végezhetünk. Kicsit zavaró néha, hogy a korábbi hibaüzenetek megmaradnak, és nem tudjuk, érvényesek-e még.

Első látásra az egész úgy néz ki, mint egy ún. *shell*, azaz az operációs rendszerre épülő, de annak funkcióit, szolgáltatásait kibővítő, a meglévőket kényelmesebbé tevő rezidens, tehát állandóan a memóriában lévő program. Nos, ez derék: aki dolgozott már IBM XT vagy AT gépen, az tudja, hogy aki a Norton Commandert vagy a PC-Shellt megismerte, az többé nem képes azok nélkül élni... Az EPDOS-t is egy ezekhez hasonló DOS-shellnek véljük, mindaddig, amíg meg nem győződünk arról, hogy valami többről van szó: az EPDOS-ban az EX-DOS parancsoknak *újnak funkciók mind-mind tovább vannak fejlesztve, azok fő hibái kijavítva*... Egyre izgalmasabbnak tűnik az ismerkedés.

Néhány dolog, persze, nem tetszik: a parancsok között csak vízszintes kurzormozgással közlekedhetünk, a sorok között "rövidíteni" igyekvő önkéntelen mozdulatunkra, azaz a botkormány fel-le mozgására a lenti ablakban, a katalóguskirírásban mozog a kurzor. Szokatlan. Mivel a teszteléshez használt gépünkön a botkormány az egyik irányban kissé "gyengélkedett", ez megnévezte a rendszert kezelését. Egyébként is, a kurzor rettentően gyors; ez végül pont az ellenkező hatást éri el, lelassítja a kezelést, hiszen a billentyűzet automatikus repetálásait nem lehet használni, rendszeresen "túllovunk" a célon, ide-oda ugrálunk az elérhetetlen funkció körül, mfg meg nem tanuljuk apró kis pöccintésekkel kezelni a botkormányt.

Ismerkedés

Rendeteg parancs... nyilvánvaló, hogy itt egy jó kézikönyv nélkül nem boldogulunk. A "csomagban" hiába keressük... de hopp, itt a HELP-lemez, vagy ahogy önmagát nevezi, a rendszerlemez! A lemez

mellékelte... bocsánat, mellé tett nyomtatott ismertetőn a program és a leírás készítőjének neve mellett (ne halgassuk el: a leírás *Bognár Attila* dícséri) megadja a HELP, azaz a segítség vagy súgás használatát. A lemez árát is megtaláljuk ugyanitt, ez azt jelenti, hogy az EPDOS EP-ROM árában ez nem szerepel...

Itt mondjuk el, hogy ha a rendszert úgy indítjuk el, hogy a HELP lemez már a meghajtóban van, egy döbbenetes élményben lesz részünk: *az ENTERPRISE megszólal, és a képernyőn lassan, soronként megjelenő bemutatkozó szöveget felolvassa*. Ez a kis csemege a lemezen lévő demó része. Ha nem éjszaka kezdünk bele az EPDOS-szal való ismerkedésbe, amikor is rémült testvérek, gyerekek stb. sírnak fel, és dühös szülők, feleségek, barátok stb. támadnak nekünk (Te és a te hülye szórakozásod stb.), akkor gépünk hangját egészen kellemesnek, kiejtését pedig türhetőnek találjuk, és amúgy hozzáolvasva az eredetét, még értendő is. Gratula.

Persze, a HELP-lemez ennél sokkal fontosabb feladatot lát el, az egyes (időnként igen szűkszavú) parancsok részletes, gondosan kivitelezett ismertetését. Itt csak két fenntartásunk van: Az egyik, hogy *a lemez tartalma, legalábbis "hagyományos" eszközökkel, nem nyomtatható ki*. Ez az akadály szándékos, és nyilván a jogtalan másolás elleni védekezést szolgálja; ehelyett a felhasználó jogos igényét csorbítja, hogy egy bármikor fellelőzhető kézikönyvet készíthessen magának. Hiába van kéznél a HELP szolgáltatás, ha használatához állandóan lemezt kell cserélni (a felhasználók többségének egy lemezegysége van; ha van...). A másik kifogásunk, hiába, szakmai (szakújságról) jellegű: *rengeteg a helyesírási hiba, főleg hosszú-rövid ékezet (következésképpen rossz) felcserélése, felesleges vesszők*... Egy ilyen jó termékben már erre is kell ügyelni! Reméljük, hogy az újabb kiadásban sikerül ezeket kijavítani...

Részletes ismertetés...

...az sajnos, nem lesz: nem hogy ez a cikk, de egy teljes ENTERPRESS lapszám is kevés lenne hozzá. Csak egészen röviden tudunk átrohanni a parancsokon, megemlítve azt, ami tetszik... vagy éppen nem tetszik.

Külön-külön paranccsal hívhatjuk meg a BASIC, a WP, az EX-DOS, az IS-DOS, az ASMON (ASMEN), a MON (MON5) és a GEN szolgáltatásait, ha ezek valamelyike ROM-ban vagy RAM-rezidens rendszerbővítőként elérhető. Ha nem, az EPDOS megpróbálja lemeztől betölteni a hívott rendszert.

Az EXT parancs számtalan szolgáltatást nyújt, többek között egy valódi monitort (az öreg CP/M-csek szíve megdobban...), a FORMAT parancs rengeteg lemezformátumot megenged, szabványosakat és nem szabványosokat egyaránt. Ha lemezegységünk megengedi, az előírt 40 vagy 80 sávon felül még újabb sávokat hozhatunk létre a lemezen. A FRESH parancs (mi inkább REFRESH-nek neveztük volna el) felfrisíti a hosszú tárolásban esetleg meggyengült mágnességet a lemezekben.

Az EXDOS-ból ismert parancsok mind-mind "megismosodva" köszönnek vissza az EPDOS-ban. A megszokott TYPE parancs például nemcsak kírja a kijelölt fájl tartalmát, hanem egy WP-szerű szöveg-szerkesztőbe belépve, átnézhetjük vagy rögtön kijavíthatjuk, újraírhatjuk a szöveget. Ehhez jön a már említett EXT parancs HFONT szolgáltatása, amely *beül egy nagyon szépen kidolgozott magyar karakterkészletet*. Külön öröm volt tapasztalni, hogy az ékezetes betűk elérésére pontosan ugyanazt a megoldást találta ki a szerző, mint amit e sorok írója megszokott, így nem jelentett gondot azon melegeben EPDOS alatt megírni ezt a kis cikket!

Ugyanakkor *képtelenek voltak a szerkesztés alatt álló fájlról más néven egy közönséges biztonsági másolatot készíteni*: a COPY mindenáron egy másik lemezegységre akart másolni. Lehet, hogy csak mi vagyunk ügyetlenek?

Ahogy az érdekesebb dolgokhoz érünk, egyre szorítóbb a helyszűke: a DCOPY a teljes rendelkezésre álló memóriát kihasználja, a minimumra csökkentve a lemezcsere számát; az UNDEL visszaállítja a törölt fájl, ha annak tartalma még elérhető. A fájlok sorrendje a lemezen átrendezhető. A merészebbeknek *disk-editor áll készenlétben*.

Összefoglalva: Az EPDOS egy olyan eszköz a gépet ismerő és még jobban megismerni vágyó ENTERPRISE tulajdonosnak, amelynek sokoldalúságát, egyszerűségét nem lehet kellően méltatni. Beszerzését bátran merjük ajánlani minden komoly felhasználónak: nem fogja megbánni. A szolgáltatások jók, sokrétűek, és minden együtt van: nem kell mindenféle lemezekkel zsonglorkodni, a meghajtó szabad a felhasználó lemezének. A kicsit szűfolt felhasználói felületet viszont nem ártana átdolgozni: a megfelelően csoportosított funkciók áttekinthető, jobban kezelhető lennének legördülő menükön keresztül. -UL-

(Az EPDOS megrendelhető: Haluska László, 1086 Budapest, Karácsony Sándor u. 18. III/41.)

Írás a státuszsorba

A CODE pointer átállításával lehetőséget kapunk, hogy a sztringet közvetlenül a memóriába írjuk. Az alábbi programot lefuttatva, szembetűnő sebességeltérést tapasztalhatunk a hagyományos és a CODE rutint használó státuszírásnál. A módszer addig lesz működőképes, ameddig a Basic nem használja a második lapot. ALLOCATE használat esetén, csak a kód elkészítése után állítsuk el a pointert, vagy alkalmazzunk elmentést és visszaállítást. (Hsofi)

```

100 LET AS="ENTERPRESS A KEDVENC LAPOM,
GUTA ÚT MEG, HA NEM KAPOM... "
110 POKE 56,201
130 ! 1. módszer
140 !
150 FOR I=1 TO 20
160   FOR X=1 TO 34
170     POKE 48829+X,ORD(AS(X))
180   NEXT
190   LET AS=AS(2:)&AS(1)
200 NEXT
220 ! 2. módszer
230 !
240 FOR I=1 TO 1000
250   !
260   ! A SEBESSEG BEALLITASA
270   ! FELKIALTOJELES SOROKKAL
280   !
290   POKE 540,190:POKE 541,190
300   CODE =AS(:34)
310   LET AS=AS(2:)&AS(1)
320 NEXT
330 !
340 POKE 56,245

```

Ilyen még a neppereknél sincs...

SPRED release 1.5

Felhasználóbarát Entersprite kompatibilis sprite editor

- Tömértelen funkció
- Pull-down menürendszer
- Esztétikus kivitel
- Exdos használat
- 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 álland.

Az
ALAPLAP
februári számának
tartalmából:

- A hónap témája: Digitális Babel
- Kooprocesszor specialitások
- Nagyágyú az íróasztalon?
- Program kezdő harkályoknak
- Mi fér a memóriába és hova?
- A DataFlex adatbáziskezelő



CÉDRUS

CÉDRUS
Kiadó Kft.

1441 Budapest, Reguly Antal utca 8.

Fizessen elő a Computerworld-Számítástechnikára! Csak nyerhet!

Informatikai iparunk vezető lapja a hetente megjelenő Computerworld-Számítástechnika. Híreit, 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 cikkei 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



DIZZY III. - FANTASY WORLD

Egyszer volt, hol nem volt, még Tojásland-en is túl, volt egyszer egy Dizzy nevű, tojásdad formájú mutáns lény. Valaha, mikor még az ember nem pusztította a természet zöldellő kincseit, ez a tojás hajótörést szenvedett. Szerencsére egy szigetre kisodorta az óceán hullámai. A sziget neve: Kincsés Sziget (Treasure Island). Meggyötörten ébredt fel a szinte végtelenségig tartó rémálmából, és elindult, hogy hazajusson Tojásland országába, Fantázia Világ városába. (Fantasy World). Nagy örömmel tért haza, szája a füléig ért, de jókedve nem tartott sokáig. Megtudta, hogy Daisy, a feleségét elrabolták a gonosz, kegyetlen király őrői, és Weird Varázsló felhőkastélyába zárták. Dizzy sem télenkedett, elindult, hogy kiszabadítsa egyetlen szerelmét, viszont óvatlansága őt is börtönbe juttatta. A király őrői elfogták még a cél előtt, és a kastély legsötétebb, legmélyebb börtönébe vetették. De Dizzy elhatározta: nem adja fel. Akár az élete árán is megmenti Daisyt.

Betöltés után megismerkedhetünk a Dizzy családdal, akikkel a játék folyamán még beszélgethetünk is. Ezek Dizzy, Daisy - a felesége, Denzil - a bátyja, Dylan - az unokabátyja és Dozy - a nővére. A játék folyamán Dizzy-t bármelyik joy-jal irányíthatjuk, illetve a [Q]-fel, [A]-le, [O]-balra, [P]-jobbra, [ENTER]-tűz billentyűkkel. Tárgyakat úgy tudunk felvenni, hogy odaállunk a tárgy elé, és megnyomjuk a tűzgombot. Ekkor egy ablakban a nálunk lévő tárgyak jelennek meg (kezdetben ez kettőt jelent), köztük azzal a tárggyal, amit éppen felvettünk. Ha a tárgyra mégsem lenne szükségünk, még mindig kiválaszthatjuk, és eldobhatjuk. Ha viszont kell, akkor az "EXT AND DON'T DROP" funkciót kiválasztva nyomjuk meg a tűzgombot. Hasonlóképpen tudunk tárgyat is letenni. De arra vigyázzunk, hogy ha már tele van mindkét kezünk, és felveszünk egy tárgyat, akkor a legrégebben nálunk lévő esik ki Dizzy markából. Hát akkor - kalandra fel...!

Kezdetben még az életszínvonal alatt, a börtönben (The Castle's Dungeon) vagyunk. Jobbról egy ór állja az utunkat, balról pedig egy óriási tűz. Próbáljunk meg az ór mellett átmenni. A troll közli velünk, hogy mi itt addig nem megyünk át, amíg ő itt áll. Majd átmegyünk, ha már nem lesz itt. Hát akkor próbálkozzunk a tűzzel. Ez sem jött be, Dizzyból sült tojás lett. Hát akkor mi legyen? Nálunk van egy alma (FRESH GREEN APPLE). Adjuk oda neki (tegyük le elé), mire a troll megköszöni, és elárulja, hogy a tűzet vízzel tudjuk eloltani. ("Nekem? Nagyon rendes vagy! En kiengednélek, de ha a király kint talál téged, akkor nekem végem! De ki tudsz szabadulni, ha a tűzet eloltod vízzel.") Vegyük hát fel az asztalon lévő vizet (JUG COLD WATER), valamint a megpenészedett kenyert (STALE LOAF OF BREAD). Menjünk a tűzhöz, oltjuk el a vízzel, majd a csempészek rejtékhelyén (Smuggler's Hideout) vegyük fel a nehéz sziklát (HEAVY BOULDER), és ugorjunk fel jobbra. Itt egy patkány állja az utunkat, de ha letesszük elé a penészes kenyeret, akkor a nyakunk helyett a kenyeret kezd el harapdálni, majd kis idő után jóllakik, és elhúzza a csíkot a helyére. Ugorjunk ki az életszínvonal fölé. A kastély halljában vagyunk (Entrance Hall). Ugráljunk fel a képig, és nyomjuk meg a tűzgombot. Felnézünk a képre, és látjuk... hiszen ez bennünket ábrázol! Úgy látszik a király trolljai követtek, és így tudták Daisy-t is elrabolni, illetve minket lefényképezni. Ha már meguntuk a művészi fotó elemzését, akkor ugorjunk át jobbra a platóra. Itt a harmadik korlátdarabkát (PIECE OF RAILING) kiemelem megtaláljuk az első coin (01). (Hurrá!!! Már csak 29 kell! - EPY) A korlátdarabot tegyük vissza, és menjünk jobbra a keleti szárnyba (East Wing), és balra felugorva vegyük fel a következő érmét is (02). Balra ismét tűz állja az utunkat, de víz sajnos nincs sem a kezünkben, sem a környéken, így kerülő úton kell mennünk. Vissza a hallba, majd onnan menjünk a balra lévő platón. Lent egy coin vár ránk, de még ne menjünk le érte, ugyanis megyünk még arra. Minket inkább a mellettünk lévő kapcsoló érdekeljen, amit állítsunk át ON állásba, mire a lent lévő vasrács elkezd fel-le mozogni. Menjünk balra tovább a nyugati szárnyba (West Wing), ahol vegyük fel a pénzdarabot (03), majd tovább jobbra. A virágcserepről ugorva egy asztalra érkezünk a bankett terembe (Banquet Hall). Itt is egy coin (04); no csak, Denzil is itt hallgatja a legújabb Metallica számot, az Enter Sandman-t.

- Mit csinálsz itt Denzil, nem tudod, hogy ez egy veszélyes hely?

- Hé, maradj nyugton Diz! Láttam a király elmenni! - válaszol Denzil, miközben a magnó Volume feliratú gombját lefelé csavarja.

- Daisy-t és engem elfogtak a gonosz király trolljai, engem börtönbe vetettek, őt pedig a Varázslók Kastélyába zárták. - panaszkodik Dizzy, mintha nem tudná, hogy Denzil segítségét hívja is várja.

- Oh, mi mindannyian izgultunk, hogy hova a fenébe tűrtetek. Sajnos én nem tudok neked segíteni, mert túl elfoglalt vagyok. De itt van a köteled, amit a múlt héten kölcsönadtál. - mondja, majd a további beszélgetést elkerülvén ismét csumára teszi a hangerőt. A beszélgetés elég hosszú volt, hiszen most már egy teljesen más zene szűrődik ki a fejhallgatóból, talán Bach d-moll toccata és fuga (BWV 565) című darabja.

Vegyük fel a kötelet (Piece Of Rope), és menjünk vissza a hallba. Innen jobbra egy alligátor úszkál a vízben. Ugorjunk rá, és tegyük le a kötelet. A kötéllal sikerült bekötözünk az alligátor száját, így ezentúl bármikor átugráthatunk rajta. A tőlparton vegyük fel a másik sziklát is (HEAVY BOULDER), majd mindkettőt tegyük le a hallban. Menjünk vissza Denzilhez, és az itt álló asztal széléről ugorjunk balra. Egy kiálló téglát tart meg minket, úgyhogy ugorjunk még feljebb. A kastély lépcsőházában vagyunk (Castle Staircase). Jobbra egy ajtó van. Próbáljunk meg bemenni, de az ajtó zárva, és az áll rajta: "Kopogj és lépj be!". Hát kopogni nem tudunk, de Dizzy szerint nem lenne az ajtó probléma, ha lenne egy bokszesztyűnk. Ugráljunk fel a lépcsőn a coinért (05), és tovább fel a padlásra (Attic) találunk egy csontot (FRESH MEATY BONE). Vegyük fel, majd a lépcsőháztól jobbra lévő keleti toronyban (East Tower) lévő fényes aranykulccsal együtt (SHINY GOLD KEY) vigyük a hallba. A csonttal és egy sziklával a kezünkben menjünk balra. Itt van a már megcsodált érme (06), és a vasajtó is. Menjünk át alatta egészen a következő pályáig, ahol egy Amorog nevezetű orrszarvú áll előttünk. Alattunk van a barlangja, úgyhogy menjünk le, és az itt lévő sziklát cseréljük ki a csontra, majd ugorjunk vissza. Az orrszarvú beszalad a balrangiába, és úgy néz ki, elég sokáig el lesz foglalva a csonttal. Menjünk tovább a következő pályán vigyázva a madárra (Mikor a felhők fölé ér, akkor tudunk átmenni gyorsan), és a pénzdarabot felvéve (07) egészen a törött hídig (Broken Bridge). Dobáljuk bele a vízbe a két követ, majd a hallból hozzuk el a harmadikat, és ezt is dobjuk bele. Ekkorra a víz szintje már elég magas, hogy átugráljunk rajta. Vegyük fel a kulcsot (SHINY GOLD KEY), majd a következő pályán fel kell ugranunk a tetőt tartó oszlopra. Ez úgy sikerül, ha hordó alatti lépcsőfokról ugrunk balra a ládára, majd a másik ládára átsétálunk, amelynek a jobb széléről már fel tudunk ugrani a tartóoszlopra. Innen már csak balra kell ugranunk, és mienk a felhőn lévő coin (08). Majd pottyanjunk vissza, és a legelső ládáról ugorjunk balra, s leesünk a ládák közé. Innen le tudunk esni ahol egy érme vár reánk (09). Néhány ugrással ki tudunk jutni, de ezzel szenvedjen csak az olvasó!!! Majd jobbra haladva a legszélső láda (nem a mólón lévő!) bal széléről ugorva mienk a következő coin is (10). Menjünk végig a mólón, de vigyázzunk, mert az egyik deszka hiányzik, így könnyen megzúdulhatunk a sós vízben. Itt napozik Dozy.

- Hé, Dozy! Kelj már fel! - rüg bele Dizzy a napozószékbe, melyben Dozy fekszik.

- Oh, mi bajod Dizzy? - kérdezi felriadva rémálmából még álmos szemekkel.

- Daisyt elrabolták és a Varázslók felhőkastélyában tartják fogva, és senki sem hajlandó segíteni nekem.

- Ahhh, sebjaj Dizzy! Én segítek neked. Itt van egy kis altató. Ez segíteni fog. - válaszol Dozy, és letesz Dizzy elé egy üveget.

- De én azt akarom, hogy TE segís nekem! - kéri leli Dizzy tovább, de úgy néz ki, Dozy sem fog segíteni.

- Sajnálom Dizzy, én is szeretnék segíteni, de túl szép ez a nap ahhoz, hogy egy lányt kiszabadítsunk. - mondta, ástott egyet, majd elaludt.

Az altatóval (SLEEPING POTION) menjünk a krokodiltól jobbra, és a sárkányt kissé megközelítve tegyük le az altatót. A sárkány tüze kialudt, s fejét lehajtván elaludt. Hozzuk el a másik

aranykulcsocskát is a hallból, és menjünk jobbra. A kút tetején egy zsák van, amit felvéve most már egyszerre öt tárgy lehet nálunk. Majd álljunk a kút jobb szélére, és lépjük le a fa lombját. Alatta egy coin (11) találtunk, majd a lombot tegyük is vissza. Ugorjunk fel a platóra, és az utolsó kerítésdarab mögött lévő érmét is vegyük fel (12). Menjünk jobbra a liftvezérlő konyhó fölött (The Lift Control Hut), és a plató végén essünk le, majd itt jöjjünk visszafelé. A fal mellett ismét egy farács mögött van elrejtve egy coin (13), majd ezt felvéve ennek a platónak is essünk le a végén, vegyük fel a pénzdarabot (14), majd jobbra menve vegyük fel a tehénkét (CUTE PIGMY COW). A trágyát is felvehetjük, de sajnos mindig kicsúszik szegény Dizzy kezéből. (Pedig milyen jó lehet egy kis meleg trágyát nyomkodni a kezünkben.-EPY) Továbbhaladva megtalálhatjuk Dylant is, de még ne foglalkozunk vele. Először menjünk Dylan háta mögé fel, majd ugorjunk balra. A tölgyfa jobb ágára érkezünk. Ugorjunk tovább a túlfoldalra, és itt a lombok között van egy coin (15). Most már beszélgethet Dizzy Dylan-nal:

- Hé, mi történt? - kérdezi Dylan, nem mintha érdekelné.

- Kérve kérek, segíts nekem Dylan! Megpróbálom megmenteni Daisyt, de nem találok a felhőkastélyt! - mondja neki Dizzy, de nem számíthat még arra sem, hogy válasza méltassa.

- Ez igen egyszerű! Emlékezz csak Jack-re, ő hogy találta meg a kastélyt. - utal a Jack and the Beanstalk mesére, majd teljesen az ital hatása alá kerül.

Miután megemésztettük Dylan hülyeségeit, menjünk jobbra, a vulkán tövében (Base Of The Vulcano) vegyük fel a kulcsot (SHINY GOLD KEY), majd a működő vulkán tetején (Active Volcano) a coin (16). Menjünk vissza balra egészen a liftvezérlő konyhóig, és próbálgassuk bele a három kulcsot a liftvezérlőbe. Amelyik sikerült, annak a színe megváltozik. (Ha nem sikerült, akkor csak letesszük a kulcsot, de ne felejtjük el visszavenni!) Majd jöjjünk visszafelé, ugorjunk fel a platóra, majd tovább jobbra a következőre, innen jobbra, majd balra fel a platókon. (A Denzil's Pad nevű helyre kell kilyukadnunk.) A jobb oldalon a két fa között lévő kerítés mögött egy coin van elrejtve, vegyük fel (17). Menjünk jobbra Daisy elhagyott konyhójához (Daisy's Empty Hut), és a jobb oldali ablak kerete mögött van elrejtve a következő érme (18). Menjünk balra egészen a liftig, majd menjünk fel vele, és a fenti plató bal széléről ugorjunk felfelé. Az öregekhez vezető liftnél (Lift To The Elders) vagyunk. Jobbra menjünk el a kopogtatóért (BRASS DOOR KNOCKER), majd sétáljunk vissza a plató legszélére, és innen ugorjunk balra, és itt vegyük fel a pénzdarabot (19). Itt a plató bal széléről ugorjunk egyet a semmibe. Egy felhőre érkezünk, melyen egy coin van (20). Menjünk vissza az öregekhez vezető liftig, és a lifttel menjünk fel. A jobbra lévő érmét felvéve (21) menjünk balra az öreg Dizzyhez.

- Jó napot kis Dizzy, kétségbeesettnek nézel ki. Talán tud neked segíteni a te jó öreg Dizzyd? - mondta az öreg mihelyt meglátta.

- Nem hallottad még, hogy Daisyt fogva tartják a felhőkastélyban? Meg akarom menteni! - panasolja Dizzy a történetek nagy keservesen.

- Várj meg itt! Máris veszem a kalapom. - indult el a papa, hogy tényleg felvegye a kalapot. De Dizzy utána szólott:

- Mi!? Kösz, hogy segítenél, de azt hiszem, jobb ha itt maradsz.

- Haddát, ha te így látod jónak ... de fogd ezt a feszítővasat. Emlékszem, mikor ... - adta oda a feszítővasat, és kezdett bele fiatalkori emlékeinek mesélésébe, melyet a eme tárgy juttatott eszébe. (Egyébként olyan szenilis, mint egy vénember. Pedig még csak 126 éves!-EPY)

Vegyük fel a feszítővasat (STRONG CROWBAR) és a balra található rozsdás csákányt (RUSTY OLD PICKAXE). Ezután menjünk vissza a kúthoz (Oda, ahol a zsák volt, aminek segítségével már öt tárgy lehet nálunk.), és a feszítővassal törjük fel, majd essünk is le. Egy különös, új világba csöppentünk (Strange New World), minden a feje tetején áll. Menjünk jobbra a templomba (Inside The Church), szerezzük meg a coin (21) és a negyedik kulcsot is. Majd balra a piactéren (The Market Square) gazdagítsuk tovább az érmegyűjteményünket (22), és várjunk egy picit. Nocsak, itt van a Kincses Sziget közlekedési vállalatának egyik eladója! Úgy látszik, hogy már a műszaki cikkeik adásvétele nem sokat hozott a konyhára, most már zöldsegéskizletbe fogott. A tehénkéért cserébe ad is egy szem zöldbabot (SIMPLE GREEN BEAN). Menjünk vissza a liftvezérlőkhöz, és indítsuk be az utolsót is. Ezután a kopogtatóval menjünk vissza jobbra ahhoz az ajtóhoz, ahova ki volt írva: "Kopogj és lépj be!"

Kopogjunk, és lépünk be! Vegyük fel itt az üres vödört (EMPTY BUCKET), majd vissza balra egészen a vulkán tövéig (Base Of Vulcano), majd a pocsolyában töltsük meg a vödört, és Dylanra hallgatva a trágyát megöntözve kinő az égig erős babfa. Ugráljunk fel a fán a felhőkastélyig. (Aki ezt egyedül szeretné megcsinálni, annak tudnia kell, hogy ütközben összesen három coin kell felvenni!) Tehát: Ugorjunk a legalsó, jobbra lévő levélre, majd balra, utána jobbra, menjünk ki a jobb szélére, ugorjunk felfelé, majd jobbra kell ugrani a felhő másik szélére. A jobb széléről (Complex Cloud Rate) ugorjunk a jobbra lévő felhőre, majd az érméhez (24). Innen a jobbra lévőhöz, majd felfelé, és a felhő bal széléről jobbra ugráljunk, majd a sok ugrálás után sétáljunk át a másik pályára (MORE ORRIBLE CLOUDS), és ugráljunk fel a coinért (25). Essünk vissza az alattunk lévő felhőre (ahonnan ide ugrottunk), és innen essünk le. Egy felhőre érkezünk a vulkán mellett (Near The Volcano), ahol a következő pénzdarab van (26). Majd essünk le, és ugráljunk fel oda, ahol a 25. coin találtuk. Essünk le jobbra a felhőre, ugorjunk jobbra, sétáljunk vissza egy picit, és ugorjunk ismét jobbra. Vegyük fel a sárkánytojást (HEAVY DRAGON EGG), és menjünk jobbra. Itt egy coin, de az utunkat állja egy fakirpad. Menjünk hát vissza, essünk le a felhőről. Menjünk vissza a liftvezérlő konyhóig, és a lift alatt essünk le, vegyük fel az érmét (27), és a lifttel menjünk le. A sárkánytojást tegyük le a másik mellé, mire a sárkány továbbneged. Menjünk jobbra, vegyük fel a pénzdarabot (28), majd a csákánnyal bontsuk ki jobbra a falat, az omláson túl vegyük fel a szőnyeget (OLD THICK RUG), és jöjjünk vissza. Jé, itt van a börtönőr! Azt mondja, most bújkál, mert mi kiszöktünk, és a király el akarja kapni. De ha ő most itt áll, akkor ki van a startpályánál? Menjünk hát vissza oda. Tovább most sem tudunk menni, de itt az utolsó előtti coin (29). Vegyük fel, menjünk vissza a felhőkastélyig, a fakirpadon terítsük le a szőnyeget, vegyük fel a legutolsó érmét (30!!!), majd a kapcsolóval eresszük le Daisyt Sikerült! Megmentettük Daisyt! Nézzük végig kettejük bájolgását, majd menjünk a Daisy konyhójához (Daisy's Empty Hut). Itt a vége, fuss el véle. Még elolvashatjuk a CODEMASTERS üzenetét:

Wow! Megmentetted őt! Hát, ez szinte semmiség volt. És most búcsúzzunk el a boldog pártól egy időre. Ki tudja, mi fog ezek után történni? Reméljük tetszett ez a játék! Ez mind az OLIVER fivéreknél köszönhető.

DIZZY III	
Grafika:	7
Zene/FX:	4
Játszhatóság:	6
Összehatás:	6

Tisztelt leendő Szerzőtársak!

Megszűntek a „szigorú” formai feltételek, írásaikat bármilyen formában (papíron, .WP, .TXT fájlként) el tudjuk fogadni. A programokat továbbra is lemezen/kazettán küldjék be!

Címünk: ENTERPRESS,

1399 Budapest, Pf. 701/334

Szerkesztőségünk

a lap szervezésével kapcsolatos teendők ellátására megbízható, agilis, budapesti, telefonnal rendelkező Enterprise

rajongót keres.

A jelentkezők leveleit postafiókunkba várjuk.

VENDETTA

A játék készítői, a SYSTEM 3 szoftverház már sok, igen színvonalas játékkal örvendeztetett meg bennünket, mint például a MYTH, illetve a LAST NINJA II. Ezekhez hasonlóan a VENDETTA is igen érdekes és szórakoztató élvezetet nyújt mindenkinek, aki veszi a fáradságot, a gépe mellé ül, és betölti a programot. (Aki nem szereti a 3D-s megoldást, az inkább be se töltse!) A játékot a 3D-s akciójátékok közé lehet sorolni, és sokban hasonlít az előbb említett LAST NINJA II-höz. Az alaptörténetről csak annyit, hogy főhősünk barátját elrabolták, és természetesen a játék célja az ő visszaszerzése, illetve a rosszfiúk halomra gyilkolása. (A cím jelentése egyébként "vérbosszú".)

Betöltés után kiválaszthatjuk a megfelelő irányítást. A fel, le, jobbra, balra, tíz gombokon kívül van még a [C] billentyű, amellyel az aktuális fegyvert választhatjuk ki (Ez a játék kezdetekor az öklünk és egy kis zsebkes áll a rendelkezésünkre.), valamint az [ENTER]-rel választhatjuk ki a használni kívánt tárgyat. Nyomjuk meg az [ENTER]-t és tartjuk folyamatosan nyomva. Ekkor a játék megáll, és a tárgyak képei mozognak a képernyő alján. Ha megvan az, amit ki szeretnénk választani, akkor engedjük el az [ENTER]-t, és nyomjuk meg a tüzgombot, ekkor ez a tárgy használatát fejtí ki. Tárgyat pedig a tüzgomb és a joy valamire való húzásával vehetünk fel. (No nem akkor, ha függőlegesen felfelé húzzuk a joy-t. Ez legfeljebb Internal joy-nál lehet hatásos, ha a célunk a bot levétele, vagy kitépése a helyéből.-EPY) Hát akkor talán kezdhetjük is a játékot, nyomjuk meg a [0] billentyűt.

1., Az ellenséges tábor szélén állunk. A tárgyaink helyén semmi sincs (NOTHING), a fegyverzetünk jelen pillanatban pedig az öklünk. Szemben velünk egy ajtó van, de zárva. Használjuk ki öklünk adta lehetőséget, és üssük bele egy nagyot. Az ajtó szétesett, szabad az út. Menjünk be. Egy szobában találjuk magunkat. A falon két géppityu lóg, hát szerezzük meg. De csak az egyiket tudjuk. Sőt, töltény híján még löni sem tudunk vele. Sebaj. A géppisztolyos fal előtt nem sokkal a földön valami furcsa heverészik. Vegyük fel. (A tárgyak felvétele nem a játék erőssége, ezért ha nem sikerül elsőre, ne csodálkozzunk. Úgy kell állni, hogy a tárgy már a hátunk mögött legyen, és eltakarjuk.) Egy feszítővasat találunk. Mit is kezdjünk vele. Mivel a szobában néhány láda áll, próbáljuk meg őket felfeszíteni. (Lásd tárgyhasználat!!!) Sikerült is, de sajnos semmit sem találunk. (Ha felfeszítünk egy ládát, az kinyílik, és akkor a felvétellel kutathatunk benne.) Menjünk ki, és jöjjünk el jobbra a ház előtt. A fal jobb szélén is egy tárgyat látunk. Vegyük fel, ami némi kis ügyességgel sikerül is. Ez bizony egy fogó. Majd menjünk tovább jobbra, másszunk fel a létrán, és menjünk vissza egészen a startpályáig. Itt egy szintén zárt ajtót látunk, de ezt nem tudjuk beütni. Nem is kell, hiszen amúgy is át tudunk jutni rajta. Egy újabb szobában találjuk magunkat. Az ágy mellett egy nyakláncot, a bal alsó (nem a legalsó) ládában pedig a puskához való töltényt találunk. Menjünk tovább jobbra két pályát, amíg egy számítógépet nem látunk. Itt egy térkép van a falon. Tárgyfelvétellel letéphetjük onnan, s mögötte egy másik térkép van egy a falba vájt üregben. Vegyük fel, ez a főnök menekülési útvonalát ábrázolja. Most menjünk a jobb szélső fiókos asztalhoz, húzzuk ki a fiókot (szintén tárgyfelvétel), és vegyük ki belőle a főnök számítógépkönyvét. (Melynek címe természetesen: ENTERPRISE User Manual!-EPY) Olvassuk el (használjuk) a könyvet, melyben az áll, hogy a lemezt a komputerbe kell tenni. (INSERT DISK INTO COMPUTER) Na igen, de még nincsen lemezzünk. Ezért menjünk tovább jobbra. Itt a bal felső asztal-fiókból vegyük ki a benne rejlő videokazettát, majd haladjunk tovább. Itt bal oldali fiókban megtalálhatjuk a diszket, a jobb oldaliban egy újabb videokazettát, a lentiek közül a jobb oldaliban pedig egy kulcsra leltünk. A szoba közepén a kor egyik vívmánya, egy videomagnó áll. Nézzünk bele a kazettákba. Az egyikben némi sisterség után megláthatjuk a barátónk képét (Na, tudod ki harcolna ezért a házisárkányért!-EPY), a másikon pedig némi sisterség után további sisterséget látunk. Menjünk a lemezzel vissza a számítógéphez, és használjuk a lemezt. A gép

kéri a fegyverkártya kódját. Hóha! Ezt honnan tudjuk meg? Megvan! Az ENTERPRISE kézikönyvet felapozva megtudhatjuk, hogy a kód: ZIP1 (WEAPON CARD CODE IS ZIP1) Tegyük be a lemezt ismét, és írjuk be a kódot. A számítógép kiadja a várva várt fegyverkártyát. Menjünk ki jobbra, másszunk le a létrán, és menjünk jobbra a kocsához. Néhány tánclépéssel álljunk az ajtaja mellé, és használjuk a slusszkulcsot. Egyből be is ülünk, majd ha a fegyverkártyát is használjuk, akkor el is tudunk indulni. Ennyi szaladgálás után még főhősünk is elfárad, ezért most autókázunk egy kicsit a tábor közepe felé. Először megnézhetjük a menekülési útvonalat, majd egyből az úton találjuk magunkat. Az ellenség természetesen ezt nem hagyja annyiban, helikopterekről lövöldöz bennünket, illetve kamikaze-ként próbál autókkal frontálisan nekünk jönni. Útközben kétfajta lövedékkel (levegőrakéták, puskagolyók) lövöldözhetjük az ellenséget. (Értelemszerűen az előbbi való a helikopterek, az utóbbi pedig a kamikaze-k ellen.) A fegyverek között az [ENTER]-rel válogathatunk. Ha sikerült elérnünk a célig, töltődik a következő szint.

2., A tábor kellős közepén vagyunk. Fent egy tank van, de nincs csöve. Ismét egy ajtó előtt állunk. Ha már egyszer sikerült szétörni egy ilyen, akkor most miért ne sikerülne?! Tehát üssük be az ajtót, és menjünk be. A falon most már komolyabb fegyvert találunk, egy páncéllökőt. (Nem holmi Kalacsnyikövet, amint az első szinten!-EPY) A ládák felfeszítésével pedig vegyük fel a töltényt, a páncéltörő rakétát, és a "golyóálló mellényt". (Ez utóbbit ne vegyék fel azok, akik sérthetlenséggel játszanak, mert akkor elkezdi fogyni az energiájuk!) Dolgunk végeztével jöjjünk ki, és menjünk jobbra, egészen a tetőre vezető létráig. Itt másszunk fel, és menjünk jobbra. A ládát fesszítsük fel, és ürítsük ki a tartalmát. Egy kulcs van benne. Másszunk vissza, és az ajtón át menjünk be a már mániává váló szétzúzás után. Itt egy női válltáskát, egy újabb térképet és egy papírlapot találunk. Majd jobbra továbbmenve a bal felső sarokban az ágy mellett egy női magassarkú cipellőt találunk. Menjünk vissza az autónkhoz. Ho-hó! A tanknak csöve nőtt! (Biztosan meglocsolták!-EPY) Állítsuk a fegyvert a páncéllökőre, a tárgyat pedig a páncéltörő töltényre. Töltsük meg az öklöt, majd lövjük ki a tankot. Most már nyugodtan átmehetünk előtte az autónkhoz. Álljunk mellé, és használjuk az első pályán már használt slusszkulcsot. Ezek után a változatosság kedvéért ismét egy kis autózgatás következik, de itt már motoros ellenségek is a ellenünk rohannak.

3., Most már a reptéren vagyunk, ahonnan a főnök a lánnyal együtt próbál menekülni. Ezt mi nem hagyjuk! Menjünk be a szemben lévő ajtón a hangárba. Itt a második szobában egy kis munícióna lelünk, majd menjünk be az ajtón. A szekrényekben található tárgyakat (sérthetetlenek a mellény kivételével!!!) vegyük fel, majd menjünk vissza az autóhoz, és menjünk jobbra, amíg el nem érjük a repülőgépet, melyen a barátónk van. Az utastérben rá is találunk a kis südren. Lövjük le a mellette ülő férfit, majd a csőfőfogóval álljunk a lány háta mögé, és szabaddítsuk meg köteleiketől. Vegyük fel a térképet is innen, majd ismét autókázunk tovább - csak először menjünk vissza a kocsához.

4., Ez a szint már nem követel logikát, csak egy Kalacsnyikövet, no meg némi töltényt. Haladjunk csak egyenesen előre, s közben lövjük le azokat, akik ártani próbálnak szerény személyünknek. Ha sikerült végigérni, akkor a program gratulál, és közli, hogy visszaszereltük a kis sárkányt. (Nem LITTLE PUFFOT, hanem a barátónket!-EPY)

VENDETTA	
Grafika:	6
Zene/FX:	3
Játszhatóság:	5
Összhatóság:	6

Postafiók 334

Igen részletes, több témakört felölelő levél írására serkentette Czeglédi Miklós dunajvárosi olvasónkat az az örvendéses esemény, hogy végre kézhez kapta az ENTERPRESS hiányzó számait: "A Mátix-ékat külön szeretném megdicsérni, utánvét helyett símán, egyszerűen elküldték a kért számokat, mindössze egy "piros" utalványt mellékeltek... (Persze, reggel postára adjuk a gubát...) ... Lelkesítő és jó, hogy rögtön olyan dolgokra találtam, amiken régen töröm a fejem, és csak részben tudtam megoldani. Ilyen volt pl. a szalagos fájlok átrása lemezre... Azután az Assembly sorozatról: kitűnő, ehető, és most, hogy több rész együtt van már, lehet előre-hátra lapozgatni, értelmezni. Őszintén megvallom, együtt olvasom, ill. mindig összehasonlítom egy C16/C+4-re írt könyvvel, így azután - és ezt dicséretnek számom - a megismerés öröme a fáradság ellenére "üzemben" tart. Hiába, nem vagyok már gyerek... sajnos. Van egy - lehet, hogy neveletlen - kérdésem is: hozzájutottam egy GM-6-os egerhez; tudom, hogy a vele adott .COM fájl EP-n nem fog menni, de úgy gondoltam, hogy a soros csatlakozón, valamilyen manipulációval, csak be lehet idomítani valahogy. Tehát tisztelettel kérem a véleményüket a dologról, mert azt kezdem sejteni, hogy ha az EP-hez készült egy a JOY csatlakozóra illik, akkor az nem egy igazi soros eszköz...

Nos, még egy dolog nyomja a bögyömet, és ez a "nepperek" kérdése. Elismerem, hogy a tisztesség az egy úri dolog, és akinek pénze van, az nyugodtan lehet úr akármelyik (akár PC) csárdában is. Na, de aki csóró? Én most pl. ezt a levelet egy ilyen úton beszerzett Wordstar-on írom. Ugye? Honnan lenne nekem dBase 2-m, Multiplanom, ha eddig nem volt bennünk, irthoniakat megcélzó, és nagyon helyesen mondják: relatíve olcsó szoftverpiac? És ha nem lennének ezek a programjaim, hogyan tudtam volna ötvenhét éves fejjel olyan PC-felhasználóvá válni, aki teljes természetességgel használ olyan dolgokat, mint az ORCAD/SDT, vagy a QUATTRO?

Én azt hiszem, hogy ez is egy olyan dolog, mint a lengyel piac: az élet alakította ki, és ha a feltételek javulnak, a fene sem fog egy kipróbálatlan, garancia nélküli fűrógépet megvenni... Szerintem a hobbigépeket inkább úgy kéne kezelni, mint egyes nagy szoftverházak az egyetemeket, pl. hogy oktatási célokra szinte ingyen bocsájják rendelkezésre - egyébként drága - termékeiket, azal a nem útköztől cédal, hogy a nebulók majdani munkahelyükön is ki fogják követelni azokat, annyira megszokják a kényelmes szolgáltatásokat, amiket egy-egy ilyen profi program nyújt.

Őszintén örülünk annak, hogy Önnek ilyen örömet szerzett a lap. Örülünk kiadónkat illető dicséretének is, természetesen átadjuk. Nem kis részük van abban, hogy ez a lap egyáltalán létezik... (Más "vállalkozó" cégek nem képesek erre.)

Igyekszünk a teljes Enterprise-felhasználói tábor egészének hasznos, érdekes, új, használható információval szolgálni. Ez nem könnyű, ha arra gondol, hogy ez a kör a csak puff-puff-ra éhes srácoktól a programozás és egyéb felhasználási területek iránt is érdeklődő fiatalokon és idősebbeken át egészen a megszállott, egy spritekezelőt, grafikus demót vagy Basic bővítést a kisujjukból kirázni képes profikig terjed... Ami az egyiknek természetes, az a másiknak unalmas vagy túl bonyolult. Ennek fényében köszönjük dicséret sorait.

Az egerhez adott .COM file, ami az eger meghajtóprogramját (driverét) tartalmazza, természetesen csak azon a gépen (illetve operációs rendszeren) működik, amelyikhez készítették. Jól sejti azt is, hogy az Enterprise egere nem soros eszköz, az tulajdonképpen egy "cincogó" botkormány: amíg a mozgatójuk, úgy viselkedik, mint egy joystick. Ez mindjárt elárulja korlátait is: egy igazi eger nem csak a mozgás irányát, hanem annak sebességét is képes továbbadni a számítógép felé, mégpedig igen széles tartományban. Ha az eger soros illesztésű, természetesen lehet vele kezdeni valamit egy soros porton át. Azonban a dolog nem ilyen egyszerű. Az Enterprise soros portja ugyanis nem "igazi" soros port abban az értelemben, hogy nem autonóm működésű: a soros funkciót teljes egészében programozott módon valósították meg, így a port használata igen sok odafigyelést igényel a processzortól. (A PC-kben a soros port kezelése külön I/O processzor - 8250 - feladata.) Ez annál is inkább bosszantó, hogy az eger kezelése egyébként is sok ravasz húzást igényel, az eger-kurzor a felhasználói programtól gyakorlatilag teljesen

függetlenül "él", mozog a képernyőn. A feladat természetesen megoldható, de nem kimondottan kezdőknek ajánlott. Érdemes igazán jól, profi módon megcsinálni, különben csak nyűg van vele. Nem kizárt, hogy létezik egy-két megoldás Enterprise-ra is. Értesüléseink szerint a Devilsoft csapata már megoldotta a problémát!

A nepperek... Igen, a tisztesség úri dolog, és akinek pénze van... Nos, szerkesztésünk egyik tagja "civilben" egy "komoly" programtermék értékesítésével foglalkozik. Cége a programot Amerikában védelem nélkül árusítja, Nyugat-Európában már kénytelenek egyfajta védelemmel ellátni a jogtalan másolás ellen. Ugyanezt a programot Kelet-Európában feltörik, és még jó, ha csak amúgy, szfvességből adják tovább, és nem árulják az eredeti ár töredékéért. A programban eddig közel száz programozó-évnnyi munka fekszik, és egy csapat folyamatosan dolgozik a további tökéletesítésén, miközben a konkurrencia is dolgozik. Nem adhatják a programot olcsóbban, hiszen százmillió nagyságrendű költségük van vele évente.

Nincs olcsó program, csak két esetben: ha nagyon-nagyon sokat adnak el belőle - ez leginkább csak az amerikai piacon lehetséges -, vagy ha a programot... lopták. A tolvajnak nincs vele költsége, vagy ha mégis, az csak töredéke annak, amit a fejlesztő, a forgalmazó rákötött.

E sorok írójának egyik ismerőse is találkozott egyik Enterprise-os programjával egy "programkereskedő árjegyzékében". Tíz forintra volt értékelve, mint az összes többi program. "Ha a program fejlesztésére "elvesztegetett" három hónap alatt valami mást csináltam volna, legalább 30-40 ezer (akkori) forint tiszta hasznom lett volna!" - mesélte rezignáltan az illető, aki keresett műszaki fordító. "Ha én tíz forintért árulom, legalább 3-4 ezer példányban kellett volna eladnom ahhoz, hogy nullszaldós legyek. Mindeki tudja, hogy ez képtelenség! De hát tíz forintért még lemásolni sem ér meg egy kazetát! Kedvem lett volna felkeresni a "kereskedőt", és megkérdezni, hogy kitől vásárolta meg a program terjesztési jogát. Meglepődött volna, ha tíz- vagy húszezer forint kártérítést kért volna..."

Azt hiszem, hogy a szoftverbitorlás "enyhébb" esetével, azaz a program jogosulatlan használatával még egy darabig együtt kell élnünk Magyarországon. Hiába a törvény, ha az emberek többségének valóban nincs húsz-harmincezer forintja egy-egy programra. A megoldást Amerikából lehet ellesni: akkor szűnik meg ez a jelenség, ha majd nem éri meg lemásolni egy programot, hiszen ki akar vele kézikönyv nélkül bajlódni, ha a havi jövedelme töredékéért bármikor, bárhol megkapja? Ez azonban nem holnap lesz...

A bitorlás "minősített esetét" azonban, amikor valaki fillérekért kótyavetyéli más munkájának eredményét, amit, mondjuk ki nyugodtan, egyszerűen ellopott, nos, ezt az ENTERPRESS szerkesztésége eddig sem fogadta el, és a továbbiakban sem fog ehhez segítséget adni. Mi nem leszünk bűnrészesek, még akkor sem, ha több száz forintnyi (sic!) hirdetési bevételtől esünk el emiatt.

A felelős szerkesztő

GENERAL	
New	
Open	
Load	
Save	CTRL-S
Save as...	
Directory	
Shell	
Quit	CTRL-Q

„A külső is számít” c. írásunk ábrája

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) 16-520/280),

- 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 részsein gépi kódú rutinokat (CODE sorok, ROM hívás) is szabad használni, ezt azonban nem szabad túlzásba vinni! A legjobb programot közölni fogjuk.

Beküldési határidő: 1992. március 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

Apróhirdetések

Új nyomtató ENTERPRISE kábellet 8500 Ft-ért eladó
Bozai Gábor, 8000 Székesfehérvár, József A. u. 70/a. fax. 1. Telefon: (22) 10-665

Eladó ENTERPRISE 128-as angol EXDOS-asai, floppy támogatással, buszhoztovítókártyával, programokkal, játékkal, alkalmazói programleírásokkal, könyvekkel.

Érdeklődni: Varga Zoltán, 5600 Békéscsaba, Wlassich sétány 13. III. em. 11.
Tel.: (06-66) 27-493

ENTERPRISE géphez EPROM-égetőt valamint Shugart 410-es ósmekétdőzelemet vennék.
Cím: Bors Norbert, 2890 Tata, Bacsó Béla u. 66/18.

AKCIÓ - április 31-ig! Új, másolásvédtől APOLLOSoft programok extra árengedménnyel, csak nálunk! APOLLO-ASSEMBLER, APOLLO-SOUNDTRACKER, SAMPLEMAKER, HQ-SOUNDDIGITAL, APOLLO-9 speed-COPY, INV-II Appgame kazettával mindössze 280 Ft! (6 program+kazetta) MAGUK NEMÉBEN A LEGJOBBAKI
Az akció keretében az ajándék egy IBM kompatibilis számítógép. A nyertes nevét a III/3-as ENTERPRESS-ben közöljük.
APOLLO Stúdió, Bóta István, 5350 Tiszafüred, Pf. 87.

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

HIRDETÉSFELVÉTEL

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

MÁTRIX Kft.

ENTERPRESS
8000 Székesfehérvár,
Honvéd utca 8.

Megjegyzés: a nem saját fejlesztésű szoftverek másolásával foglalkozó üzletek 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, Ányos u. 1/3. Tel.: (80) 29-493 ifj. Márföldi Béla, 4400 Nyíregyháza, Új u. 28.
Tóth István, 7305 Mecsekpölöske, Szeffő u. 1.

KLUB

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

*** ENTERPRESS Kéthetente az Enterprisse számítógépek felhasználóinak * III. évfolyam 1. szám * 1992. január-február *** Kiadja a MÁTRIX Kft., Székesfehérvár * Felelős kiadó: Juhász István ügyvezető * A kiadó címe: MÁTRIX Kft., 8000 Székesfehérvár, Zichy liget 10. Telefon:(22) 16-520/280 Telefax:(22) 11-585 *** Felelős szerkesztő: Ujjkai László * A szerkesztőség tagjai: Hajnal Csaba főszerkesztő, Devlecoff, JÓVI, Ari Sándor, Bozai Gábor, Haluska László, Lolassoff, 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 HELIUS, 1900 Budapest, vagy a MÁTRIX Kft. címén * Előfizetési díj egy évre 294 Ft, fél évre 147 Ft *** Következő számunk márciusban jelenik meg. *** Az ENTERPRESS-ben közreadott információk célja az, hogy segítsék, tudnivalókkal lássák el a gép felhasználókat. A közölt programokat, kapcsolási rajzokat, leírásokat mindenki szabadon felhasználhatja, de tilos azokat a kiadó írásbeli engedélye nélkül másolni, terjeszteni. * A szerkesztőség kizárólag nem őriz meg, és nem küld vissza; továbbá külön levelezésre nincs módja. *** ENTERPRESS © 1991 MÁTRIX Kft. ***