

# ENTERPRESS

KÉTHAVILAP AZ ENTERPRISE SZÁMÍTÓGÉPEK FELHASZNÁLÓINAK

## VL-1772: a legnagyobb átvágás!?

Közismert, hogy az ENTERPRISE lemezvezérlő kártyája a WD-1772-es vezérlő IC-vel működik. Ezt a típust még kb. két évvel ezelőtt közel 2000 forintért lehetett beszerezni – igaz Budapesten csak egy helyen, a MIKRONIKA nevű cégnél. Nem tudni miért, de az utóbbi két évben teljesen eltűnt a WD-1772, vagy ha hozzá is lehetett jutni, akkor 3-4 ezer forintért vesztegették. Ez jelentősen megdrágította a lemezvezérlő kártya árát. Ezután jött a fordulat! Felbűnt a piacon egy ún. VL-1772-es IC, amely a forgalmazók szerint teljesen kompatibilis a WD-1772-essel. Rögtön hozzáteszem, hogy az ára 3500 Ft + ÁFA. Szegény ENTERPRISE felhasználó elhiszi, hogy a VL ugyanolyan mint a WD... ..de amikor oda kerül a sor, hogy egy lemezt kell formázni, akkor meredt szemekkel néz a képernyőre, ahol ezt a feliratot látja pár sáv formázása után: Foglalt...

Elkezdtek tesztelni a VL-1772-est. Formáztunk EXDOS-al, EPDOS-al, az eredmény ugyanaz volt. A legnagyobb meglepetés akkor ért minket, amikor a ZozoTools 1.8-as FAFO programjával formáztunk! A formázás sikeresen lezajlott és a lemez is tökéletesen használható volt! Féltreértés ne essék, nem a ZozoTools programot dicsérjük agyon ebben a cikkben – bár ezt is tehetnénk –, hanem egy rejtélyes hibára (?) világítunk rá. Nem tudom, hogy minden VL-1772-es így viselkedik-e. Mi már három ilyen VL-el találkoztunk, kipróbáltuk több lemezvezérlő kártyán is, de az eredmény ugyanaz volt: csak a FAFO gyorsformázó program segített. Így hát csak azt tudom tanácsolni, hogy kinek VL-1772-es IC van a lemezvezérlő kártyájában és ilyen problémákkal találkozott, az írjon a szerkesztőség címére és mi ingyen (a legrövidebb ha-

táridővel, – most bizonyára sokan felhördülnek – tényleg pár nap alatt!) elküldjük neki a FAFO lemez verzióját. Az biztos, hogy ismét átvágták az EP-s tábor! Vagy nem? Ezt a kérdést a hardveres kollégáknak teszem fel. Ha tudnak rá válaszolni, következő vagy akármelyik ENTERPRESS-ben közölhetik az erre vonatkozó írásukat.

**Most pedig néhány szó a nagy port kavart programküldő szolgálatról.** Bizony sokan joggal panaszkodtak, hogy novemberi megrendeléseiket még áprilisban sem kapták meg.

Igen, jogos a Tisztelt felhasználók felháborodása, de azt hiszem, ha mindenki komolyan vette volna az 1993-as első számunkban megjelentetett kérdőívet, ahol a lehetőség adva volt arra, hogy megírják: „csak egy-két programmal rendelkezem”, „vagy nincs egy programom se az ENTERPRISE gépemhez”, akkor mi nem 12 összeállítással indultunk volna, hanem csak négygyel! A 12 összeállítás listái egy-két kivétellel ősrégi programokat tartalmaznak. Itt Pesten sokan „hörögtek” a listákat látva: „kinek kellenek

majd ezek a régi, már jól ismert programok?”. Kinek? Sorolom: több mint 300 db kazetta, 240 db 5,25"-os lemez, és 150 db 3,5"-os lemez, 63 egyéni kérés. Minderre volt 7 emberünk, így hát nem is csoda, hogy „belebuktunk az első fordulóba!”

**Ennek ellenére folytatjuk! Megnyugtatóképpen elmondom, hogy a második fordulóttól, azaz mostantól Tóth István, szerkesztőségi tagunk foglalkozik egy személyben a programküldő szolgálattal. Tehát Tisztelt Olvasóink az ő címére küldhetik a megrendeléseiket, sőt telefonon is megrendelhetik a kért programokat! A vállalási határidő 1 hét lesz, ezt nagyon komolyan be fogjuk tartani! Kérjük olvassák el a 14. oldalon lévő tájékoztatónkat.**

Megköszönve türelmüket és megértésüket:

Matusa István, felelős szerkesztő

## TARTALOM ♦ 94/2.

PGCOPY – rendszerbővítő	2
Gépi kód kezdőknek (II. rész)	4-5
Az EDC WINDOWS (I. rész)	6-7
SRAM kártya	8
Az ENTERPRISE DOS lemezek felépítése III.	9-10
A dBase II. adatbáziskezelő II. rész	11-12
Leaderboard Golf I-II.	13
Programküldő szolgálat II.	14
BASIC programok láncolása	15
Hirdetések, információk	16

## PGCOPY – már van jobb!

Még emlékszem, mikor kb. 1-2 éve a budapesti ENTERPRISE klubban szinte mindenki a PGCOPY programmal másolt. Sorozatunkat a PGCOPY másolóprogram ismertetésével folytatjuk.

A PGCOPY-t betölthetjük lemezzel, de EPROM-ba égethető formában is létezik. Miután behívtuk a programot lefoglal egy hatalmas RAMDISK-et (a megkérdezésünk nélkül). A státusz sorban az aktuális meghajtó betűjelét, utána pedig a kijelölt fájlok számát láthatjuk. Itt látható a RAMDISK mérete és a PGCOPY 2.6 felirat. A státusz sor alatt egy ablakban a felhasználót segítő parancsbillentyűk jelentése található. Sorrendben: ERASE – fájl kijelölés-törlés (az összes fájl kijelöli illetve a kijelölést megszünteti.) Egyébként egyenként a SPACE vagy ENTER billentyű lenyomásával jelölhetünk ki fájlokat. Következő funkció a ZIP. (Valószínű tömörítést hajt végre). Nekünk nem sikerült rájöttünk a használatára. Mint később kiderült, használatához a VENUS rendszerbővítőre is szükség van... Kiadhatunk EXOS parancsokat a kettőspont lenyomása után. REN (R betű) – fájl átnevezése. Szöveges fájlokat kétféle formában is megnézhetünk a T illetve a 2-es billentyűk lenyomása után. Az F8-as billentyűvel a RAMDISK-et törölhetjük. (A státusz sorban semmi nem változik, miután töröltük a RAMDISK-et!) DEL – a D billentyű lenyomása a fájl vagy a kijelölt fájlok törlését eredményezi. ATTR – (A billentyű), a fájl attribútumát állíthatjuk (Hidden, Read Only). A SHIFT+ESC kombinációval léphetünk ki a programból. H-val kérhetünk HELP-et (ha lenne!). A perjellel T-ki vagy T-be között válthatunk (mi lehet ez?).

A munkaablak alatt látható: F1 – DIR (directoryt kérhetünk), C – Copy (másolás, ez történhet magnóra is), F3 – CD (belépés egy könyvtárba), ESC – ki: minden elindított parancsból az ESC-el léphetünk ki. Lehet állítani a billentyűismétlést is 1-9-ig. A jobb alsó sarokban a lemezen lévő fájlok számát, a kijelölt fájlok bájtszámát illetve a lemezen még szabadon levő üres hely méretét láthatjuk.

**Összességében:** a PGCOPY tudásban elmarad a PG-család többi tagjától! Sok hiányosság található a programban és nem igazán felhasználóbarát – gondolok itt a parancsbillentyűk programozására. Miután kilépünk a PGCOPY-ból a RAMDISK-et nem törli és ez zavaró. (Talán mindjárt az elején ha nem nyitna olyan baromi nagy RAMDISK-et, az elviselhetőbb lenne).

Ha valaki szeretné megrendelni a PGCOPY-t, a következő címen teheti:

Haluska László, 1086 Budapest,  
Karácsony Sándor u. 18. 3/41.

Ára: 200 Ft + az adathordozó, vagy EPROM ára.

Matusa István

## Új EPROM-ba égethető programok

**EPCKMB.ROM – Méret: 32K. Szerzői díj: 400,- Ft**

**A következő programokat tartalmazza:**

### EPROM version 1.1

EPROM-égető program

*Felhasználás:* A Mészáros Gyula féle égető hardver kezelése. Fájlok editálása, vágása, egyesítése stb.

Jellemzők: 64K puffer, 2716–27512 epromok fogadása, 5–7 féle égető algoritmus. Lemezes rendszereknél kurzorral történő fájl kiválasztás. Parancsbevitelhez minimális billentyű használat.

### PCK version 1.0

Csatorna tömörítő

*Felhasználás:* Fájlok tömörítésére. A kezelést legegyszerűbben BASIC programmal lehet megoldani. A kicsomagolás nem automatikus.

### UPCK version 1.0

Csatorna kicsomagoló

*Felhasználás:* A PACK-kal csomagolt fájlok helyreállítására.

### FPACK version 1.0

Egy blokkos fájl-tömörítő

A FENAS MERGE parancsával egy programhoz memóriaterületeket csatolhatunk tömörített alakban.

(A HELP-ben megtalálható a kicsomagolás algoritmusai).

*Felhasználás:* ASSEMBLY programozásnál igen hasznos. Lemezes rendszerrel auto-listázással történik a fájlválasztás. Az elkészült tömörített fájl \*.PDT nevet kap.

### PCK version 1.1

RAMbővítéses-lemezes rendszerhez

BATCH/H Játékprogram csomagoló

:PCK INP fájl OUTósvény

Játékprg = Betöltő&Hidden fájlok

Az elkészített fájl önindulós



**Az egészség  
csatornája  
a kábeltévéken**

**A Szív tv műsora az ország számos helyi  
és körzeti kábelhálózatán látható,  
több mint egymillió lakásban.**

*szórakoztatás, filmek, információ  
riportok, tréningek*

A Szív tv postacíme: 1656 Budapest, Pf. 6.  
Telefon: 256-6136 (fax is), 257-1270

T=-.TRN örökéletű indítás

A-E = Hova történjen a kicsomagolás

*Felhasználás:* A több fájlból álló programokat egyetlen, öninduló, tömörített programra csomagolja. A futtatás ramdiszkból történik. A tömörítés hatékonysága játékoknál kb. 40-80 százalék.

#### PP version 1.0

EPDOS 1.7-szelekt a PCK-hoz

*Felhasználás:* Lemezrendszer, EPDOS 1.7 és RAM-BŐVÍTÉS esetén a PCK parancs kényelmes kiadására.

#### DPP version 1.0

EPDOS 1.7-szelekt a \*.PCK fájlok kicsomagolásához

*Felhasználás:* EPDOS 1.7 esetén a \*.PCK fájlok kényelmes kicsomagolására.

#### FILE version 1.0

PUFFER: a 0. lapon lévő 256 bájt

ÖSVÉNY: megh:útvonalnévszűrő

OPCIÓ: H = hidden S = system

A 208-209-es csatornákat használja

A-E = meghajtó \.=könyvtár választás

A FILE-nevet a pufferbe teszi

*Felhasználás:* Lemezrendszer esetén bármely program (így BASIC is) alkalmazhatja autolistázásos fájlnev kiválasztásra.

#### KEYCH version 1.0

KEYBOARD csatorna kereső

Visszatérésnél D = csatornaszám

*Felhasználás:* Gépi kódú bővítlők alkalmazhatják a felhasználó által már megnyitott KEYBOARD csatorna számának lekérésére.

#### MB version 1.1

miniBANK adatbázis kezelő

*Felhasználás:* A miniBANK EPROM-os változata. Lemezrendszerrel autolistázásos fájl-kiválasztás.

#### PBOX.ROM - 32 K.

A PAINTBOX egérrel is kezelhető EPROM-os változata.

#### MONITOR.ROM - 64 K. Szerzői díj: 200 Ft.

*Tartalma:*

GEN version 1.1

MON version 1.1

MON5 version 1.0

DTEST version 2.3

FENAS version 1.2

#### PASZIANS.ROM - 64 K. Szerzői díj: 200 Ft.

*Tartalma:*

PA (PASZIANS)

KA (KASZINO)

64K-s EPROM megrendelés esetén (korlátozott mennyiségben) doboz + 64 K nyák is rendelhető.

*Megrendelhető az alábbi címen*

HALUSKA LÁSZLÓ, 1086 Budapest,  
Karácsony Sándor u. 18. 3/41.

© 1994. HSOFT

## CHIP-CHIP kódok

1 BDHP	2 JXMI	3 ECBQ	4 YMCJ	5 TQKB
6 WNLP	7 FXQO	8 NHAG	9 KCRE	10 VUWS
11 CNPE	12 WVHI	13 OCKS	14 BTDY	15 COZQ
16 SKKK	17 AJMG	18 HMJL	19 MRHR	20 KGFP
21 UGRW	22 WZIN	23 HUVE	24 UNIZ	25 PQGV
26 YVYJ	27 IGGZ	28 UJDD	29 QGOL	30 BQZP
31 RYMS	32 PEFS	33 BQSN	34 NQFI	35 VDTM
36 NXIS	37 VQNK	38 BIFA	39 ICXY	40 YWFH
41 GKWD	42 LMFU	43 UJDP	44 TXHL	45 OVPZ
46 HDQJ	47 LXPP	48 JYSF	49 PPXI	50 QBDH
51 IGGJ	52 PPHT	53 CGNX	54 ZMGC	55 SJES
56 FCJE	57 UBXU	58 YBLT	59 BLDM	60 ZYVI
61 RMOW	62 TIGW	63 GOHX	64 IJPQ	65 UPUN
66 ZIKZ	67 GGJA	68 RTDI	69 NLLY	70 GCCG
71 LAJM	72 EKFT	73 QCCK	74 MKNH	75 MIDV
76 NMRH	77 FHIC	78 GRMO	79 JINU	80 EVUG
81 SCWF	82 LLJO	83 OVJP	84 UVEO	85 LEBX
86 FLHH	87 YJYS	88 WZYV		

Kiss L.

## ÚJDONSÁGOK • INFORMÁCIÓK

Javában készül az EPDOS 2.0-ás verziója. Ha HSOFT befejezi a programot, természetesen részletesen közöljük lapunkban az EPDOS 2.0-át. Annyit elárulhatunk, hogy az ENTERPRISE történelmében ez a szoftver hatalmas fordulópont lesz!

♦ ♦ ♦

EXDOS kártyák turbósítását, SPECTRUM emulátorok átalakítását vállaljuk. Jelentkezni lehet a szerkesztőség címén. Részletesebb információk előző lapszámainkban találhatóak.

### Felajánljuk megvételre a következő MACINTOSH konfigurációt:

APPLE MACINTOSH LC számítógép,  
4 MB RAM, 40 MB Winchester,  
APPLE 12" RGB monitor,  
Personal Laser Writer NT (APPLE) nyomtató,  
MICROTEK SCANMAKER 600 ZS  
színes szkennel.

Érdeklődni: VMH Kereskedelmi Kft.  
6000 Kecskemét,  
Deák Ferenc tér 3.  
Fejes Ferenc

Telefon: 76/482-232 • Telefax: 76/328-214

## Gépi kódú programozás kezdőknek – II. rész

Sorozatunk második részében – mint ahogy előző számunkban ígértük – először megtanuljuk, hogyan lehet a hosszbjájtot egyszerűbben meghatározni. Az előző számban lévő példaprogramban így határoztuk meg a hosszbjájtot:

```
LD A,1
LD BD,(HOSSZ)
LD DE,ESC1
EXOS 8
...
HOSSZ DW 15
ESC1 DB 27,"o"
      DB 27,"C",255,0,255,180,0,0,0,0
      DB 27,"I",0
```

A hosszbjájtot egyszerűbben meghatározhatjuk a következőképpen:

```
LD A,1
LD BC,HOSSZ
LD DE,ESC1
EXOS 8
...
ESC1 DB 27,"o"
      DB 27,"C",255,0,255,180,0,0,0,0
      DB 27,"I",0
HOSSZ EQU $-ESC1
```

Mint láthatjuk a HOSSZ címenévnél történt a változás. Az EQU \$-ESC1 utasítás megnézi, hogy az ESC1 címke hány karakterből áll és automatikusan meghatározza a hosszbjájtot, megkönnyítve munkánkat. Azért tanultuk meg az előző variációt is, hogy könnyebben megértsük a hosszbjájt meghatározását. (Vö.: Ha valakinek az IBM DOS parancsait akarjuk megtanítani, ne a NORTON Commander-t tanítsuk meg neki először, mert soha nem tanulja meg a lemezkezelést!)

Most pedig folytassuk. Előző programunkban a kiírás után egy HALT-ciklussal várakoztattuk a programot, azért, hogy lássuk: mit írunk ki a képernyőre. Most egy praktikusabb módszert alkalmazunk. Megtanuljuk a billentyűzet figyelését, azaz a képernyőn való megjelenítés után a program egy billentyű leütésére vár. Hogy ne legyen unalmas sorozatunk, most nem írni fogunk a képernyőre hanem rajzolni, mondjuk egy kört. Utána várakoztatjuk a programot addig, míg egy billentyűt nem ütünk le. Először szokás szerint lássuk BASIC nyelven a programot:

```
100 SET VIDEO MODE 1
110 SET VIDEO COLOR 0
120 SET VIDEO X 30
130 SET VIDEO Y 20
140 OPEN #1:"video:"
150 SET #1:PALETTE 203,0,0,203,0,0,0,0
```

```
160 DISPLAY #1:AT 1 FROM 1 TO 20
170 PLOT #1:600,300,
180 PLOT #1:ELLIPSE 120,120
190 DO
200 LOOP UNTIL INKEY$<>""
```

Ugyanezt a programot láthatjuk listánkon gépi kód-ban. Lépjünk be az ASMON programba és az E billentyűvel lépjünk be az EDITOR-ba. A lista begépelése után az F8-as billentyűvel lépjünk ki. A Z billentyű leütése után végezzük el a szokásos beállításokat:

```
Assembly Listing ON
List conditions NO
Force Pass 2 NO
Memory assembly NO
Object file name: TAN2.COM
EXOS module header YES
EXOS module type: 5
```

Az opciók beállítása után nyomjuk meg az A betűt. Elkészült második gépi kódú programunk, amit természetesen – ahogy múltkor is leírtuk – bárholnan betölthetünk.

### Az új utasítások magyarázata:

```
LD DE,BILL
LD A,3
EXOS 1
LD A,3
EXOS 5
```

Először a BILL címenévre hivatkozunk (BILL DB 9,"KEYBOARD:") A KEYBOARD 9 karakterből áll, tehát a DB után hosszbjájtunk 9-et kell írunk. Ezután megnyitjuk billentyűzetcsatornának a 3-as csatornát (LD A,3 és EXOS 1), majd az EXOS 5 funkcióhívás következik, amely a billentyűzetcsatornáról (3-as csatorna) egy karakter leütésére vár. Az EXOS 5 funkcióhívás jelentése: karakterolvasás.

**DB 27,"A"**

Grafikus sugár pozicionálása.

**DW 600,300**

Itt lesz a kör középpontja.

**DB 27,"E"**

Kör rajzolása, majd megadjuk a kör méretét a

**DW 120,120**

utasítással.

Természetesen programunkban már a hosszbjájt a cikk elején ismertetett módon használjuk. (HOSSZ EQU \$-ESC1).

*(Folytatjuk)*

*Matusa István*



SET	ORG 100H	; Fordítási cím
	MACRO @VALTOZO,@ERTEK	
	LD B,1	; Írás
	LD C,@VALTOZO	
	LD D,@ERTEK	
	EXOS 16	; EXOS változó olvasása, írása vagy átbillentése
	ENDM	; Makró vége (END Macro)
	LD SP,3FFFH	; STACK beállítása (veremmutató)
	.SET 22,1	; 1-es videómód
	.SET 23,0	; 2 színű üzemmód
	.SET 24,30	; 30 oszlop széles
	.SET 25,20	; 20 sor magas
	; Csatorna megnyitása	
	LD A,1	; A-ba a csatornaszám
	LD DE,VID	; DE a név előtti hosszújtra mutat
	EXOS 2	; Csatorna megnyitása írásra
	; Videolap megjelenítése	
	LD B,1	; Alfunkció kódja 1=videolap kijelzése
	LD A,1	; A-ba a csatornaszám
	LD C,1	; Az első kijelzendő sor az 1-es
	LD D,20	; 20 sort kell megjeleníteni
	LD E,1	; Az első sorban kezdődik a megjelenítés
	EXOS 11	
	; ESCAPE sorozatok kiírása	
	LD A,1	; A-ba a csatornaszám
	LD BC,HOSSZ	; BC-be a blokk hossza (A HOSSZ címen lévő érték)
	LD DE,ESC1	; DE-be a blokk kezdőcíme
	EXOS 8	; Blokk kiírása
	; Várakozás billentyű leütésre	
	LD DE,BILL	; DE a név előtti hosszújtra mutat
	LD A,3	; A-ba a csatornaszám
	EXOS 1	; Csatorna megnyitása
	LD A,3	; A-ba a csatornaszám
	EXOS 5	; Karakterolvasás
	JP VISSZATERES	; Ugrik a befejezésre
VISSZATERES	LD A,1	; A-ba a csatornaszám
	EXOS 3	; Csatorna lezárása
	LD DE,BASIC	; Visszatér a BASIC-be
	EXOS 26	; Bővítések vizsgálata
VID	DB 6,"VIDEO:"	; A megnyitáshoz szükséges név
BILL	DB 9,"KEYBOARD:"	; A megnyitáshoz szükséges név
BASIC	DB 5,"BASIC"	; BASICcímké
ESC1	DB 27,"C",203,0,0,203,0,0,0,0	; A paletta színeinek beállítása
	DB 27,"A"	; Rajzolásugár pozicionálása
	DW 600,300	; A kör középpontja
	DB 27,"E"	; Kör rajzolása
	DW 120,120	; A kör méretének megadása
HOSSZ	EQU \$-ESC1	; Hosszújt meghatározása
	END	; Program vége

Fizessen elő a

**RÁDIÓTECHNIKA**

és a

**hobby**  
**elektronika**

folyóiratokkal így biztosan hozzájut!

Címünk: 1374 Budapest, Pf. 603.

A szerkesztőségben regisztrált HE előfizetőknek díjmentes nyák-film melléklet.

## AZ EDC WINDOWS

### BEVEZETŐ

Az ENTERPRISE felhasználók egyik régi vágya egy olyan grafikus felhasználói felület, amelynek használatával kényelmesen, esztétikus környezetben képesek dolgozni, ahogy az sok más típusú számítógépen látható.

Próbálkozás több is volt, közöttük nem egy profi munka is akad, de ezek mind leragadtak egyedi programok kiszolgálásánál. A végső megoldást egy szoftver teremtette meg, ami bármilyen alkalmazás számára képes ilyen felületet biztosítani, idegen szóval „menedzselni”. Ez az EDC Windows.

Az EDCW teljes egészében grafikus környezetben fut. A felhasználó a rendszert egy a képernyőn szabadon mozgatható nyilacska (továbbiakban: pointer) használatával irányíthatja. A képernyőn megjelenő ablakokban láthatóak végrehajtható műveletek, az eredmények – azaz a felhasználó csak a legkritikább esetekben kényszerül a billentyűzethez nyúlni – az „élet” a képernyőn folyamatosan zajlik.

A felhasználók alapvetően lusták. Nem akarnak zavaros parancsokat, bonyolult paraméterezést megtanulni – az ő számukra az EDCW kiváló lehet.

Az EDC Windows terjedését csak egyvalami gátolja: Mind-ezidáig kevés alkalmazás készült hozzá. Jelen cikksorozatnak is az a célja, hogy minél több felhasználót ismertessen meg az „új módival” s mások is megpróbálkozzanak egy-egy EDCW alkalmazás fejlesztésével ebben a kényelmes, komfortos környezetben.

Nos, vágjunk bele!

Az EDCW 4.7 rendszerlemez a következő programokat tartalmazza:

- EDCW.EXT – Az EDC Windows rendszerbővítő modulja
- WTOOL.EXT – Segédprogram, az új KEYBOARD: eszközt és néhány parancsot tartalmaz.

### EDCW ALATT FUTÓ ALKALMAZÁSOK

- WBASE.WIN – Adatbáziskezelő
- TAPEMAST.WIN – Abszurd magnó-lemez másoló
- SECRET.WIN – Kalandjáték. Garantált izgalmas.
- TEST.WIN – Lemez meghajtó-sebességeteszt
- DIGITRAX.WIN – .MOD – .DTM zenemodul fordító
- ZXSCR.WIN – ZX Spectrum típusú képernyők feldolgozását megkönnyítő program.

### EGYEBEK

- Az összes forrásszöveg
- ROM változatok
- HELP file-ok
- Ajándék játékprogram: SHOOTING RANGE

### KONFIGURÁCIÓ

Az EDCW kényelmes használatához szükséges lemez meghajtó (E nélkül is működik, de több funkció nem érhető el és néhány program működése bizonytalan) valamint ajánlott a memóriabővítés is, mert a grafikus felület bizony „falja” a memóriát. Ez persze nem jelenti azt, hogy ne lehetne igazán jól alkalmazni egy alapképzés gépen is.

### AZ EDCW INICIALIZÁLÁSA

Miután az EDCW rendszerbővítő modulja betöltődött, a :HELP parancs kiadásával győződhethetünk meg róla, hogy valóban a memóriában van-e a rezidens kód. A rendszer elindításához az :EDCW parancs begépelése szükséges, amely után több paramétert is megadhatunk:

- /M – A bejelentkező képernyő elhagyása
- /E – Indítás után azonnal a Setup menü jelenik meg
- /X – Az Extensions menü jelenik meg
- /F – A File manager jelenik meg
- /Y – A System menü jelenik meg
- /S – Memória SETUP figyelmen kívül hagyása

A megadott paraméterektől függően bejelentkezik az EDCW, és néhány információt jelenít meg a rendszerről:

- Az EXOS és az EXDOS verziószámát
- A RAM nagyságát (saját teszt alapján)
- Ha van, a memória SETUP dátumát
- Felismeri az hardverórát (le is olvassa) és az egérkártyát, de ez utóbbit még nem kezeli (Reméljük hamarosan ezt is kezelni fogja! – a szerk.)
- Ha valamelyik porton ismeretlen eszközt talál, a port(ok) számát.

A képernyő legfelső sorában foglal helyet az EDCW státusz sora, ahol állandóan látható a szabad memória nagysága (Kbyteban és a teljes RAM százalékában is) valamint a billentyűzet állapotjelzői is.

A jobb alsó sarokban a rendszeróra kapott helyet, tőle balra jelennek meg a hibaüzenetek. A pointeret bármelyik botkormány-nal és az O, P, Q, A billentyűkkel irányíthatjuk. Az aktuális opció mindig inverzben jelenik meg. Ha ki akarjuk választani (aktivizálni) a SPACE, ENTER billentyűkön kívül használhatjuk a tűz gombot is.

### A BILLENTYŰZET FUNKCIÓI

Az EDCW-nek saját, közvetlen billentyűzet kezelő alprogramja van, mely rendelkezik néhány plusz funkcióval is:

F1 – O, P, Q, A, SPACE irányítás kikapcsolására szolgál. Ha kikapcsolt állapot van, a státusz sorban megjelenik a „Keyboard locked” felirat.

F2-F5 – Az aktuális menüt lehet velük új pozícióba mozgatni. Amíg az EDCW-ből ki nem lépünk, a menük az új helyükön fognak megjelenni. (Ha a SETUP-ot lementjük, később is)

F6 – Angol/Német billentyűzet között kapcsol át, azonban ezt kijelzés nem támogatja, így csak egy billentyű lenyomásával állapítható meg az aktuális mód.

CAPS+LOCK – SHIFT üzemmód. A LOCK lenyomásával megszüntethető.

ESC – Visszalépés. (Pl. az előző menübe)

Ha az opció szövegében aláhúzott karakter szerepel, akkor az opció azzal a billentyűvel is elérhető.

Miután így tisztáztuk az alapokat, következnek egy részletes leírás az EDCW beépített programjairól.

### A SETUP

Amikor az :EDCW parancs begépelésével elindítjuk az EDCW-t, az aktuális meghajtó gyökérkönyvtárában megkeresi a

SETUP file-t, az EDCW.STP-t és megpróbálja betölteni. Ebben a file-ban vannak azoknak a paramétereknek az értékei, amelyek az EDCW megjelenését, külsejét szabályozzák (színek, menük pozíciói, változók stb.). Ha ez a file hiányzik, alapértelmezésű jellemzőkkel fog futni a program.

#### A legtöbb változó a SETUP menüben állítható be:

Cursor sensivity HIGH/MED/LOW – A pointer érzékenysége  
Inverse screen – Invertálja a képernyőt, azaz megcseréli a papír és tintaszíneket  
Inverse menus – A menük inverzek lesznek a képernyőhöz képest  
Move screen – Kép mozgatása jobbra/balra  
Dec/Inc ink number – A tintaszín növelése/csökkentése  
Bold font – Megvastagítja a karaktereket  
Recall old font – Visszamásolja az EXOS karakterkészletét  
Actual menu inv. – Az aktuális menü inverz lesz a többihez képest („kipipálva”)  
Shadow – Árnyék ki/be  
Clear screen – Minden új menü kirakásánál képernyőtörlés ki/be  
Save SETUP – Kimentí az új EDCW.STP-t, így az EDCW legközelebbi elindításakor is ugyanazokkal a paraméterekkel jelentkezik be.  
Make MEMORY SETUP – Nem lemezen, hanem a memóriában hozza létre a EDCW.STP-t, így gyorsabb a betöltés. A Setup EPROM-ba is égethető.

Ha van MEMORY SETUP, az EDCW meg sem kísérel lemezzről betölteni, hacsak meg nem adjuk a „/S” paramétert.

#### A FILE MANAGER

Ez a legösszetettebb beépített alkalmazás. Elindításakor egy nagy ablak jelenik meg, ahol egyszerre 20 file-t láthatunk, a méretükkel és attribútumaikkal. Egy műveletet úgy hajthatunk végre, hogy először a parancsra „kattintunk”, majd pedig arra a file-ra, amire vonatkozik. Az ablak alsó három sorában foglalnak helyet a parancsok:

Up/Down – görgetés fel/le  
CHDIR – Könyvtárváltás  
DiskInfo – Megjeleníti a lemez méretét, a szabad hely nagyságát, a formázó program nevét és a lemez azonosítóját.  
Dir – Újra beolvassa a lemezt  
Delete – File törlése  
Vol – Lemez átnevezése  
Drive – Egy almenüből kiválaszthatjuk a meghajtót  
Copy – Másolás. Először be kell írni a cél útvonalat, majd a másolandó file-okra kattintani. (Ha ENTER-t nyomunk, az utolsó útvonalat használja)  
Move – File mozgatás. Kezelése mint a COPY-é  
MakeDir – Könyvtár létrehozása  
Rename – Átnevezés. Először az átnevezendő file-t kell kiválasztani, majd beírni az új nevet.  
Select – Alapállapotban nincs funkciója. Ha valamelyik EDCW bővítés file betöltését igényli, ezzel választható ki a betöltendő file.  
Cmd – Parancsértelmező. A képernyő alján nyílik egy videolap, ahol az üzenetek megjelenhetnek.  
Macro – Macro Editor. Őt előre definiált parancs közül választhatunk, az EDIT opcióval átírhatjuk valamelyiket. (Használata nagyon hosszú parancsszerkezeteknél előnyös)  
RamDisk – Ramdiszk menü

RamDisk size: xxx – Ramdiszk mérete  
Open RamDisk – Ramdiszk megnyitása  
Delete RamDisk – Ramdiszk törlése  
Options – Almenü, a következő opciókkal:  
Verify – Ellenőrzés ki/be  
Hidden files – Rejtett file-ok listázása ki/be  
Dir Mask – A megjelenő file-ok maszkja (PI \*.\*)  
Load – File betöltése. A File Manager a következő file típusokat kezeli:  
EXOS fejlécek (5, 6, 7 stb.)  
EDCW rendszerbővítő modulok  
.DTF file-ok (A ZOZOTOOLS DL-re hivatkozik)  
.BAT, .INI file-ok (EXDOS parancsértelmezőt használja)  
.TXT, .DOC, .HLP, .ASM, .PAS, .BAK, .TPU, .ASC tokenizálatlan szövegfile-ok tartalmát megjeleníti.

#### AZ EXTENSION MANAGER

Hasonlóan a File managerhez, egy nagy ablak jelenik meg max. tíz a memóriában lévő EDCW bővítő nevével. A következő parancsokat használhatjuk:

Scroll up/down – Görgetés fel/le  
Extension info – Rövid információ egy bővítőről  
Delete extension – Bővítő törlése  
Activate – Bővítő aktivizálása

Az EDCW rendszerbővítői .WIN kiterjesztésűek, 64-es modulfejléccel. Ha egy ilyen alkalmazást betöltöttünk, az mindaddig a memóriában marad, amíg hidegindítást nem hajtunk végre, vagy ki nem töröljük (az Extensions menü „Delete extension” opciójával, esetleg ha van, a ZOZOSOFT féle EXOS 2.3-mal).

A rendszerbővítőket a File Manager „Load” opciójával kell betölteni (használhatjuk az EXDOS „LOAD” parancsát vagy az EPDOS-t is, persze csak akkor, ha az EDCW már a memóriában van – ilyenkor azonban gyakran megeshet, hogy lefagy a gép).

Ezekről az alkalmazásokról lesz még szó bőven, mivel itt kapcsolódhatunk be az EDCW „vérkeringésébe”.

#### EGYÉB ALPROGRAMOK

**System menü:**  
Restart – Újraindítja az EDCW-t  
Cold reset – Hideg reset  
Return – Óskorból maradt opció, az EDCW 4.7-ben leginkább lefagyást eredményez. (Eredetileg az EDCW-t 100h-n hívó programba tért vissza)

View status – Hasonló az AsMon ugyanilyen opciójához.

#### Néhány hasznos tanács:

Ha kevésnek bizonyulna a szabad memória, a rendszerlemez SOURCES könyvtárában található EDCW.EXT modult töltsük be, mert az nem tartalmazza WTOOL-t, így 16K felszabadul.

Ha gépelésnél problémák akadnak, kapcsoljuk ki az F1-gyel a billentyűzet irányítást, mert az gyakran „bekavar”.

A SECRET1.WIN és a SECRET2.WIN file-oknak egyszerre kell a memóriában lenniük, csak így indítható el ez a fergegetes kalandjáték.

(Folytatjuk)

© 1994. EDC



94/2.

#### A szerkesztők listája

Legjobb játék program:	SWAP
Legjobb felhasználói program:	ZOZOTOOLS 1.8
Legjobb demo program:	SMALL DEMO
Legjobb programozó:	HISOFT
Legjobb programátíró:	ZOZOSOFT
Legjobb szoftver stúdió:	HISOFT

## K Ó K O B A K

Ennek a címnek kevesebb az értelme, mint a legutóbbinak, de mégsem rugaszkodtam el teljesen a rögválóságtól; ahogy ígértem, az EPROM/SRAM-bővítő kártyáról fogok írni. Az EPROM pedig arról ismert széles a világon, hogy marha nehezen tanul (külön EPROM-égető készülék kell a beprogramozásához), igaz, amit egyszer megtanult, azt elég nehéz kiverni a fejéből; sokáig kell „napoztatni” (azaz ultraibolya fényvel bevilágítani az ablakán), amíg megfő az agya, mindent elfelejt. Írhatnám: buta lesz, mint a tők, és akkor lehet kezdeni a tanítását előről. De nem ezért szeretjük, hanem mert a számítógép kikapcsolása után is megmarad benne a nagy fáradtsággal beírt program, így az a gép következő bekapcsolásakor újra kéznél van. Ilyen Csak Olvasható Memóriába (angolul: Read Only Memory, azaz ROM) célszerű elhelyezni a gép operációs rendszerét, hiszen annak bekapcsoláskor azonnal üzembe kell lépnie: a legeslegelső utasítást valahonnan olvasnia kell a processzornak. De más programok is elhelyezhetők ROM-ban, különösen a jó öreg ENTERPRISE kínál könnyű lehetőséget kedvenc sakkprogramunk, lemezkezelőnk, szövegszerkesztőnk és egyéb gépnnyűző programjaink állandó készenlétben tartására és villámgyors előrántására, mindenféle lemezről, vagy pláne kazettáról (fűj!) történő betöltötés nélkül.

Hogy lesz a ROM-ból EPROM? Az E az, hogy Erasable, azaz Törölhető, a P az, hogy Programmable, azaz Programozható, tehát többször is fölhasználható, bár kissé macerás módon (lásd fent). Hogy megszabaduljunk az ősi magyar nyelvet előntő angolszász mételeytől, hívhatjuk a tárgyalat áramkört TPC-SOM-nak is, de nem árt, ha ezt először otthon, tükör előtt, majd szűk családi körben gyakoroljuk, csak ezután lépünk ki vele a nyilvánosság elé.

A kártya nevében szerepel még a SRAM betűszó is, amelyet két részletben fogok magyarázni. A RAM elnevezés bizonyára ismert, a számítógépek szabadon írható-olvasható memóriáját jelöli. Az ENTERPRISE-ban ebből 128 KB-ot találhatók, mégpedig Dinamikus RAM-ok, azaz DRAM-ok formájában. A „dinamikus” jelző azt jelenti, hogy a memóriááramkör állandó frissítést igényel, ami ha elmarad, az adatok elvesznek. Miért használják mégis az ilyen memóriákat? Mert olcsók, és kis helyen elférnek: egy bit tárolóáramkörre egyetlen tervezérlésű tranzistorból (MOS-FET) áll. Egy közönséges húszlábú keskeny IC-tokban elfér egy 4 Mbit-es memória, ami testvérek között is négyemillió tranzisztort jelent. (Körülbelül ennyi van a tényleg Pentium mikroprocesszorban.) Az ENTERPRISE-ban az említett frissítésről maga a Z-80, a RAM-bővítő kártyán egy pár IC-ből álló segédáramkör gondoskodik.

Na most térhetünk rá az S betűre, amely a „Sztatikus” szó rövidítése. Éppen ellentéte a dinamikusnak, így is fogalmazhatnánk, hogy normálisan működik: beírom az adatot, és benne van, bármikor kiolvashatom, persze amíg a gép be van kapcsolva. Hátrányai: (relatív) nagy, drága, és kikapcsoláskor felejt. Az első kettőn sajnos nem nagyon lehet változtatni. Egy 32 KB-os SRAM (62256) 28-lábú széles tokban lakik, ára boltban kb. 1200 Ft. nálam (használtan) 600 Ft. A felejtésen segít viszont egy megfelelően alkalmazott elem, vagy akkumulátor, ugyanis ezek a SRAM-ok 3 V tápfeszültségen olyan keveset fogyasztanak, hogy akár évekig is vidáman megőrzik az adatokat (persze ehhez jó minőségű akku kell). A SRAM egy-egy nyitóirányba kapcsolt diódán keresztül kapja a tápfeszültséget mind a géptől, mind az akkutól. Ha a gép be van kapcsolva, az ő tápfeszültsége magasabb, ezért a diódán keresztül ő táplálja a SRAM-ot, az akku diódája zárva van. Ha kikapcsoljuk a gépet, a tápfeszültsége leesik nullára, ilyenkor lép be az akku. A diódák arra kellene, hogy elválasszák egymástól az akkut és a gép tápfeszültségét. Egy ellenállás és egy harmadik dióda segítségével megoldható az is, hogy a gép bekapcsolt állapotában töltsse az akkut (különben használhatnánk elemet is).

Ezek után a fentiekből levezethető: az EPROM/SRAM-bővítő olyan kártya, amelybe EPROM-ot, ill. SRAM-ot tehetünk be, és

akkumulátorral is föl van szerelve a SRAM tartalmának megőrzése végett. A memóriákat vegyesen alkalmazhatjuk, összesen hat darabot egyszerre. Az EPROM-ok lehetnek 8K-tól 64K-ig jumperelhetően bármekkora, a SRAM célszerűen 32K-s lehet. Választhatunk 8 K-s SRAM-ot is, csak akkor nagyon kell vigyáznunk, mert az adott szegmensben belül kétszer, az alsó és a felső felében fog látszani egyszerre ugyanaz a memória. A 32 K-snál ez a probléma nem lép fel, hiszen az teljesen kitölt két szegmenst; itt csupán két másik gond lép fel. Az egyik, hogy hiába védjük akkuval a SRAM tartalmát, a gép bekapcsolásakor a memóriateszt szempillantás alatt kinullázza a drága jó előre feltöltött memóriánkat. A másik, hogy a kártyán minden IC egy 64 K-s címtartományban látszik, tehát a 32 K-s SRAM kétszer, ugyanúgy, ahogy a 8 K-sról fentebb írtam. Hát akkor mire jó ez az egész? - kérdezhetné az esetleg már nem is olyan nyájas olvasó.

A válaszhoz új bekezdést nyitottam, mert logikailag különálló egység következik. Ugyanis a hardver mit sem ér megfelelő szoftver nélkül, sőt, lehet, hogy sok szoftvermágus, wizard, és egyéb varázsló számára újdonságot mondok, de ez fordítva is igaz. Ezért felvettem a kapcsolatot két tűrhető szoftvercéllel, úgymint ZozoSofttal és HSofttal. (Nem rossz ajánlólévél, mi? Mármint az ő számukra, hogy nekem dolgozhattak...) Nem pont az én kártyám miatt, de a ZozoSoft-féle EXOS 2.3 nem teszteli azokat a szegmenseket, amelyekben rezidens rendszerbővítő talál, valamint kiszűri a kétszer szereplő szegmenseket is. Tehát megtehetjük, hogy közönséges RAM-nak használjuk a SRAM-ot, de ha beletöltünk egy EPROM-ba való programot, akkor az EXOS 2.3 ezt felismeri és nem írja felül. További előnye a programnak, hogy minden szegmenst végignéz rezidens rendszerbővítő után kutatva, így egy EPROM-ban, vagy SRAM-ban több programot is tárolhatunk. (Az eredeti EXOS 2.1 csak a nullára végződő hexadecimális című szegmenseket nézte át.)

HSoft a várva-várt (természetesen egérrel kezelhető) EPDOS 2.0-ban használja majd ki a 32 K nem felejtő RAM által adott lehetőséget. Nem akarok előre vágni a témának, ha publikus lesz, HSoft megírja, hogy mi minden beállítás, kezdeti értéket, és egyebet tárol majd az EPDOS 2.0 ezen a területen, ismereteim szerint lehetővé téve, hogy ott folytassuk a munkánkat, ahol előző alkalommal kikapcsoláskor abbahagytuk. HSoft memóriatesztelője is ismeri az EXOS 2.3-nál felsorolt trükköket, így teljesen kihasználhatjuk vele is az EPROM/SRAM-kártya előnyeit.

A kártya címdekódolásra az előző cikkben tárgyalat GAL16V8 áramkört használja. Alapvetően kétféle címzés létezik: a gyári EXOS 2.1, ill. az újabb memória-tesztelő programok számára. Az előbbi a 10h-13h, 30h-33h, 40h-43h, 50h-53h, 60h-63h, 70h-73h; az utóbbi a 60h-77h címtartományt foglalja el. (Természetesen már születtek egyéni változatok, minden szabványosítási kísérlet dacára.) Látható, hogy a második megoldás sokkal kompaktabb, sőt, némi változtatással lehetővé teszi az esetleges RAM-bővítő kártya megzavarása nélkül újabb EPROM/SRAM-bővítők behelyezését. A kártya spártaian egyszerű, hasonlóan az előző EPROM-bővítő kártyához: van rajta hat foglalat a típusbeállító jumperekkel, egy 74LS245-ös adatbusz-meghajtó, némi dióda, ellenállás, egy 7805-ös stabilizátor és a GAL16V8. Opcionálisan lítium-akkumulátor is rendelhető hozzá (700 Ft). A végére jön a csúf önreklám: ha valamely kedves olvasónak fölfelé keltette az érdeklődését a kártya iránt eme kis ismertető, az alábbi címhez fordulhat további információkért és/vagy megrendeléssel (szépen kérem, ha valaki kérdéseket tesz föl, és még választ is vár rájuk, mellékeljen egy felbélyegzett válaszborítékot... Köszöni!)

Mészáros Gyula

1029 Bp., Zsiroshgyi út 110.



## AZ ENTERPRISE DOS lemezek felépítése III.

### FISH hívások: a használt rövidítések:

STR	eszköz:\ösvény\ fájlnev .kiterjesztés
UV	útvonal (ösvény)
MEGH	meghajtó száma (1 = A, 2 = B, 3 = C, stb, 0 = aktuális)
FMEGH	fizikai meghajtó száma
KFCB	kereső fájl kontrol blokk
MFCB	meghatározott fájl kontrol blokk
TRC	transzfercím (felhasználói puffercím)
SIZE	bájtok száma (fájl vagy blokk méret)
SZEK	logikai szektorszám (kezdő szektorszám)
SZEKN	szektorok száma (mozgatandó szektorok)

### A hívás általános tudnivalói:

- IY a FISH változókra mutat. ("EXDOS", 0FDH-vel lehet lekérni.)
- 0. lap = nullás szegmens.
- 1. lap = tetszőleges.
- 2. lap = FISH RAM területet tartalmazó szegmens, általában a rendszerszegmens
- 3. lap = EXDOS ROM szegmens. (IY-5E)-ről olvasható.
- Verem a 0. vagy 2. lapon. (EXDOS 1.0 és E: esetén a 2. lapon)
- Transzfercím használat előtt be kell állítani a felhasználói szegmensregisztereket. (IY-5D) = P0 - (IY-5A) = P3
- A FISH hívás CALL 0C010H utasítással adható ki. A = funkció (1-30)
- Visszatéréskor NZ esetén A-ban található a hiba kódja.

### 01. Útvonal és KFCB készítés fájlnevből

DE = KFCB C = STR következő karaktere  
HL = STR DE = STR folytatása  
HL = UV (az STR-ben kialakítva)  
IX = KFCB (meghajtó és fájlnev kitöltve)  
B 0. bit = 1 ha a meghajtónéven kívül más kielemezett karakterek is vannak

1. ha volt specifikált útvonal
2. ha van specifikált meghajtónév
3. ha az utolsó elem file-név
4. ha az utolsó elem kiterjesztés
5. ha az utolsó elem többértelmű ('?' vagy '\*')
6. ha az utolsó elem speciális ('.' vagy ':')
7. ha az utolsó elem '.', csak ha 6. bit is 1

### 02. Aktuális katalógus megadás

HL = UV HL = UV

### 03. Aktuális katalógus lekérés

B = MEGH HL = UV (a FISH területen)

### 04. Első bejegyzés keresése

BC = MFCB BC = MFCB  
DE = KFCB DE, IX = KFCB  
HL = UV HL = UV (már a FISH területen)

### 05. Következő bejegyzés keresés

BC = MFCB BC = MFCB (kitöltve)

DE = KFCB DE, IX = KFCB

HL = UV (FISH területen)

### 06. Fájl létrehozás

BC = MFCB BC = MFCB  
DE = KFCB DE, IX = KFCB  
HL = UV HL = UV (FISH területen)

### 07. Katalógus létrehozás

BC = MFCB BC = MFCB  
DE = KFCB DE, IX = KFCB  
HL = UV HL = UV (FISH területen)

### 08. Fájl vagy katalógus törlés

DE = MFCB IX = MFCB  
HL = KFCB

### 09. Fájl vagy katalógus átnevezés

BC = MFCB (új névvel)  
DE = MFCB IX = MFCB  
HL = KFCB

### 10. Fájl vagy katalógus áthelyezés

BC = UV (új útvonal)  
DE = MFCB IX = MFCB  
HL = KFCB

### 11. Attribútum és idő változtatás

DE = MFCB (módosított attribútum/idő/dátum)  
HL = KFCB IX = MFCB

### 12. Fájl megnyitás

DE = MFCB DE, IX = MFCB  
HL = KFCB

### 13. Fájl lezárás

DE = MFCB DE, IX = MFCB  
HL = KFCB

### 14. Pufferek törlése

(Meghajtó adatterület lekérdezésre is használható)  
B = MEGH B = 0

(Ha B = 255, akkor minden puffert töröl)

C = 0 csak puffertörlés

HL = MEGH PTR (HL) = MEGH. puffer szegm.

(HL+1) = cím az 1. lapon

C = 255 pufferek törlése és érvénytelenítése

DE = MEGH terület az 1. lapon

HL = DE+9

### 15. Szekvenciális olvasás

BC = SIZE BC = sikeresen beolvasott  
bájtok száma

DE = MFCB DE = MFCB

IX = TRC IX = TRC a következő  
bájtra mutat

### 16. Szekvenciális írás

BC = SIZE BC = sikeresen kiírt bájtok száma

DE = MFCB DE = MFCB

IX = TRC IX = TRC a következő bájtra  
mutat

### 17. Szekvenciális írás 0 feltöltéssel

BC = SIZE BC = sikeresen kiírt bájtok száma

DE = MFCB DE = MFCB

IX = TRC IX = TRC a következő bájtra  
mutat

**18. Szektorok olvasása (logikai szektor)**

DE = SZEK DE = következő SZEK  
 H = SZEKN H = beolvasott SZEKN  
 L = MEGH L = MEGH  
 IX = TRC IX = TRC a következő  
 bájtra mutat

**19. Szektorok írása (logikai szektor)**

DE = SZEK DE = következő SZEK  
 H = SZEKN H = kiírt SZEKN  
 L = MEGH L = MEGH  
 IX = TRC IX = TRC a következő bájtra mutat

**20. Rendelkezésre álló meghajtók lekérdezése**

DE+HL = 26 bites szám amelyik bit 1 az a meghajtó létezik

L 0. bit = A

L 1. bit = B

...

L 5. bit = E

stb.

**21. Allokációs információ**

B = MEGH B = szektor/cluster

DE = cluster/lemez

HL = szabad cluster

IX = meghajtó adatterület+8

**22. Fizikai meghajtó lekérdezés (az 1.0-ás verzió nem kezeli)**

B = MEGH B = MEGH

C = FMEGH

**23. Fenntartva****24. Fenntartva****25. Mapdisk megadás**

B = MEGH BC = 0

C = FMEGH

**26. Assign megadás**

B = MEGH BC = 0

C = FMEGH

**27. Fenntartva****28. Ramdiszk létrehozás-törlés**

B = Szegmensek száma (0 = törlés)

**29. Formázás**

B = forma (F8-FF)

C = MEGH HL = MEGH adatterület

DE = puffterület (6500 bájtt)

**30. Fenntartva****KFCB: (használatba vétel előtt a területet nullázni)**

00 FF (keresési FCB)  
 01-05 0  
 06 Keresési attribútum (csak VOL-ra)  
 07 Meghajtó (0 = aktuális)  
 08-0F Fájlnév > (lehet ? is)  
 10-12 Kiterjesztés /  
 13-1C ?  
 1D-1E Logikai sektorszám  
 1F A katalógushoz tartozó szektorok száma: (DEC ciklusszámláló 1 = utolsó)  
 20 A szektorhoz tartozó fájlleírók száma: (DEC ciklusszámláló 16 = 1. fájl)  
 21-2B ?  
 2C-2F Lemez azonosító (ID szám)

**MFCB: (megtalálva)**

00 FE (meghatározott FCB)  
 01-05 0  
 06 Keresési attribútum  
 07 Meghajtó  
 08-0F Fájlnév (08-27 a beolvasott fájlleíró adatok)  
 10-12 Kiterjesztés  
 13 Fájl attribútum  
 14-1D  
 1E-1F Idő  
 20-21 Dátum  
 22-23 Kezdő cluster  
 24-27 Fájl méret  
 28-2B Fájlmutató  
 2C-2F Lemez azonosító (ID szám)

**MFCB: (megnyitva)**

00 FE  
 01-05 0  
 06 Keresési attribútum  
 07 Meghajtó  
 08-12 Fájlnév+kiterjesztés  
 13 Attribútum  
 14-16 0  
 17-1A Fájl meglévő mérete (SIZE)  
 1B-1C Dátum  
 1D-1E Idő  
 1F ?  
 20 ? (ha bit 6 = 1 akkor írás történt)  
 21-22 Kezdő cluster  
 23-24 Aktuális cluster  
 25-27 Cluster számláló (0 = első)  
 28-2B Fájlmutató  
 2C-2F Lemez azonosító

**MEGHAJTÓ ADATTERÜLET: (60H méretű)**

00 ? 0,2  
 01-02 Meghajtó RAM címe az 2. lapon (FISH IY)  
 03 Meghajtókezelő szegmensszáma  
 04-05 Meghajtókezelő címe a 3. lapon  
 06 ? (0 = létező meghajtó?)  
 07 Fizikai meghajtószám  
 08 Logikai meghajtószám  
 09 1-2 esetén csere ellenőrzés (64H>DEC 1/50 sec. időszámláló)  
 0A 0-1  
 0B 0-1 egy clusterben lévő szektorok száma  
 0C 1 > BOOT szektorok száma  
 0D 0  
 0E 2 > FAT példányok száma  
 0F 0  
 10 ROOT-DIR szektorok száma  
 11 FAT szektorok száma  
 12-13 0. cluster logikai sektorszáma  
 14-15 2. cluster logikai sektorszáma  
 16-17 A legnagyobb lehetséges cluster + 1  
 18 0  
 19-1C Lemez azonosító (ID)  
 1D Formázás típusszám

(Folytatjuk)

## A dBASE II ADATBÁZISKEZELŐ RENDSZER ISMERTETÉSE (II. rész)

### II. A RENDSZER PROGRAMOZÁSA

Programok készítésére a MODIFY COMMAND parancs szolgál. Ez után meg lehet adni a program nevét; ha nem adjuk meg, rákérdez. Amennyiben van már ilyen nevű program a lemezen, akkor azt betölti. Ha nincs, kiírja, hogy NEW FILE, és csak ezután jelentkezik be a beépített szövegszerkesztő.

(Megjegyzés: A program szövege normál szöveges állományként tárolódik, tehát nem kódolva, mint a BASIC programok.)

Minden utasítást külön sorba kell írni. Az első sor ne kezdődjön szóközzel! A többinél már nem számít, tehát a szöveget tetszés szerint tabulálhatjuk.

A beírás végén a programot a CTRL-W kombinációval tároljuk el, amivel ki is lépünk a szerkesztőből. Ezután a programot a DO <PROGRAMNÉV> paranccsal futtathatjuk.

Legjobb, ha az utasításokat néhány példaprogramon keresztül ismerjük meg. Az első program megnyit egy címet tartalmazó adatbázist, az első rekordra áll, beírat egy család- és egy keresztnévet a kezelővel, majd tárolja ezeket a rekordban.

```
use CIMEK index CSALAD
go top
store CS_NEV to CS
store K_NEV to k
@ 2,2 say "Családnév:" get CS
@ 3,2 say "Keresztnév:" get K
read
replace CS_NEV with cs
replace K_NEV with k
disp all
use
```

A programban a változónevek nagybetűvel, az utasítások kisbetűvel vannak írva.

Az első sorban megnyitjuk a CIMEK nevű adatbázist, ami a CSALAD indexállománnyal van indexelve. A "go top" utasítás az első rekordra áll. Ezután kezdőértéket adunk két változónak: a CS-ben a beolvasott családnévet, a K-ban a keresztnévet tároljuk majd. (Ha kezdőértéket nem adunk, a változó nem létezik, és futás közbeni hibaüzenetet kapunk.) A szöveges változókat nem kell a név után írni "\$" jellel megkülönböztetni, értékadáskor dől el, hogy a változó milyen típusú lesz.

Vegyük észre, hogy a változók kezdőértékeként tárolt neveket abból a rekordból olvassuk ki, amelyen állunk. (Feltételezzük, hogy a CS\_NEV a családnév mező, a K\_NEV pedig a keresztnév mező neve a rekordban.)

Arra mindenki rá fog jönni, hogy a következő utasítások mire valók. A "say" egy szöveget ír ki a képernyőre a megadott koordinátáktól kezdve. A "@" jel után egy szóközt kell hagyni! A "get" pedig, természetesen, a változók beolvasására való, azaz a BASIC INPUT utasításának felel meg. Önmagában azonban még nem olvas be semmit, csak elkészíti a képernyőmaszkot, amelyben annyi helyet hagy, amennyi a változó kezdőértékének hossza volt. A tényleges beolvasást a következő sorban a "read" utasítással kezdjük meg. Ilyenkor a rendszer lehetővé teszi, hogy a kezelő a képernyőmaszkba beírja az adatokat, az

egy mező között akármilyen sorrendben lépkedve. A bevitel végét az utolsó mezőben lenyomott ENTER jelenti.

Most látjuk értelmét annak, hogy az előbb a változóba a rekord mezőinek tartalmát tároltuk. Ugyanis a "get" utasítás a beolvasott változók eredeti tartalmát kiírja a képernyőmaszkba, így mindig látjuk, hogy mit módosítunk. Amikor lenyomjuk az ENTER-t, a rendszer mindig azt a szöveget tárolja el, ami éppen a képernyőn látható.

E tulajdonsága miatt a "get" beolvasás nélkül, csak a mezők tartalmának megjelenítésére is jó. Ilyenkor a kiírás után nem a "read", hanem a "clear gets" utasítást kell kiadni.

Az utolsó lépésben a mezők tartalmát kicseréljük (replace) a beolvasott változókra. (Az újonnan beírt nevet a rendszer az index szerint be is rendezzi.) Ezután ellenőrzésképpen kiíratjuk az összes rekordot (disp all) a képernyőre, majd lezárjuk az adatbázist.

Természetesen egy valódi programban bizonyos szempontok szerint kell kikeresni a módosítandó rekordot, és a mező tartalmának átirása előtt a kezelőnek biztonsági kérdést illik feltenni.

Nézzünk most egy "menükészítő" példaprogramot, amelynek segítségével a kezelő választhat a program szolgáltatásai közül. Tegyük fel, hogy a menüpontokat már kiírtuk a képernyőre (say).

```
accept "Válasszon!" to W
do case
  case w="1"
    do RESZ_1
  case w="2"
    do RESZ_2
  case w="3"
    do RESZ_3
  otherwise
    @ 10,2 say "Ilyen pont nincs!"
endcase
```

Nem nehéz felismerni a BASIC-ből már ismert CASE ágak megfelelőit. A RESZ\_1, RESZ\_2 és RESZ\_3 alprogramok nevei, amelyeket külön írtunk meg és a lemezen tároltunk. Az "otherwise" (angolul "egyébként") természetesen az ELSE ágnak felel meg. Az első sor "accept" utasítása kiírja a megadott szöveget a képernyőre, majd egy billentyű lenyomását várja, és azt tárolja a W változóban.

A következő program az összes rekord tartalmát egyenként kiírja a képernyőre, és vár, amíg a kezelő elolvassa azokat és lenyom egy billentyűt.

```
do while .not. eof
disp
accept "Ha elolvasta, nyomjon meg egy billentyűt!" to W
skip
enddo
```

Ciklusszervezést látunk, ami már szintén ismerős. Az első sor azt jelenti, hogy a ciklust addig kell ismételni, amíg még nem értük el (.not.) a file végét (eof = end of file). A relációs operátorokat (not, and és or) valóban két

pont közé kell tenni, nem sajtóhiba. Kírjuk az aktuális rekordot (disp), majd billentyűnyomásra várunk, végül a következő rekordra lépünk (skip). Egyébként várakozásra a rendszer a "wait" parancsot is ismeri, ami WAITING... üzenetet ír a képernyőre, de ezután más szöveg nem adható meg.

Végül ismerkedjünk meg a feltételes utasítással is. Írassuk ki az összes rekord tartalmát, kivéve azokat, ahol a családnév mező tartalma "Kovács". Az alábbi program (amelyet a HELP szövegből alakítottam át) ezt egy kissé csavarosan oldja meg, de működik.

```
do while .not. eof
  if CS_NEV="Kovács"
    skip
  loop
endif
disp
skip
enddo
```

Vagyis: ha a családnév Kovács, akkor kihagyjuk ezt a rekordot (skip), és visszaadjuk a vezérlést a "do while" utasításra (ezért ilyen ravasz a példa, hogy a "loop" használatát is be lehessen mutatni vele). Ellenkező esetben az "if" utasítás utánra kerül a vezérlés, ahol is kiírjuk a rekord tartalmát és csak utána lépünk a következőre.

*Kissé egyszerűbben ugyanez:*

```
do while .not. eof
  if CS_NEV<>"Kovács"
    disp
  endif
  skip
enddo
```

Természetesen az "if" utasításnak "else" ága is lehet. Hogy még mindig ennél a példánál maradjunk:

```
do while .not. eof
  if CS_NEV="Kovács"
    remark Kihagyva.
  else
    disp
  endif
enddo
```

Könnyű kitalálni, hogy a "remark" utasítás az utána következő szöveget kírja a képernyőre. Itt jó ezt használni, mert nem kell neki koordinátát adni, mint a "say"-nek. A szöveget nem kell idézőjelbe tenni.

*Még egy kis példa, kommentár nélkül:*

```
remark Nyomd meg!
wait
remark Na végre!
```

A "say" és a "get" utasítás után egyaránt használhatunk kírás mintát. Erre a két utasítás esetében két különböző kulcsszó szolgál, amelyeknek más-más mintát adhatunk meg:

@ <koordinatak> say <változó- vagy mezőnév>  
using "kép"

@ <koordinatak> say <változó- vagy mezőnév>  
get <változó- vagy mezőnév> picture "kép"

@ <koordinatak> get <változó- vagy mezőnév>  
picture "kép"

### "Using" után megadható paraméterek:

- X a mező/változó tartalma, vagy ha azon a helyen nincs semmi, akkor szököz.  
(Az X-et nagybetűvel kell írni.)
- \* és S a mező/változó tartalma, vagy ha azon a helyen nincs semmi, akkor ezek a jelek jelennek meg.

Bármely egyéb jellel a mező/változó tartalmát a képernyőn felülírja.

### "Picture" után megadható paraméterek:

- # csak számot, vagy szöközt fogad el.
- a csak betűt fogad el.
- X mindenféle karaktert elfogad.
- 9 csak számot fogad el.
- ! nagybetűsíti a beírt karaktert  
(szókezdéskor hasznos)

Bármely egyéb jel a kiírásban elválasztójelként jelenik meg, és ezeket a beolvasáskor a kurzor átlépi. Pl. telefonszámba tehetünk kötőjeleket, vagy a körzetszámnak zárójeleket: picture "+(###) - ## - ## - ##"

A programban használt memóriaváltozók értékeit bármikor lemezre menthetjük (save to <filenév>), majd visszaolvashatjuk (restore from <filenév>). A "release" utasítás pedig a változókat törli és az általuk használt memóriaterületet felszabadítja.

Ha nyomtatni szeretnénk, adjuk ki a "set format to print" utasítást. Ezután minden "say" a nyomtatóra fog írni, egészen addig, amíg "set format to screen" utasítással vissza nem állunk a képernyőre.

A memóriában egyszerre két adatbázis is lehet. Ezenkor a két adatbázis adatterülete között a "select primary" (elsődleges) és "select secondary" (másodlagos) utasításokkal lehet oda-vissza lépkedni. Változóterület csak egy van, tehát a két adatbázishoz nem rendelhetünk azonos nevű memóriaváltozókat.

## III. A BEÉPÍTETT FÜGGVÉNYEK

- chr(szám) azt a karaktert adja, amelynek ASCII kódja a megadott szám.
- int(szám) a megadott tizedesszám egész része.
- len(string) a karakterlánc hossza.
- trim(string) a karakterláncról a végén levő szökőket levágja.
- val(string) a karakterlánc számértéke.
- !(string) a karakterláncot nagybetűsíti.
- file(filénév) igaz, ha a lemezen a megadott file létezik, hamis, ha nem.
- type(mezőnév) a megadott mező típusa (karakteres, szám vagy logikai, azaz C, N vagy L).
- date() a rendszer dátumot adja meg.
- str(számértékű kifejezés, mezőszélesség, tizedesek száma) a megadott számot vagy a kifejezés értékét karakterlánc formára alakítja, ahol a mezőszélesség a kívánt karakterlánc összes szélességét, a tizedesek száma pedig ezen belül a tizedesjegy-helyek számát adja meg.
- rank() elvileg a zárójelben lévő karakter (vagy karakterlánc első karaktere) ASCII-kódját kéne hogy adja, de nem tudtam rájönni, miért csak SYNTAX ERROR-t válaszol makacsul.

Természetesen ahol a felsorolásban a zárójelben "string" szerepel, ez lehet szöveges változó, vagy szöveg típusú mező neve is.

(Folytatjuk)

Szalontai Andrea



## LEADERBOARD GOLF I-II.



## LEADERBOARD-I.

Az eredeti sporthoz jó állóképesség, kitartás, nagy figyelem, és sok pénz szükséges. Vagyis a felső 100000000000 sportja.

Szerencsére pompás gépünk segítségével mi is részesei lehetünk ennek a szűkkörű rétegnek! A számítógépes változat kifejezetten fejleszti az ülőképességet! Nem az ideges emberek feszültség-levezető játéka!!! Az első olyan játék, amiben minél kevesebb pontunk van, annál jobbak vagyunk. Ha már átlagban 0-ára teljesítjük a pályákat, akkor már indulhatunk a megyei bajnokságon, a nyeretlen két-évesek reményfutamán.

Mi a játék célja??? A lehető legegyszerűbb. Adott pályán, adott számú ütéssel kell a labdánkat a pálya végén levő lukba betereni. Ezt megtenni már nem is olyan egyszerű! Segítségre van mindehhez, külső, belső joystick és néhány billentyű. A joystick-ot jobbra, balra mozgatva, az ütésünk irányát állíthatjuk be (célkereszt). I.e. fel mozgatásával az ütőnk között válogathatunk. 13 ütőnk van. W jelzésű 1, 3, 5, I jelzésű pedig 1-9-ig. Van még egy PW jelölésű is amit a luk közelében használunk. Nagyobb számozású ütővel kisebbet tudunk ütni (angol logika).

A játék erdős területen kialakított pályákon zajlik, néhol tócsákkal, folyókkal, vagy mocsarakkal tarkítva. A pályák számát a játék elején mi választhatjuk ki. Ne legyünk túl mohók az elején! Ne ijedjünk meg, a barna színű víz nem környezetszennyezés eredménye, hanem csak a Spectrum-képernyő felépítésének köszönhető. Még mindig sokkal jobb mint mindig kék színű fákat bámulni!! (Lásd BAMSOFI átiratok!)

A W jelű ütőkkel a pálya jó talajáról üthetünk jó nagyokat, az I jelűekkel a lazább talajról is jól fel tudjuk emelni a labdát, pl. átemelhetjük a fák felett. Az ütés erősségét a tűzgomb nyomvatartásával határozhatjuk meg, ebben segít a képernyő alsó, jobb oldalán levő skála. A bal alsó sarokba a játékosok neve (1-4) és az ütések száma van kiírva. A képernyő bal felső részén a pálya száma, a jobb oldalon pedig, hogy hány ütésből kell teljesíteni! Törekedjünk negatív számot elérni! A távolban leszúrt karó jelzi

a keresett luk hollétét, ha idáig már eljutottunk, már csak be kell gurítani a lukba a labdát! Ügyesebbek és jó formában levők akár 5-6 inch-ről is mellé tudják gurítani. Miért??? Mielőtt még dührohant kapnánk, elárulom. Mert nem vízszintes a talaj! Hogy merre és milyen mértékben lejt azt elárulja a képernyő jobb alsó oldalán levő kis bot árnyéka! Ezt mind az irányzásnál, mind az erősség megválasztásánál figyelembe kell venni!!! Ha pl.: tőlünk elfele mutat akkor kisebbet kell ütni. A luk valódi helye csak a kirajzolás pillanatában látszik!!!

**Még néhány billentyű:**

V = Quit

T = térkép

P = az ütőerőt csökkenti (felezi?)

Még egy összefüggés, a bűvös szám kb. 21. Pl: 137 yardra vagyunk a luktól akkor a 8 I jelű ütővel üssünk, vagy pl: 170 yard esetén 4 I-vel próbálkozhatunk. A játék elején kell megválasztani az erősségi fokozatot:

KID  
AMATEUR  
PROFESSIONAL

Nekem a KID fokozat is megfelel mert akkor nagyjából oda megy a labda, ahová szeretném! (HA-HA-HA), (ez a ZOZO volt).

AMATEUR: fúj a szél, és a labda kedves ívben arra megy amerre akar.

PROFI: ugyanaz, csak a luknál a pálya lejtése sincs kijelezve.

A zenét F1-F2-vel kapcsolgathatjuk. Még egy jó tanács: LEHETŐLEG PÁRNÁZOTT SZÉKEN, és nyugalmas környezetben játsszuk.

## LEADERBOARD-II.

Csak az eltéréseket írom. Ez a játék a vízisportokat kedvelők igényeit is kielégíti. Kék víz, kellemes háttér, jó zene, mi kell még? Csak annyi, hogy ne a vízbe üssük a labdánkat, mert mehetünk utána. Aki nem tud úszni, az automatikusan kap másikat. A pályák ugyanis szigetekből tevődnek össze. Nincs térkép, viszont a játék elején nézőpontot(?) választhatunk. P, és V sincs, de egy RESET-tel a játék elejére léphetünk.

Kellemes kikapcsolódást kíván: **APUCI**

**LEADERBOARD-I.**

Grafika: 4

Zene: 10

Játszhatóság: 7

Az átirat minősége: 5

Összhatás: 7

**LEADERBOARD-II.**

Grafika: 9

Zene: 10

Játszhatóság: 9

Az átirat minősége: 10

Összhatás: 9

## Programküldő szolgálat II.

<p><b>13. lemez</b></p> <ul style="list-style-type: none"> <li>Jet Set Willy 1. 59</li> <li>Jet Set Willy 2. 61</li> <li>Monte Carlo casino 62</li> <li>Bounty Bob 64</li> <li>Quazatro 57</li> <li>Mordon Quest 65</li> <li>Hundra 56</li> <li>Human Torch 62</li> <li>Death Ride 31</li> <li>Xevious 1</li> <li>Time Gate 52</li> <li>Rampage 62</li> <li>Fire (csak lemezen!) 115</li> </ul>	<p><b>14. lemez</b></p> <ul style="list-style-type: none"> <li>Karnov 33</li> <li>Pang 58</li> <li>? Gauntlet III. 115</li> <li>Spy v. Spy 3. 30</li> </ul>	<p><b>15. lemez</b></p> <ul style="list-style-type: none"> <li>Dizzy 1. 35</li> <li>Dizzy 2. 35</li> <li>Dizzy 3. 4</li> <li>Dizzy 3,5 9</li> <li>Dizzy 4. 70</li> <li>Dizzy 5. 70</li> <li>Yolkfolk Dizzy 9</li> <li>Rapids 7,8</li> <li>Cybernoid 1. (TRN) 66</li> <li>Cybernoid 2. (TRN) 3</li> <li>Operation Wolf 8</li> </ul>	<p><b>16. lemez</b></p> <ul style="list-style-type: none"> <li>Ping-Pong (128!) 50</li> <li>Time Scanner 24</li> <li>Ramparts 43</li> <li>Savage 1. 30</li> <li>Savage 2. 33</li> <li>Savage 3. 29</li> <li>Boulder Dash 1. 63</li> <li>Boulder Dash 2. 30</li> <li>Boulder Dash 3. 21</li> <li>Mickey Mouse 46</li> <li>Madmix 69</li> <li>Dwarfs' Land (*)</li> </ul>
<p><b>Felhasználói - 4.</b></p> <p>EDC WINDOWS 4.9 (*) Csak lemezen!!!</p> <p>(Folyamatos fejlesztés alatt!)</p> <p><b>Upgrade lehetőség!</b></p> <p>Az összeállítás ára: 350 Ft.</p>	<p><b>17. lemez</b></p> <ul style="list-style-type: none"> <li>Swap (*) 53</li> <li>Myth 21</li> <li>Sim City 60</li> <li>Train 72</li> <li>Paperboy I. 32</li> <li>Paperboy II. 60</li> <li>Spindizzy 60</li> <li>Slightly 78</li> <li>Raw Recruit 7</li> </ul> <p>Az összeállítás ára: 350 Ft.</p>	<p><b>Felhasználói - 5.</b></p> <p>FENAS 1.2 (assembler, editor) + HELP (*)</p> <p>Hisoft Pascal IS-FORTH IS-LISP</p> <p>+ sok-sok rendszerbővítő és +3 DEMO (Nasa-Guy)</p> <p>Az összeállítás ára: 350 Ft</p>	<p>Az összeállítás ára: 350 Ft</p> <p><b>DEMO-3.</b></p> <p>ORK DEMO I. Doc-Show (128!) Somb 1-2. TNT-DEMO</p>

**Tisztelt Felhasználók!** Az első oldalon már olvashattak a programküldő szolgálatról. Nos alaposan megváltozott minden a programküldő szolgálatnál. Nagy örömeinkre jelentkezett a szerkesztőségnek a budapesti ENTERPRISE klub tagja Tóth István, aki egyszemélyben elvállalta a programküldő szolgálat bonyolítását. Újdonság, hogy telefonon is fel lehet adni megrendeléseket, így még gyorsabban lehet hozzájutni a kért programokhoz. A határidőt nagyon szigorúan betartjuk, ez maximum egy hét lesz! Arra kérjük a Tisztelt Felhasználókat, hogy az egyéni kéréseknél csak a listákon szereplő programokból rendeljenek. Lemez megrendelésnél kérjük feltüntetni a formátumot: 5,25" 720 K, 3,5" 720 K vagy 5,25" 360 K. Kazettás megrendelésnél az összeállítás számát kérjük felírni, pl.: „Megrendelem a 13. és 14. kazettát.” Sajnos rá kellett hogy jöjünk: a kazetta másolás nagyon időigényes, így 50 Ft-al megemeltük az árát. (Egy lemez másolása pár perc, egy kazettáé pedig jóval több és eddig mindkettő 150 Ft volt. A kazetták ára így iszonyú alacsony volt!) Az egyéni kéréseknél is kénytelenek voltunk +50 Ft-al emelni a kazetták árát.

A lemezek ára: 5,25" 720 K - 150 Ft/db,  
3,5" 720 K - 180 Ft/db,

A kazetták ára: 200 Ft/db, (EGYÉNI KÉRÉSEK ESETÉN: 250 Ft/db).

Ha valakinek 360 K-s meghajtója van annak egy összeállítást két lemezen tudunk elküldeni, így egy összeállítás ára 300 Ft (két lemezen).

Amelyik lemezen eredeti ENTERPRISE program található azt csillaggal (\*) jelöltük - ezek szerzői díjas programok. Ezeknek az összeállításoknak az árát a programok alatt tüntettük fel. A programokat utánvétellel küldjük. A megrendeléseket levelezőlapon, vagy telefonon várjuk. Egyéni kéréseket is ki tudunk elégíteni, de csak a listákon szereplő programokból lehet válogatni. Egyéni kérések esetén kérjük a programok nevét olvashatóan felírni, mellette pedig azt, hogy melyik lemezen szerepel. (Így megkönnyítik munkánkat!) Az EDC lemeznél az „UPGRADE lehetőség” azt jelenti, hogy aki ezt a lemezt megveszi, és az újabb verziószámú EDC-re is igényt tart majd, annak legközelebb csak a lemez árát és a postaköltséget kell majd kifizetni. A programok kiterjesztéseiről: .COM = eredeti program (nem örökéletes), .TRN = örökéletes, .ALT (alternatív) = eredeti vagy örökéletes formában is betölthető. **A következő számban újabb listákat adunk majd közre!** A 13. összeállításon a FIRE program csak lemezen működik, ezért ha valaki ezt kazettán rendeli meg válasszon helyette más programot a listákról.

A megrendeléseket levelezőlapon az alábbi címre várjuk:

Tóth István, 1173 Budapest, Újlak u. 9. IX/90.

Telefonon is megrendelhetők a kért programok az alábbi telefonszámon:

Tóth István, 257-1990.

## BASIC PROGRAMOK LÁNCOLÁSA

Az Enterprise memóriájában több programot is tárolhatunk egyidőben. Az egyes programokat 0 és 127 közé eső sorszámmal különböztetjük meg egymástól. A nullás számú program számára legfeljebb 42K, minden további számára legfeljebb 32K helyet biztosít a BASIC rendszer. Az EDIT <sorszám>, vagy EDIT<név> paranccsal határozzuk meg, hogy éppen melyik programot akarjuk szerkeszteni. Ennek számát a gép a képernyő felső részén ki is írja. Az éppen szerkesztett programot a sima SAVE és LOAD parancsokkal tudjuk elmenteni és betölteni, illetve a NEW-val törölni. (Ha valaki nem tudná: ha a program elején a PROGRAM "név" utasítást használjuk, akkor a SAVE után nem kell kiírni a nevet, mert tudja.) Ha az összes programot csoportosan akarjuk kezelni: a SAVE ALL parancs az összes programot az első neve alatt tárolja, a LOAD parancs az így tárolt programokat egyben visszatölti, a NEW ALL parancs pedig az összes, memóriában lévő programot törli.

Lehetőség van rá, hogy az egyes programok egymást indítsák, így láncot tudunk létrehozni. Az utasítás formája:

```
CHAIN <sorszám> (paraméterek)
vagy:
CHAIN <név> (paraméterek)
```

Paramétereket csak akkor kell adni, ha a másik programnak adatokat akarunk átadni.

A láncolásra a legegyszerűbb példa:

```
0. PROGRAM:
10 CHAIN 1

1. PROGRAM:
10 PRINT "Hello!"
```

Ha a 0. programot elindítjuk, azonnal meghívja az egyes számút, amely végrehajtja a PRINT utasítást. Amikor a futás befejeződik, az egyes program területén maradunk (lásd a számot a képernyő tetején).

Az alábbi példában a paraméterátadást figyeljük meg:

```
0. PROGRAM:
10 PROGRAM "nulla"(a,b,c)
20 PRINT "Nulladik prog"
30 PRINT a,b,c
40 CHAIN 1 (1,2,3)

1. PROGRAM:
10 PROGRAM "egy"(a,b,c)
20 PRINT "Egyes prog"
30 PRINT a,b,c
40 CHAIN 0
```

Mivel a két program kölcsönösen indítja egymást, végtelen ciklust kapunk, és a képernyőn, amíg a STOP billentyűvel le nem állítjuk, az alábbi feliratok fognak futni:

```
Nulladik prog
0 0 0
Egyes prog
1 2 3
```

Magyarázat: az egyes programot a=1, b=2 és c=3 paraméterekkel hívtuk meg, az 0 számára tehát a, b és c értéke ennyi. A nulladik programot paraméterek nélkül hívtuk meg, ezért az 0 számára az a, b és c változók értéke nulla. Ha most az egyes program 40. sorába paramétereket is írunk, pl:

```
40 CHAIN 0 (4,5,6)
```

akkor ezt kapjuk:

```
Nulladik prog
0 0 0
```

```
Egyes prog
1 2 3
Nulladik prog
4 5 6
Egyes prog
1 2 3
Nulladik prog
4 5 6
Egyes prog
1 2 3
```

és így tovább. Az első indításkor a nulladik program paramétereinek még nincs értéke, amikor azonban az egyes program hívja meg, már van.

A CHAIN hatására a meghívott program mindig az elejétől indul! Ezért sajnos „szubrutinként” nem használhatunk egy másik programot, mert annak végrehajtása után, ha visszaaduk a vezérlést a főprogramnak, az újra előlről fog indulni, és nem a CHAIN utasítás utáni sortól. Ha azonban a nulladik program pl. csak egy menüt ír ki, akkor már lehetőségünk van a tárban lévő programok közül aszerint meghívni egyiket vagy másikat, hogy a kezelő melyik menüpontot választotta.

Ha a hívott programot nem az elejétől szeretnénk indítani, megtehetjük, hogy paraméterként adjuk át annak a sornak a számát, vagy eljárásnak a nevét, ahonnan a meghívott programot futtatni kell. Például:

```
0. PROGRAM:
10 CHAIN 1("puff")

1. PROGRAM:
10 PROGRAM "egy"(A$)
20 SELECT CASE AS
30 CASE "piff"
40 call piff
50 CASE "paff"
60 call paff
70 CASE "puff"
80 call puff
90 END SELECT
```

Nevet azért jobb paraméterként átadni, mint sorszámmat, mert ha az egyes program szerkezetén változtatunk, és a sorszámmal megváltoznak, arról a nullásnak nem kell tudnia. Megjegyzendő, hogy az egyes program indulásakor a változók mindenképpen törlődnek (ahogy egy normál RUN, vagy START után is), egy esetleges előző futásból tehát semmi nem őrződik meg.

Szalontai Andrea



SZERVEZÉSI,  
SZÁMÍTÁSTECHNIKAI  
és KERESKEDELMI Kft.

Alaplapok, RAM-ok, modulok, floppy-k, Winchesterek, kontrollerek, házak, monitorok, hálózati tartozékok, billentyűzetek, mouse-ok, kábelek, nyomtatók valamint komplett gépösszeállítások 1 év garanciával.

Appli-COMP Kft.

Üzlet:

X. ker. Állomás u. 27. (Kőbánya városközpont)

### A funkcióbillentyűk használata dBASE-ben

A dBASE parancsai meglehetősen hosszúak, még akkor is ha rövidítjük őket.

Az EPDOS felhasználók egy parancsfájlban könnyen definiálhatják a funkcióbillentyűket a gyakrabban használt parancsokra:

FKEY 1,"USE "  
 FKEY 2,"DISPLAY "  
 FKEY 3,"DISPLAY STATUS",13  
 FKEY 4,"DISPLAY MEMORY",13  
 FKEY 5,"MODIFY COMMAND "  
 FKEY 6,"SELECT PRIMARY",13  
 FKEY 7,"LIST FILES LIKE +.",13  
 FKEY 9,"APPEND",13  
 FKEY 10,"DISPLAY FILES",13

FKEY 11,"DISPLAY STRUCTURE",13  
 FKEY 12,"LIST",13  
 FKEY 13,"INDEX ON "  
 FKEY 14,"SELECT SECONDARY",13  
 FKEY 15,"QUIT"  
 DBASE  
 EPDOS

A fenti listát a WP-ben gépeljük be, és a PRINT paranccsal (F3) mentjük ki BAT kiterjesztésű fájlként (Pl.: DB.BAT).

Az F8-as funkcióbillentyűt ne használjuk, mert a dBASE nem szereti.

Végezetül egy megjegyzés a dBASE ismertető első részéhez: A 2.41 (német), és a 2.41/H (magyar) verzióban működik a DISPLAY MEMORY.

Kiss László



## ENTERPRISE SZÁMÍTÓGÉPEK JAVÍTÁSA.

*Fólia billentyűzet-csere  
 beszereléssel: 2120,- Ft*

1133 Budapest, Kárpát u. 7/a.  
 Telefon: 149-7909

Nyitva: hétfőtől-csütörtökig, 9-17<sup>h</sup>  
 Pénteken: 9-15<sup>h</sup>

### APRÓHIRDETÉSEK

Enterprise számítógéphez 3,5"-os vagy 5,25"-os lemez-egység ízléses dobozban, tápegységgel eladó 11 000 Ft-ért. Ugyanitt floppy-vezérlő kártya 512 Kb bővítési lehetőséggel: 512 Kb RAM-mal 12 000 Ft, 0 Kb RAM-mal 10 500 Ft. *Érdeklődni:* Faragó Gyula, Telefon: 136-7400

◆ ◆ ◆

Eladó ENTERPRISE 128, beépített PC billentyűzet, színes monitor, 720 K floppy, egér, joystick, programok, szakirodalom. Esetleg külön is.

Szalontai Andrea, Tel.: 178-5342

A géppel kapcsolatos témákban levelezne:

Kókai György, 2116 Zsámbok, László u. 80.

T BOY'S ENTERPRISE KLUB  
 2803 Tatabánya, Pf. 3026

PÉCSI ENTERPRISE KLUB

Tagtörzsi: Kurta László,  
 7636 Pécs, Fagyóngy u. 10. fsz. 3

\* \* \*

Vidéki ENTERPRISE klubok  
 jelentkezését várjuk a szerkesztőség címén.

### 1994. január 1-től választóriték kötelező!

Akik a szerkesztőség részére kazettán vagy floppy-lemezen küldik el programjukat leírásukat, vagy kérdéseikre választ szeretnének kapni, küldjenek olyan méretű választóritéket, amelyben vissza tudjuk küldeni az adathordozót, vagy válaszolni tudunk a kérdésekre.

Köszönjük!

### HIRDETÉSFELVÉTEL

Az apróhirdetések ára: 2 Ft karakterenként. A szöveget és a befizetést igazoló nyugtát (rözsaszínű postautalványon) az alábbi címre kérjük feladni:

AMEKO Kft.  
 ENTERPRESS

1094 Budapest, Tüzoltó utca 96.

Megjegyzés: a nem saját fejlesztésű szoftverek másolásával foglalkozó üzletkötők hirdetésait nem áll módunkban elfogadni.



Budapesti ENTERPRISE klub

Budapest, VIII. kerület, Puskin u. 4.

\* \* \* Minden héten kedden 17 órától 20 óráig \* \* \*

ENTERPRESS – kéthavilap az ENTERPRISE számítógépek felhasználóinak. † V. évfolyam 2. szám. † Kiadja az AMEKO Kft., Budapest. Felelős kiadó: Kovács Gábor ügyvezető. † A kiadó címe: AMEKO Kft. 1094 Budapest, Tüzoltó utca 96. † Alapító főszerkesztők: Ujlaki László és Hajnal Csaba. † Felelős szerkesztő: Matusa István. † A szerkesztőség tagjai: Szalontai Andrea, Kulcsár Tibor, Haluska László, Mészáros Gyula, Zozosoft & Apuci, EDC, Tóth István. † A szerkesztőség csak levélben érhető el! A cím: ENTERPRESS, 1399 Budapest, Pf. 701/334. † Olvasószerkesztő: Tóth Sándor × Technikai szerkesztő: Vincze Györgyné. † Készült az AMEKO Kft. nyomdájában, Felelős vezető: Kovács Gábor. † Előfizethető az AMEKO Kft. címén. † A lapot csak előfizetés útján lehet megrendelni! † Előfizetési díj 1994. évben: fél évre 250 Ft, egy évre 500 Ft. † A közölt programokat, kapcsolási rajzokat, leírásokat mindenki szabadon felhasználhatja, de tilos azokat a kiadó írásbeli engedélye nélkül másolni, terjesztetni. † ENTERPRESS © 1994. AMEKO Kft.