

ENTERPRESS

Magazin az ENTERPRISE felhasználóknak

2017/2-3. március – június

Hisoft Pascal

SIDBASIC

Adventure



Külső
színbemenetes
kártya

ENTERPRISE
ONE TWO EIGHT

STOP

Az idő: ellenség vagy jó barát?



írta: **Matusa István**
(Tutus)

Többségünk 18-20 éves korában találkozott először az Enterprise számítógéppel. Akkoriban, ilyen fiatalon szinte mindenkinek több ideje maradt a gépre. Persze anno csak vártuk és vártuk azt, hogy a frissen vásárolt új számítógéphez jönnek majd a játékok, felhasználói programok, hardver kiegészítők. Akkor még senki sem tudta, hogy ezekből semmi sem lesz... Pedig ha tudtuk volna, nem itt tartanánk.

Cikkünk címéből pedig ma egyértelmű a válasz: az idő a legnagyobb ellenségünk! Többségünk családos, vagy ha nem, akkor olyan munkahelyen dolgozik ahol „fél életét eltölti”, vagy mindkettő. Így nagyon kevés idő marad programokat írni, hardvereket fejleszteni. Így talán marad a legjobb megoldás, melyre már többször volt példa az enterpriseforever.com oldalain: több ember összefogva fejleszt programokat. Így is hosszú idő, míg egy új Enterprise szoftver elkészül, de ez ilyen viszonyok mellett érthető.

Enterprise DevCompo #2

Kicsit az előző bekezdés sorai ide is simán beillenek. A mostani pályaművek 80%-a kis méretű demo anyag, a többi meg átirat. Tehát új program most nem készült. Pedig ez lett volna a lényege ennek a versenynek. Ha jövőre is lesz, lehet, hogy jobban át kell gondolni a versenykiírást. Személy szerint én ezt csak Enterprise gépre írt új programokra írnám ki!

A rosszul tervezett klubtagság

Nem gondoltam volna, hogy az Enterprise Klubba és az Enterpress Magazinra ilyen kevesen fizetnek elő (24 db 1088 k-s előfizetés és 3 db 128 k-s, valamint 14 db ingyenes 64 k-s előfizetés érkezett...

Tisztázzuk: 2017-ben hat klubtalálkozót szerveztünk / szervezünk. Bármennyire is sikerült jó és olcsó helyet találnom a klubnak, így is a hat alkalom 72 000 Ft. És akkor még nem beszélünk az Enterpress Magazin nyomtatásáról, a domain név és tárhely fenntartásról, a dombornyomott kártyákról és az egyéb költségekről. Nem kell nagy matek tudás ahhoz, hogy lássuk: nem lesz elég az előfizetők által befizetett összeg a klubtalálkozókra valamint az Enterpress Magazinra...

Nyilván ez hiba volt részemről is, mert ezt a verziót is szem előtt kellett volna tartanom: mi van, ha kevés előfizető lesz?

Így most arra kérnék mindenkit, hogy lehetőségéhez képest legalább egyszer nyomja meg a **DONATE gombot**, vagy az alábbi számlára tud még utalni:

Matusa István,
11600006-00000000-15684200
Köszönjük!

Lehet, hogy vezércikkem kicsit negatívra sikeredett, de úgy érzem, hogy bár sokan vagyunk egy retro számítógépes klubhoz képest, de még mindig kevesen. Nagyon bízom benne, hogy a 2018-as év ennél jobb lesz! Kérem mindenkit, ha van ismerőse akit érdekel az Enterprise, szóljon neki, hogy van ilyen lehetőség!

Ajándéktárgyak és dobozos Bricky Prise

Hevesi József barátunk jóvoltából készülnek Enterprise ajándéktárgyak. Jelenleg még csak bögrék, de bízunk benne, hogy hamarosan bővül a kínálat! Tervezték egy web áruházat, de ennek elindítása még nem sikerült. Reméljük, hogy a közeljövőben elindulhat, hiszen a tapasztalatok azt mutatják, hogy igenis van ezekre igény!

Már készül a népszerű EREDETI Enterprise játékprogram, a Bricky Prise, dobozos verziója! Egy exkluzív kiadásban adjuk majd ki, kazettán és 3,5"-os floppy lemezen. A részletekről hamarosan tájékoztatunk Benneteket!

Azok a fránya cikkek!

Nézegetem az Enterpress általam szerkesztett régebbi számait és már akkoriban is ez volt a gondunk! Pontosan az, hogy az Enterpressben nincs elég cikk. Mostani késői megjelenésünknek is ez az oka. Most egy összevont, dupla számmal kellett előrukkolnom és így „időben vagyunk”.

Ebben is segítségeteket kérném!

Bármilyen Enterprise-szal kapcsolatos témában várjuk írásaitokat, cikkeiteket!

Ha szeretné támogatni az ENTERPRESS Magazin megjelenését, itt megleheted:

<http://enterpress.news.hu>
(az oldal alján „TÁMOGATÁS”)

Külső színbemenetes kártya

A külső színbemenetes kártya a NICK EC3..ECO bemeneteinek a tesztelésével indult, csak egy kicsit elszaladt a fantáziája az enterpriseforever.com tagjainak. Ez persze nem baj, sőt... Ez eredeti szándéka a készítőknél az volt, hogy ezen bemenetek segítségével a géphez csatlakoztatható egy olyan elektronika, amivel a képernyőn hardveres sprite-ok jeleníthetők meg. Csak itt – a felépítésből adódóan – több lehetőség is van. De a fő csapásirány a sprite-ok megjelenítése. Ennek ugye az a lényege, hogy a program betölti a megfelelő memóriába a sprite-ok képét, majd a hardver „rákeveri” az eredeti Enterprise képre a megfelelő pozícióban. Ez a pozíció állítható, így a sprite mozgatása a koordináták átírását jelenti a gép processzora számára, ami sokkal kevesebb munka, mintha „ki kellene másolni” (a megfelelő maszkolással) a NICK által generált képre.

Az Enterprise hardverében a NICK úgy lett tervezve, hogy az EC3..ECO bemeneteken érkező pixel „keresztülmegy” a NICK palettázó áramkörén, majd úgy jelenik meg a monitoron. Viszont ezt a legmagasabb felbontáson is tudja, tehát a kívülről jövő sprite-kép 16 színű lehet a maximális felbontáson, ahol az Enterprise eredeti képén maximum 2 szín lehetett egy sorban.

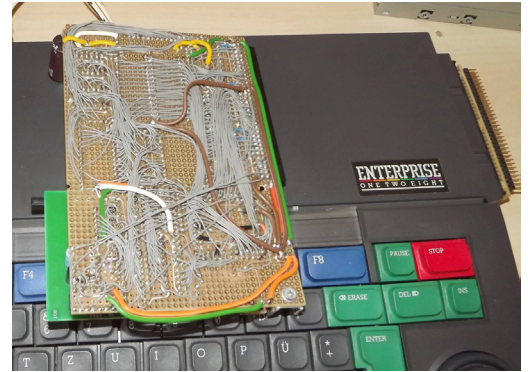
Az alapfelállítás az volt, hogy lesz majd pár darab megjeleníthető sprite. Ehhez jött hozzá az az ötlet, hogy nem muszáj csak sprite-képeket „beküldözgetni” a NICK-nek, jöhet kívülről akár egy egész képernyőnyi tartalom is. Ennek az lehet a jelentősége, hogy olyan üzemmódo(ka)t és olyan felbontással lehet csinálni, amit az eredeti NICK nem is tudna alpból megjeleníteni (ilyen például a 640(+) \times 288-as felbontás, amit a NICK 2 színnel tud, kívülről ez meg mehet 16 színnel is.)

Kérdések és válaszok:

1. Ez a játékefejlesztőknek azért jó, mert a sprite-okat nem kell szoftverből megjeleníteni, (nem is mindig lehetséges,) ezt tudja a hardver. A „drága” CPU idő használható másra. De ez kell majd a játékosoknak is, mert csak így fog megjelenni a végeredmény az ő gépükön is. Szóval „jó lesz ez” majd mindenkinek.

2. A kártyának nincs video-csatlakozója, maga a kártya az, ami generálja az EP-be menő adatokat, erről szól majd ez a projekt. Szóval de, „simán a kártyából „mennek” az adatok az EP felé”.

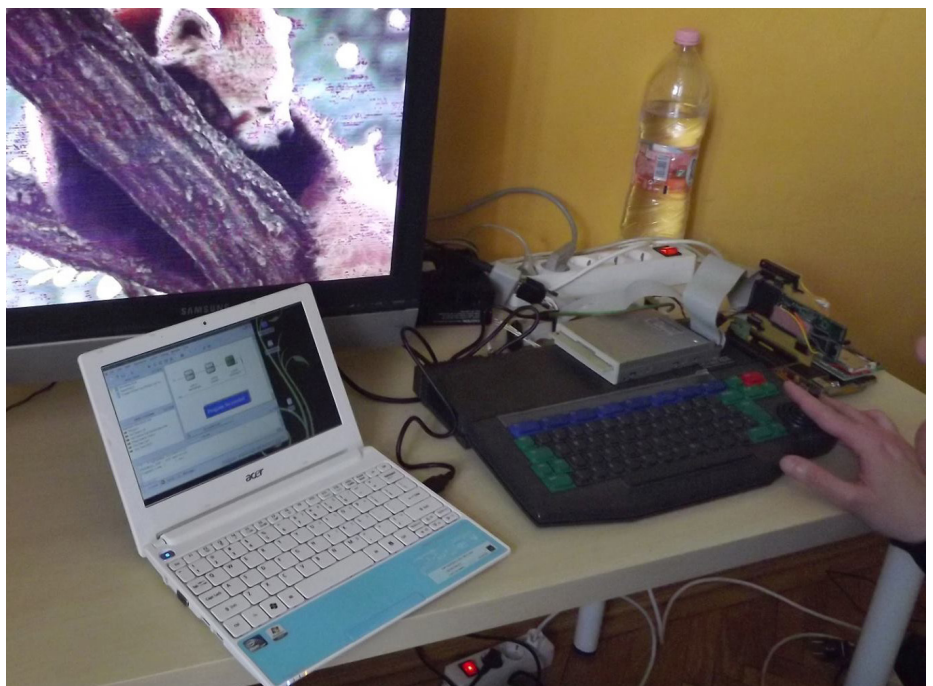
3. A kártya közvetlenül nem használható „külső videokártya”-ként, vagyis nem úgy. Videokártyaként használható, de a kép továbbra is az EP video-kimenetén jelenik meg, és ugyanazokat a színeket tudja megjeleníteni, mint amit az EP eddig is tudott. Ami több lehet a NICK tudásánál, az a magasabb felbontáson használható színek száma,



illetve azért nekem is vannak plusz ötleteim. (Pl. tudhat végre hardveres vízszintes finom-szkroll-t.)

Jelenleg itt tart a dolog, de az eddigi ilyen jellegű hardvereimen egyelőre „túlmutat” bonyolultságban, szóval van benne kihívás bőven. Remélem a történet egy kicsit „világosodott” az Enterprise felhasználóknak. Ha bármilyen kérdés van ezzel kapcsolatban, az enterpriseforever.com fórumon a „Tesztelés” topicban válaszolok.

Balagesz



Enterprise tartozékok



Írta: Matusa István
(Tutus)

Lassan már 32 éves lesz kedvenc számítógépünk, az Enterprise. Ez idő alatt nyilván a gép külsején és belsejében lévő tartozékok felett is eljárt az idő. Nézzük, hogy mit sikerül pótolni és mit nem.

Tartozékok, melyeket sikerül pótolni:

Billentyűzet fólia: Kiss Tamás (MCU Developments) <http://www.mcudevelopments.com/> kiváló fóliát árul, melyet weboldalán, valamint a Vate-rán lehet megrendelni tőle.

Joystick kupak: Szintén Kiss Tamástól lehet rendelni, nagyon jól sikerült darab, nem 3D nyomtatóval készült, hanem öntött műanyag.

3D nyomtatóval az alábbi Enterprise alkatrészeket lehet nyomtatni:

- Joystick kupak
- hátsó hűtőborda takaró rács (itt kérdés, hogy az anyag bírja-e a hűtőborda által leadott nagy hőt)
- jobb oldali takarólemez
- Cartridge SD kártyához

A nyomtatáshoz a fájlokat innen lehet letölteni: <http://www.thingiverse.com/search?q=Enterprise+128>

A gép alján lévő gumitalpakat szerencsére lehet kapni a <http://www.mouser.com> oldalon, valamint Budapesten a Conrad szaküzletében. Hihetetlen, de ezt még mindig a 3M cég gyártja mint annak idején, mikor az EP megjelent.

A **belső fekvő kondenzátorok** is elfáradtak a 32 év alatt, szerencsére ezeket is lehet pótolni (<http://www.mouser.com>).

Tartozékok, melyek jelenleg még nem pótolhatóak:

- a joystick-ot körbevevő gumiharang
- a billentyűk
- a billentyűzet és a fólia között lévő gumi betét
- a magnókimenetet kívülről takaró sötétszürke papír

Enterprise hardverek

Az Enterprise történetét szerintem már mindenki tudja. A gép megjelenésekor sajnos nem sok hardverrel kényeztettek el minket, pedig mint utólag kiderült, nagyon sok lehetőség lett volna erre...

Nézzük, annak idején mit lehetett beszerezni hozzá:

- EXDOS kártya
- Speak Easy
- Egér
- Nyomtató

A felmentősereg – hazai fejlesztések

A magyarországi képviselettel megbízott 'A' Stúdió mindössze egy Spectrum Emulátort és egy EP-Plus cartridge-et hozott össze több év alatt.

A MICROTEAM Kft. tervezett egy **EXDOS kártyát**, melyen 515 kb RAM bővítés van.

Az SMD Team tervezett a gép belsejébe építhető **6 MHz-es turbó kártyát**.

Hazánkban **Mészáros Gyula** gondoskodott arról, hogy változzon a helyzet, és tegyük hozzá, szuper dolgokat tervezett az Enterprise gépéhez:

- **5 foglalatos buszbővítő egység**
- **1 MB-os bővítő kártya**
- **6 férőhelyes EPROM kártya**
- **6 férőhelyes EPROM / SRAM**

bővítő kártya
- **Soros vonali illesztőkártya**
- **Órákártya**

Hazánkban **Zozo tervezett először winchester vezérlő kártyát** Enterprise-hoz, mely nagyszerűen működik.

Talán az eddigi legnagyobb szenzáció **Sárközi Gergely / Sebestyén Pál / Németh Zoltán Zozo SD kártyája**, melyet a cartridge helyére lehet csatlakoztatni. Ha belegondolunk, így elfelejthetjük az EXDOS kártyát, a floppy meghajtókat, a winchester vezérlőkártyát és a winchestereket... Nagyszerű fejlesztés!

Külföldi fejlesztések

Szintén 2015-ben **Maciej Gruszczycki (Pear)** lengyel barátunk alkotta meg az **EterMice kártyát**, mellyel PS/2-es egeret valamit két külső joystick-et lehet a géphez csatlakoztatni.

Saint angliából tervezett a felső RAM panel helyett egy **1 MB-os RAM kártyát**, ez is nagyszerű kiegészítés, bár sajnos már nem kapható

Maciej Gruszczycki tervezett egy külső **EXDOS kártyát**, de igazán (talán nem sértődik meg) nem lett az igazi.

Készülő fejlesztések

Bruce Tanner EP Net kártyája, melyet nagyon várunk már. Ez közvetlenül a gépre dugható, valamint a Mészáros Gyula féle buszbővítőre is csatlakoztatható lesz.

Maciej Gruszczycki (Pear) három külső kártyát is tervezett, mely szakit az eddig megszokott élcsatlakozós buszbővítővel, ezt egy szabványos tükörsor helyettesíti és így a kártyákat „egymás tetejére” lehet pakolni.

A három kártya: **Flexi Bridge** (ez csatlakozik a géphez), **IDE – Compact Flash kártya, RAM-Flash-Clock kártya.** Ezeket is nagyon várjuk!

Balagesz a külső színbemenetes sprite kártyát tervezi éppen, melyet legutóbb elhozott az Enterprise klubba. Jelenleg ez még csak egy teszt verzió, így nagyon sok munka lesz még vele. Egyértelmű, hogy ez lesz az EP történetének (az SD kártya után) a második legnagyobb dobása! Mindenki nagyon várja!

SzörG (Sárközi Gergely) tervez a felső memória panel helyére egy **RAM / Flash kártyát**, és szintén ő tervez Enterprise-hoz egy **turbó kártyát.** Egy **EP – Scart kábel** is tervben van.

Kevés vagy sok?

A Mészáros Gyula féle kártyákból szinte már alig lehet találni párat. Felmerült egy **új buszbővítő** tervezése is, mely már nem élcsatlakozós lenne.

Mindig is nagy kérdés, hogy érdekes-e ilyen kis szériában hardvereket gyártani? Szerintünk a válasz: igen! Hiszen ez egy retro közösség, és nyilván nem több ezer embert fog össze.

Ami hiányzik

A **dátum és órakezelést** már jó régen megoldotta Zozo, de hardver

igazán nincs hozzá, ez nagy hiányosság (pedig szó szerint nem nagy kártya, így más kártyára is rá lehetne tervezni...). Igaz, hogy Mészáros Gyula tervezett ilyen, de csak 1-2 db felkelhető belőle és már sajnos nagyon elavult.

Akik még mindig szeretnék a floppy meghajtókat használni, biztos szeretnének egy **memóriabővítő EXDOS kártyát** (hasonló mint a Microteam kártyája) ami már persze jóval modernebb lehetne.

Az **EP - Scart** csatlakozót csak Spanyolországból lehet beszerezni elég borsos áron és sajnos a minősége nem túl jó.



ENTERPRISE BÖGRÉK rendelhetők az alábbi e-mail címen:

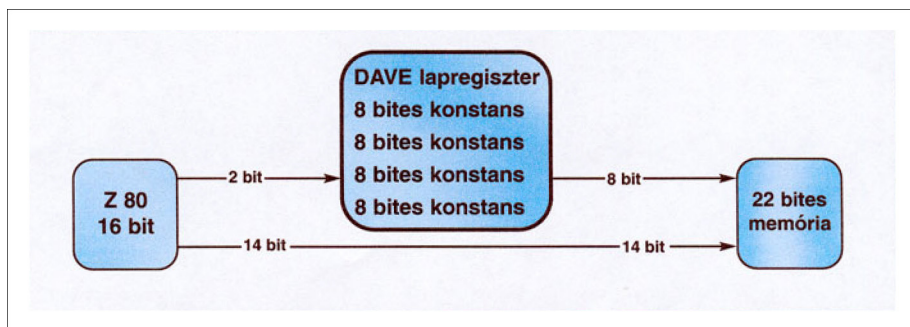
inkedpixelshop@gmail.com

Az Enterprise memórialapjainak kezelése

Az EP-ben Z80-as processzor dobog, mely a memóriát 16 biten címzi meg ami csak 64 kilobyte tartományt eredményez. A Z80 illesztését a DAVE-vel oldották meg, mely a következő feladatokat végzi. 4 db 2*6 bites sztereo hangcsatorna hardver alapja. Programozható hang vagy időalap. 2 belső és 2 külső megszakítás kiszolgálása. A memória címzés magas bitjeinek generálása. Az EP 22 bites tárat tud kezelni, ami 4 Megabyte tartománynak felel meg. A címzés a Z80 és a DAVE tartalmától függ. Az alsó 14 bitet a Z80, a felső 8-at pedig a DAVE generálja az alábbi módon. 4 írható-olvasható lapregisztere van, melyben 8 bites értékeket tárol. A Z80 2 felső bitje megcímzi a lapregiszterek egyikét és a benne tárolt értéket küldi ki helyette.

Az ábrából kitűnik, hogy az EP 64 K-s Z80-ja 4 Megát tud címezni. Ehhez viszont időnként módosítani kell a DAVE lapregiszter tartalmát. Ezt lapozásnak nevezzük.

Miért kell lapozni? Mert a memória meghaladja a Z80 által kezelhető 64 kilobyte-ot. Amikor egy könyvet olvasunk, 2 oldalt láthatunk egy időben. A többi lapozással érhetjük el. Az EP-nek 4 lapja van, egyenként 14 bites azaz 16 kilobyte memória kapacitással. Bármely lapra, bármely szegmens belapozható, így akár mind a 4 lapra belapozható ugyanaz a szegmens. A szegmensek száma 256, ebből a valóságban keveset használunk. A gép bővítésével számuk növelhető. A memória 2 féle lehet. Csak olvasható ROM, írható és olvasható RAM. A ROM-memória kikapcsolás után is megőrzi a tartalmát. Hiányában elképzelhetetlen a hidegindítás. A ROM-okon nem csak az operációs rendszer rezidens részét tárolhatjuk, hanem tetszőleges bővítéseket, programokat és adatokat.



Egyszerűbb vezérlési rendszerek akár csak ROM kiépítésű memóriával is működhetnek. A bonyolultabb feladatokhoz, ilyen pl. a képernyő kezelés, már elengedhetetlen a RAM memória. Az EP operációs rendszerre szintén megköveteli a minimum 2 szegmensnyi RAM-ot.

Mit, mikor, hova lapozunk? Az alapkiépítésű EP a következő memória szegmenseket tartalmazza:

00 ROM operációs rendszer
01 ROM operációs rendszer,
Basic rutinok, WP
04 ROM Basic interpreter

F8 RAM
F9 RAM
FA RAM
FB RAM

FC RAM videó RAM
FD RAM videó RAM
FE RAM videó RAM
FF RAM videó RAM

A videó RAM fogalma

A NICK videó chip a Z80-tól független módon kezeli a képernyőt, és a 64 K-s videó RAM-ot a NICK prioritásban elsődlegesen tudja olvasni. A címei úgy alakulnak, mintha FC FD FE FF szegmenseket lapoznánk be a Z80-nak. A valóságban ilyen belapozással nem futtatunk programot,

ezért a videó RAM egy-egy szegmensét belapozzuk valamelyik lapra és átszámoljuk a NICK címét Z80 címre, vagyis a 16 bites érték 2 felső bitjét módosítjuk. Az EP hidegindítás és RESET utáni környezete: A NICK 4 regiszterének értéke 0. Ezek a FIXBIAS, BORDER, LPL, LPH. A DAVE 16 hangregisztere, 4 lapregisztere, 1 megszakítás regisztere, 3 I/O regisztere, 1 memória elérést szabályzó regisztere szintén 0. A Z80 PC-regisztere szintén 0. Ebből kiindulva tehát mind a 4 lapon a nullás szegmens látható. A Z80 az induló kódot tehát a nullás lap 0000 címéről olvassa, melyen a nullás ROM szegmens van belapozva. A megszakítás tiltott állapotban van. IM 1-rc kapcsol, majd elugrik a 3. lapra ahol szintén a nullás romszegmens látható. A 4. szegmenseket a nullás lapra lapozva ellenőrzi a „TEST_ROM” kezdést. Ha illet talál, átadja neki a vezérlést 0008 címen. így működnek a gyorsesztes bővítések. Tovább futva ellenőrzi a melegindítási mód lehetőségét. Ez annyit jelent, hogy némi inicializálás után visszaadja a vezérlést az aktuálisan futó felhasználói programnak. Tovább futva megkezdődik a hidegindítás. RAM teszt, illetve a RAM szegmensek listába vétele. Ekkor már kialakul az operációs rendszer lapkiosztása. A nulladik lapon a legalacsonyabb értékű RAM-szegmens található, melynek értéke alapgép esetén F8. Ide másolódik 0030-005A közötti területre az operációs rendszer nulláslapú rutinjai. Az RST

30 az EXOS funkcióhívásokat, az RST 38 pedig a megszakításokat szolgáltatja. Ezt a nulláslapot tehát nem szabad soha elállítani, kivéve úgy, hogy tiltott megszakítás mellett ellapozzuk majd helyreállítjuk! Az első lap tartalma virtuális, ez a munkaszegmens. A második lapra az FF RAM-szegmens, vagyis a rendszerszegmens kerül. Itt van kiépítve az operációs rendszer veremtára, tehát ki lapozás alatt gondoskodni kell arról, hogy ne történjen megszakítás, illetve ne legyen vermet igénylő művelet. A harmadik lapon a nullás ROM-szegmens marad.

A továbbiakban történik a ROM-szegmensek könyvtárba vétele, ROM-ok RAM igényének kiszolgálása, alaperifériák könyvtárba vétele és inicializálása, majd a ROM-bővítők inicializálása. Ezt követi a bejelentkezés, mely le is tiltható. Utána a hidegindítás akciókóddal megtörténik a bővítők letapogatása. Amikor egyetlen bővítő sem igényli, akkor a 01-es ROM-szegmens elfogadja és elindítja a WP felhasználói programot.

Az operációs rendszer alatt két féle program futtatható: Felhasználói és rendszerbővítő.

Mindkét program létrehozhat felhasználói periféria bővítést, mely leegyszerűsítve harmadik lapon futó rutinkönyvtár, belépési táblázzal. Az EP alatt minden esetben egyetlen felhasználói program futhat. E programból EXOS-hívás vagy megszakítás által meghívható az operációs rendszer.

Az operációs rendszer tovább hívhatja a bővítőket és perifériákat. Ebből következik, hogy legnagyobb szabadsággal a felhasználói program rendelkezik. A nullás laptól eltekintve a szegmenseket szabadon lapozhatja. Veremtárát a harmadik lap kivételével bárhol kialakíthatja. Felhasználói RAM memóriát igényelhet vagy szabadíthat fel az operációs rendszeren keresztül. Amikor egy másik felhasználói program veszi át az irányítást, akkor az operációs rendszer feladata a memória, perifériák és bővítések inicializálása. Felhasználói futást 2 féleképpen lehet kialakítani. Fájl-típusú vagy bővítésből indított. Fájl-típus. A fejléc 00, 05-tel kezdődik. Ezt követi a két bájtos betöltési méret. A rendszer betöltési rutinja nem telje-

sen korrekt. 0100-7FFF-ig hibátlan. 0100-BFFF-ig töltve a szegmenseket nem lapozza be, tehát ki kell találni. Ezért azt javaslom, hogy a 7FFF-et ne haladja meg az 5-ös fejlécű programfájl. A további adatokat a program is beolvashatja egy másik fájlról. A betöltő 5-ös fejlécű programnál felszabadítja a felhasználó által foglalt memóriát, lefoglalja a betöltéshez szükséges memóriát, majd végrehajtja a betöltést. Töltési hiba esetén hibaüzenetet küld az alapértelmezésű csatornára, majd hidegindítás akciókóddal meghívja a bővítőket. Hibátlan töltésnél inicializálja a perifériákat és bővítőket, felhasználói futásra kapcsolja a rendszert és 0100-tól elindítja a programot. Újabb hibája hogy a megszakítást nem tiltja, pedig a felhasználói program még nem építette ki a veremtárát. Első lépés tehát az LD SP,? vagy a DI programsor legyen! Kis veremigényű programoknál bevált szokás az LD SP, 100H utasítás.

A lapok kiosztása ebben az esetben:

0. Nulláslap RAM-szegmens
1. Programot tartalmazó RAM-szegmens, rövidebb programok esetén tartalma tetszőleges lehet.
2. FF rendszerszegmens.
3. Nullás ROM-szegmens.

Bővítőből indított felhasználói futás: hidegindítás akciókód elfogadása esetén a teendők.

```
LD C,0
EXOS 0
LD SP,?
EI
```

Parancs sztringgel indított felhasználói futásnál a felismerés utáni ág.

```
LD C,60H
EXOS 0
LD SP,?
EI
```

A felhasználói program lépései a továbbiakban.

- Beállítja az EXOS-változókat.
- Megnyitja a létfontosságú csatornáit.

- Igény esetén kiépíti a felhasználói megszakítást.

```
LD A,255
OUT (0B2H),A
LD HL, inter
LD (0BFEDH),HL
```

- Igény esetén kiépíti a SOFTWARE megszakítást.

```
LD HL, softint
LD (3DH),HL
```

- Igény esetén kiépíti a melegindítási címet.

```
LD A,255
OUT (0B2H),A
LD HL,melegstarl
LD (0BFFRH),HL
```

A felhasználói program nem fejezheti be a futását mint egy bővítő rutin. Kilépési lehetőségek.

- Parancs sztring átadás.
LD DE,BASIC vagy WP stb.
EXOS 26

JP HIBA ;nem ismeri fel a sztringet a rendszer BASIC: DB 5,"BASIC"
WP: DB 2,"WP"

- Bővítések hívása hidegindítással.

```
LD A,255
OUT (0B2H),A
ISMET XOR A
LD (0BF78H),A
EXOS 26
JR ISMET
```

Hidegreset

```
LD C,80H
EXOS 0
```

A bővítő rutinja visszatérés előtt a nullás és második lapot állítsa helyre. Vigyázz a lapozásoknál mivel:

0. lap Itt van a megszakítás belépési pontja.
2. lap Itt van a veremtár.
3. lap Itt fut maga a rutin.

Javasolt könyv:

az EXOS 2.1 VÁLTOZAT MŰSZAKI LEÍRÁSA

Haluska László

(Az 1999-es Enterpress News tervezett 1. számából, mely nem jelent meg)

EXOS kompatibilis memóriakezelés - II. rész



Írta: Németh Zoltán
(Zozosoft)

Akkor menjünk bele a részletekbe, először egy kis összefoglaló: Van 4MB címtartományunk, ez fel van osztva 256 db 16K-s szegmensre. Meglepő módon 0-255-ig számozzuk őket, hexában 00-FF. Alapvetően bármelyik lehet RAM vagy ROM vagy maradhat üresen, kivéve amit maga az alaplap határoz meg: 00-03 az alaplapi ROM-hoz van rendelve, FC-FF pedig a szintén az alaplapi RAM-hoz, aminek kiemelt szerepe van, hiszen egyben a Nick chip által látott videó memória. 04-07 tartozik a cartridge foglalat-hoz, de itt akár RAM-ot is elhelyezhetünk. A 128-as gépekben helyett kapott egy plusz 64K-s bővítpanel, ami F8-FB szegmenseket tartalmazza. Ami még nagyon elterjedt: a MICROTEAM kártya 512K bővítése a 40-5F területet foglalja el. 320K-ra bővített gépben pedig EC-FB található a bővítpanelen. Itt érdekességként megjegyzem, hogy az egyik NASA&GUY demo (azt hiszem, valami Jean Michell Jarre digitális zenét játszik), ilyen „EC-s” gépre íródott eredetileg, amin nagyon csodálkoztam anno, akkor még nem jelent meg az Enterpressben a 320K átalakítás cikk. És hiába volt MICROTEAM kártyával 640K-s gépem, mégse játszotta le a teljes zenét, hála a fix című programozási stílusnak... aztán én meg átirtam, hogy a MICROTEAM kártyára eső szegmenseket használja, így végre meghallgathattam a teljes zenét.

Ha a RAM-ot nézzük, egy 64K-s gépben van FC-FF. 128-asban F8-FF. Ha a korábban emlegetett MICROTEAM+EP64 konfigot nézzük, akkor pedig 40-5F, FC-FF, ami darabon bőven jó, csak hiányzik az a bizonyos F8-FB, amire az EXOS-t nem használó 128-as gépen programozók által írt programok nagy része hivatkozik közvetlenül.

A helyes programozáshoz felejtünk is el ezeket a számokat, egyedül az FC-FF-et kell megjegyezni, kitüntetett videó memória mivoltuk miatt. A többről csak annyit kell tudnunk, hogy nekünk hány darabra van szükségünk, a konkrét szegmensszámokat majd megmondja az EXOS!

Most nézzük az EXOS szerinti RAM felosztást: két szegmensnek van kitüntetett szerepe, az egyik az FF ami a rendszerszegmens, és a legalacsonyabb sorszámú RAM szegmens ami a nulláslap szegmens. Itt található meg az EXOS hívások, illetve a megszakítási program belépési pontja. És ide kerülnek 100H címtől töltve az 5-ös fejlécű programok is. Egy 128-as gépen az az F8 szegmens. De ha pl. van egy MICROTEAM kártyánk, akkor már a 40-es lesz az. És ez máris gondot okoz sok programnak (általában a komplett módosított Spectrum ROM-ot tartalmazó béna átiratoknak)... De bővíts gépen is lehet F8 a nulláslap, ha VENUS-t használunk, ill. az EP-DOS 2.1-nek is van ilyen lehetősége. Ez az eset meg egy másik adag programnak okoz gondot... A maradék RAM négy csoportba tartozhat: rendszer, eszköz, felhasználói, szabad. Ha pl. csatornákat nyitunk meg, kü-

lönösen, ha nagy RAM igénnyel járó videó lapokat, akkor az EXOS elkezd lefelé terjeszkedni, és ha kihizza az FF szegmenst, akkor további szegmensek válhatnak rendszer által lefoglalttá. A különböző beláncolt EXOS periféria kezelők által igényelt teljes RAM szegmensek az eszköz (device) kategóriában kerülnek lefoglalásra. Erre tipikus példa a RAMDISK. Szintén ebben a kategóriába kerülnek lefoglalásra a betöltött rendszerbővívők által elfoglalt szegmensek is. És végül van az aktív felhasználói program, ez lehet egy rendszerbővívő vagy egy 5-ös fejléccel betöltött „új alkalmazói program”. Az ezek által igényelt szegmensek a felhasználói kategóriában kerülnek lefoglalásra.

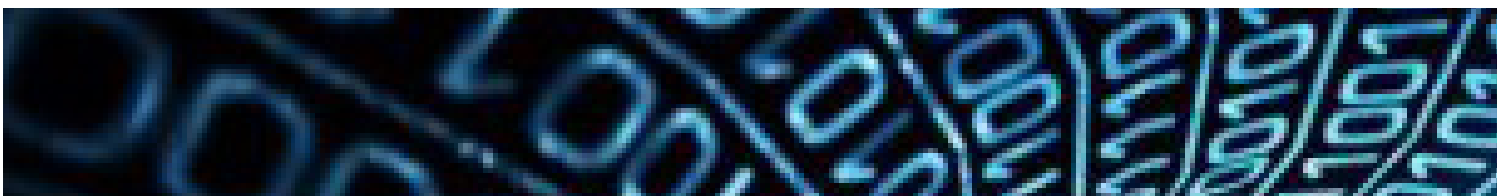
Ami nagyon fontos: a felhasználói program csak felhasználói szegmenst tud felszabadítani, eszköz vagy rendszer szegmens felszabadításához nincs joga!

És ezzel el is érkeztünk ahhoz a bizonyos Spectrum Világ-os hibás módszerhez:

```

1. LD C,0FAH
2. EXOS 25
3. LD A,0FAH
4. OUT (0B2H),A
5.
```

Egyrészt a szándék dicséretes, hiszen nem csak bummm belapozza a szegmenst, hanem szabaddá teszi... csak sajnos a nem túl sikeres fordítású EXOS leírást félreértelmezve Mert az EXOS 25 hívás csak akkor lesz sikeres, ha előtte az a szegmens nekünk, azaz felhasználóiként ki lett utalva. Ha az FAH már mondjuk a RAMDISK



része, vagy egy betöltött EXOS bővíthető van ott, akkor mivel eszköz szegmensnek van lefoglalva, hibajelzést kapunk vissza, hogy a szegmens nem szabódítható fel. A program viszont ennek ellenére nyugodtan neki áll használni. És ott van még az a eset is, hogy nem is létezik ez a szegmens az adott konfigurációban, mert pl EP64-ről van szó... A helyes megoldás itt az, hogy kérünk kiutalni egy szegmenst az EXOS-tól, és azt használjuk. Ha hibajelzést kapunk vissza, mert elfogyott a szabad memória, akkor azt a programtól függő módon le kell kezelni (pl. kilépés, vagy lehet folytatni korlátozott funkcionalitással a program futását).

1.	EXOS 24
2.	JP NZ, HIBA
3.	LD A, C
4.	OUT (0B2H), A
5.	

Kicsit macerásabb, ha egy konkrét számú szegmensre vágyunk: ekkor ciklusban kell ismételtetni az EXOS 24 hívást, mindaddig, amíg meg nem kapjuk a kívánt szegmenst, vagy pedig elfogy a memória, és ekkor megyünk a HIBA rutinra. De erre a módszerre új program írása esetén csak egy esetben lehet szükség: ha videó szegmensre van szükség. Ekkor addig ismételjük a hívást ameddig FC vagy nagyobb szegmenst nem kapunk. És akkor kell még ehhez a módszerhez folyamodni, ha egy régebbi fix címzések programot akarunk kibővíteni úgy, hogy amit a program használ, az legyen szabályosan lefoglalva, mint ahogy ezt tettem tegnap a TUSKER-rel. Ez volt Attus eredeti rutinja, amely az SpV módszeren alapult:

1.	LD BC, 5FAH
2.	SZABAD PUSH BC
3.	EXOS 25
4.	POP BC
5.	INC C
6.	DJNZ SZABAD

Ebből kiderül, hogy az FA-tól kell nekünk 5 szegmens, ill. később

kiderült, hogy az LPT táblát az FF szegmens elejére helyezi el.

És íme mindez helyesen:

1.	FOGLAL	EXOS 24
2.		JP NZ, HIBA
3.		LD A, C
4.		CP 0F9H
5.		JR NZ, FOGLAL
6.		LD BC, 5FAH
7.	EZKELL	PUSH BC
8.		EXOS 24
9.		LD A, C
10.		POP BC
11.		CP C
12.		JP NZ, HIBA
13.		INC C
14.		DJNZ EZKELL
15.		EXOS 24
16.		CP 7FH
17.		JP NZ, HIBA
18.		LD DE, 3200
19.		EXOS 23
20.		JP NZ, HIBA
21.		LD L, 0F9H
22.	VISSZAAD	LD C, L
23.		EXOS 25
24.		DEC L
25.		JR NZ, VISSZAAD

Az elején addig foglalunk, amíg az F9-ig el nem jutunk. Ha közben kifogyunk a memóriából, akkor ugrás a HIBA-ra. Ezután kérünk még 5 szegmenst, aminek FA, FB, FC, FD, FE-nek kell lennie, különben HIBA... És még egyet igényelünk, itt már csak az FF jöhet, és „megosztott szegmens” hibajelzést kell kapunk (ez a 7F hibakód), különben hiba. És itt jön még egy fontos dolog: megosztott szegmens használata esetén meg kell mondanunk az EXOS-nak, hogy mi meddig akarunk terjeszkedni, erre szolgál az EXOS 23: felhasználói határ beállítása. Ha itt hibajelzést kapunk, akkor nincs annyi szabad hely amennyire nekünk szükségünk van, tehát ugrás a HIBA-ra. Ha minden ok, akkor egy nem túl szép, nem túl gyors, de egyszerű módszerrel visszaadjuk a felesleges szegmenseket. Vagyis F9-től lefelé mindet megpróbáljuk felszabadítani, úgyis csak az fog sikerülni amit felhasználónak utaltak ki. A teljesen korrekt megoldás természetesen az, ha az elején eltároljuk sorban a kapott felesleges szegmenseket, és ezen lista alapján szabadítunk itt fel. Erre akkor van szükség,

ha az 5-ös fejlécű programunk már nem fér el a nullslapon, mert ekkor a további programszegmensek is felhasználóként kerülnek lefoglalásra, és ezzel a primitív módszerrel azokat is felszabadítanánk, pedig azt nem lenne szabad. A TUSKER betöltő esetén nincs ilyen gond, jó így is.

Ez volt a fix szegmensek használata szabályosan című fejezet.

Kérdések és válaszok:

Povi kérdezte:

Konkrét szegmens igénylésére nincsen valami elegánsabb módszer? Mert így szélsőséges esetben akár 249 szegmenst is lefoglalhatunk magunknak, míg végre megkapjuk pl. a hön áhított FC szegmensünket. És akkor ezután még fel kell szabadítani a fölöslegesen lefoglalt szegmenseket is, hátha kellene még azok valamire...

Sajnos nincs :(ez egy kicsi hiányosság az EXOS-ban. Jó lenne ha külön lehetne jelezni, hogy videó szegmenst szeretnénk kérni.

Erre egyedül a periféria kezelő programon keresztül van lehetőség, ha videó periféria típust állítunk be.

Így jobb híján marad ez a ciklusos megoldás. Lehetséges még közvetlenül az EXOS rendszerterületében matatni, ezt több könyvben, cikkben tárgyalták is. De az ilyen programok máris elszállnak EXOS 2.0 esetén... esetleg meg lehetne írni külön két szubrutint EXOS 2.0 ill. 2.1 esetére. De mi van ha véletlenül mégis előkerül az a szuper EP, amit állítólag Kopácsy rejteget? (azóta előkerült - a szerk.)

Azon megint csak nem fognak futni az ilyen „kotorászós” programok...

Szóval szerintem jobb maradni a szabályos megoldásnál, az a pár tizedmásodperc futásidő amibe legrosszabb esetén kerülhet, nem egy nagy veszteség a kompatibilitás oltárán.

Folytatjuk!

HiSoft Pascal



Írta: Povászy Zoltán
(Povi)

Az Enterpress magazin első három évében (1990–92) egy 12 részes cikksorozat ismertette a Pascal nyelvet, Ujlaki László tollából. A publikáció elsődleges célközönsége azok az Enterprise felhasználók voltak, akik az IS-BASIC-et már jól ismerték, és „szintet léptek” volna. A BASIC sebessége korlátozza a „komolyabb” programok írását, a ZZIP fordítónak pedig egyik legnagyobb hiányossága, hogy nem ismeri a lebegőpontos számokat.

Ennek ellenére az Enterprise-on kevés alkalmazás készült Pascal nyelven, azok, amikről tudomásom van, mind az IS-DOS (CP/M) alatt futtatható Borland Turbo Pascallal készültek; ilyen pl. a PCBPrise nevű NYÁK-tervező program.

Az Enterprise-on elérhető másik Pascal fordító a HiSoft terméke. Ez a program érzésem szerint méltatlanul alulértékelt, ami részben érthető, hiszen a fordító csak a szabványos Pascal-t érti meg, emiatt tudása elmarad a Turbo Pascal-étól: a legnagyobb hiányossága a string típus ismerete.

Jelenlegi cikkemmel azonban szeretnék egy kis kedvet csinálni a HiSoft Pascal használatára, mert bár a string típus hiánya tényleg fájó pont, a programnak azonban vannak előnyei is a Turbo Pascal-hoz képest. Mik is ezek? A brit HiSoft eredetileg Spectrum-ra készítette el a Pascal fordítóját, de később több Z80 alapú gépre is kiadta azt, többek között a CPC-re, és az Enterprise-ra is. A „legfejlettebb”, CP/M-re portolt változat a HiSoft Pascal80 névre hallgatott. A Spectrum-os gyökerek ellenére nem Spectrum átiratról van szó, hanem egy olyan program-

ról, ami maximálisan (és szabályosan) használja az EXOS-t, és a forráskódot 5-ös fejlécű futtatható állománnyá fordítja (Translate parancs). Ezzel a tulajdonságával máris szerzett egy plusz pontot a Turbo Pascallal szemben, hiszen ez azt jelenti, hogy nincs szükség IS-DOS-ra se a fordító, se a vele fordított programok futtatására.

```

HiSoft Pascal Amstrad CPC464
Version of 26/3/84
Copyright HiSoft 1983,84
All rights reserved

Alter          Compile
Delete        Edit
Find          Get text
Help         Insert
List         Move
reNumber     Put text
Separator    Run
Set Speed    Translate object
Upper line   Verify
Width       text info
Verify      Z print text
I default
I RSA call
>■

```

CPC464 verzió

Bármennyire is meglepő, a Turbo Pascal – a nevével ellentétben – lassabb kódot állít elő, mint a HiSoft Pascal. Ez a sebességbeli különbség főleg a lebegőpontos számokkal való számolás esetében szembetűnő, aminek az az oka, hogy a Turbo Pascal 6 byte-on tárolja a real típusú számokat, míg a HiSoft 4 byte-on. Természetesen a kisebb méret miatt a számok ábrázolása kevésbé pontos, és az ábrázolható számtartomány is kisebb, de véleményem szerint számunkra bőven megfelelő ez a pontosság is (hiszen nem az ösrobbanás első másodpercét szeretnénk modellezni), és a gyorsaság ebben a gépkategóriában sokkal nagyobb előny.

A HiSoft Pascal méltatlanul mellőzött szerepéhez hozzájárulhatott az is, hogy korábban (a 90-es években) nem volt elérhető dokumentáció. Ma már a Google segítségével szinte bármi megtalálható a neten, ennek ellenére – bár több különböző HiSoft Pascal verzió leírását megtaláltam – az Enterprise verzióé mintha nem is létezne. Az Enterprise specifikus pa-

rancsokat a program részleges disassemblálásával sikerült feltérképezni.

Mint említettem, az Enterprise verzió kihasználja az EXOS lehetőségeit, de korábban sehol nem lehetett olvasni arról, hogy a fordítóban implementálva van egy EXOS nevű függvény, és az ehhez kapcsolódó, előre definiált RA, RB, RBC stb. változók! De nem lehetett olvasni azokról az eljárásokról és függvényekről sem, amelyek – bár nem része a szabványos Pascal-nak – a HiSoft Pascal-ban elérhetők (pl. INCH, MAKESTR, TIN, TOUT stb.)! A most induló cikksorozatban szó fog esni ezekről a nem dokumentált parancsokról, különböző tippeket és trükköket fogok megmutatni, amivel remélhetőleg többen is kedvet kapnak majd a Pascal használatához!

```

HISOFT PASCAL COMPILER

HiSoft Pascal Enterprise
Version 1.2
Copyright HiSoft 1985
All rights reserved

Alter      Buffer left
Compile    Delete
Edit       Find
Get text   Help
Insert     List
Move       reNumber
Put text   U delin
Run        Substitute
Translate  Upper line
Verify    Width 48/88
load exos module Y defaults

```

HiSoft Pascal 1.2

Bár a HiSoft Pascal utolsó hivatalos 1.1-es változata is igen hatékony eszköz, de PovySoft által készített 1.2-es változata nagyon hasznos módosításokat és javításokat tartalmaz:

A rendszer nem 40, hanem 80 karakteres képernyővel indul.

Javítva lett az a bug, ami akkor jött elő, ha a programsorban a sorszámot 13 szóköz követte.

A forráskódot lehet kisbetűvel is írni (a jobb olvashatóság érdekében). Ettől függetlenül a bevitel után, listázáskor már nagybetűvel jelennek meg azok a szavak, amelyek tokenizálva vannak tárolva a memóriában.

A számok után nem ír szóközt, igazodva a Turbo Pascal-hoz.

A PAGE eljárás javítva lett: képernyőre CHR(26)-ot küld, nyomtatóra CHR(12)-t.

Az EXOS függvény eljárásá alakítva (használatát lásd lejjebb).

Három új eljárás, és három új függvény.

Új függvények:

```
function Swap(x: integer) : integer;
Felcseréli az argumentum alsó és felső byte-ját.
```

```
funcion Hi(x: integer) : char;
Visszatér az argumentum felső byte-jával.
```

```
funcion Lo(x: integer) : char;
Visszatér az argumentum alsó byte-jával.
```

Új eljárások:

```
procedure CtrScr;
Törli a képernyőt és a kurzort a bal felső sarokba helyezi.
```

```
procedure GotoXY(x, y: integer);
Pozicionálja a szövegkurzort a képernyő x-edik oszlopába és y-adik sorába.
```

```
procedure SetVar(ExosVariable: integer; NewValue: integer);
Beállítja az ExosVariable EXOS változót a NewValue értékre.
```

Példa:
SetVar(27, 255);
A keretszint fehérre állítja.

Az EXOS eljárás használata

Az EXOS eljárással – bármily meglepő – EXOS funkciókat tudunk hívni Pascal-ból. Az EXOS hívások a paramétereket az A, BC és DE regiszterekben várják, és az eredményeket ugyan-ezekben a regiszterekben adják vissza. Az A-regiszter állapotkódot hoz vissza, ami nulla, ha a hívás sikeres volt, egyébként pedig egy nullától különböző értékű állapotkód.

A HiSoft Pascal-ban előre definiálták az RA, RB, RC, RD és RE char típusú,

valamint RBC és RDE integer típusú változókat: ezeket tudjuk használni az EXOS eljárás hívása során.

Példa:

```
program time;
begin
  exos(32);
  write(Current time is: );
  write(ord(rc):2:H, ,:');
  write(ord(rd):2:H, ,:');
  write(ord(re):2:H)
end.
```

Megjegyzés:

A HiSoft Pascal korábbi verzióiban az Exos parancs nem eljárásként, hanem függvényként volt implementálva, ami az A regiszter értékét adta vissza eredményül. Mivel a Pascal sajnos nem engedi a függvények hívását anélkül, hogy a visszatérési értékét ne használjuk fel valamiképpen, ezért ha nem volt szükségünk az eredményre, akkor is legalább egy dummy változó értékadást kellett tenni, pl.: tmp := exos(32);.

Ennek kiküszöbölésére lett az 1.2-es verzióban az Exos parancs eljárásá alakítva. Mivel az A regiszter értékét automatikusan felveszi a beépített RA változó is, így nem veszítjük el azt az információt, amit az Exos függvényként implementált változata adott: az RA változó tesztelésével leellenőrizhető, hogy sikeres volt-e az EXOS hívás.

Billentyűzet és joystick

használata a

HiSoft Pascal-ban

Az előző számban bemutatam, hogyan lehet Exos függvényeket hívni a Pascalból, most ezek alkalmazásával megnézzük, hogyan lehet a billentyűzet és a joystick olvasását megoldani. A cikk második felében pedig megmutatom, hogyan lehet gépi kódu betétekkel még egy kicsit gyorsítani az elkészített függvényeinket.

A HiSoft Pascal-ban „gyárilag” implementálva van az inch nevű függvény, ami a billentyűzet olvasására használható.

```
function inch: char;
```

A függvény nem vár a billentyű le-

nyomására: ha nem volt lenyomott billentyű, akkor nullával, egyébként a lenyomott billentyű ASCII kódjával tér vissza.

Példa:

```
program TestInCh;
```

```
var ch : char;
```

```
begin
  writeln(.Press any key...');
  repeat
    ch := inch;
  until ch <> chr(0);
  writeln(.You pressed ,, ch)
end.
```

Az inch függvény működése a Turbo Pascalból megismert ReadKey és KeyPressed függvényekhez képest egy öszvér megoldás, nézzük, hogyan lehetne ez utóbbiakat implementálni! A HiSoft Pascal a KEYBOARD: eszközt használja, a 105-ös csatorna van megnyitva neki, mi is ezt fogjuk használni a függvényeinkben:

```
function KeyPressed : boolean;
begin
  ra := chr(105);
  exos(9);
  KeyPressed := rc = chr(0)
end;
```

```
function ReadKey : char;
begin
  ra := chr(105);
  exos(5);
  ReadKey := rb
end;
```

Ezen függvények használatával már képesek vagyunk figyelni a billentyűzetet, és akár a beépített joystick-ot is, hiszen a fő irányoknak az alábbi ASCII-kódok vannak megfeleltetve:

```
fel = chr(176), le = chr(180), bal = chr(184), jobb = chr(188)
```

A joystick olvasására azonban elegánsabb módszer lenne egy olyan függvény, ami úgy viselkedik, mint az IS-BASIC JOY parancsa. Azzal már lehetséges lenne nem csak a fő-, hanem az átlós irányok (pl. balra fel), és a külső joystick-ok olvasása is. Szerencsénkre az EXOS 11-es funkció pontosan ezt (is) csinálja. Lássuk a megvalósítását:

```
function ReadJoy(n: integer) : integer;
```



```
begin
  ra := chr(105);
  rb := chr(9);
  rc := chr(n);
  exos(11);
  ReadJoy := ord(rc);
end;
```

Mivel az integer és char típusok között nincs implicit típuskonverzió, kénytelenek vagyunk használni a chr() és az ord() függvényeket. Ez természetesen nem baj, hiszen a char típust alapvetően karakterek tárolására találták ki; nem véletlenül vezették be a későbbi Pascal változatokban a 8 bites byte és shortint egész típusokat. Ez utóbbi típusok hiánya miatt láthatjuk a fent alkalmazott kényszermegoldást.

Még egy érdekesség, ami a HiSoft fordító hiányosságának tudható: ha az ra:=chr(105) sor helyett ra:=i-t írunk (a kis i betű ASCII kódja 105), akkor a fordító 7 órajellel gyorsabb kódot állít elő. Ennek ellenére nem javaslom ezt a megoldást, mert az a forráskód olvashatóságának és karbantarthatóságának rovására menne (nem ér ennyit 1,75 µs).

A ReadJoy függvény bemenő paramétere a joystick típusa (0: internal, 1: ext1, 2: ext2), visszatérési értéke az érzékelt irány:

- 0 = semmi
- 1 = jobbra
- 2 = balra
- 4 = le
- 8 = fel
- 16 = tűz (szóköz).

Átlós irány esetében a két fő irány összege az eredmény (pl. 5 = jobbra le). A függvény nem ellenőrzi az argumentum helyességét, érvénytelen érték esetében az ext2 joystick-ot fogja olvasni (EXOS tulajdonság). Láthatjuk tehát, hogy függvényünk működése megegyezik az IS-BASIC-ben már megismert JOY paranccsal.

Természetesen a fent ismertetett függvényeken kívül lehetőségünk van a portok közvetlenül olvasására is, így olyan billentyűk lenyomását is érzékelni tudjuk, amelyek önmagukban nem adnak ASCII-kódot (pl. SHIFT, ALT stb. gombok).

A Z80 portok olvasására és írására a HiSoft Pascal az inp függvényt és out eljárást biztosítja:

```
function inp(Port: integer) : char;
procedure out(Port: integer; Value:
char);
```

Gépi kód használata a HiSoft Pascal-ban

Sajnos a HiSoft Pascal-ban nincs lehetőség inline assembly betétek írására (mint pl. a PC-s Turbo Pascal-ban), de az INLINE paranccsal elhelyezhetünk gépi kódu rutinokat a forráskódban. Ahhoz, hogy a gépi kódu rutinunkban fel tudjuk használni a függvény, vagy eljárás paramétereit, ismerjük meg, hogyan tárolja őket a fordító! procedure test(i: real; j: integer); Ebben az esetben a j változó az IX+2 és IX+3 címen, az i változó az IX+4, IX+5, IX+6 és IX+7 címen helyezkedik el. Az érték szerint átadott paraméterek helyfoglalása a típus méretétől függ (ezért foglal 4 byte-ot a real típusú i változónk).

Más a helyzet a referenciaként átadott paraméterek esetében: procedure test(i: integer; var x: real); Ebben az esetben az x-nek nem az értékét (ami 4 byte-ot foglal) adjuk át az eljárásnak, hanem csak a memóriacímét, ami két byte-ot foglal. Tehát az x változó címét az IX+2 és IX+3, az i értékét pedig az IX+4 és IX+5 címen kapja meg az eljárás.

Függvények esetében a visszatérési érték a bemenő paraméterek feletti memóriacímen lesz visszaadva: function test(p, q: integer) : real; Tehát ebben az esetben a q értéke az IX+2 és IX+3, az p értéke az IX+4 és IX+5 címen helyezkedik el, a függvény visszatérési értéke pedig az IX+6, IX+7, IX+8 és IX+9 címen lesz eltárolva.

Ezek után lássuk, hogyan lehetne gépi kódu rutinként megírni az előbb ismertetett KeyPressed, ReadKey és ReadJoy függvényeket:

```
function KeyPressed : boolean;
function ReadKey : char;
```

A fenti két függvény „egyszerű eset”: nincsenek bemenő paramétereik, és mindkettő függvény visszatérési értékeinek típusa egy byte helyfoglalású:

tehát az IX+2 címre kell majd helyeznünk az eredményt. A Pascal-ban implementált boolean típusról tudni kell, hogy a „false” értéket 0-ként, a „true” értéket pedig 1-ként tárolja (elentétben pl. a C nyelvvel, ahol minden nem 0 érték „true”-t jelent), erre figyelni kell majd a rutinunk írásakor. Lássuk tehát a megvalósítást:

```
function KeyPressed : boolean;
begin
  inline(#3E,#69) {LD A,105};
  inline(#F7,#09) {EXOS 9};
  inline(#79) {LD A,C};
  inline(#ED,#44) {NEG};
  inline(#3C) {INC A};
  inline(#DD,#77,#02) {LD (IX+2),A};
end;
```

```
function ReadKey : char;
begin
  inline(#3E,#69) {LD A,105};
  inline(#F7,#05) {EXOS 5};
  inline(#DD,#70,#02) {LD (IX+2),B};
end;
```

Nézzük a ReadJoy függvényt: function ReadJoy(n: integer) : integer;

A függvény n argumentumát az IX+2 és IX+3 címen kapjuk meg, kis-endián formában (azaz az alsó bájt lesz az alacsonyabb címen, vagyis az IX+2-n). A függvény eredményét pedig az IX+4 és IX+5 címre kell helyeznünk, szintén kis-endián formában. Mivel az EXOS 11 függvényhívás eredménye csak egy byte lesz, az IX+5 címre 0-t kell helyeznünk. A lenti példában kihasználjuk azt, hogy a hívás után az A regiszter értéke mindig 0 lesz, így azt írjuk az IX+5-re:

```
function ReadJoy(n: integer) : integer;
begin
  inline(#3E,#69) {LD A,105};
  inline(#06,#09) {LD B,9};
  inline(#DD,#4E,#02) {LD C,(IX+2)};
  inline(#F7,#0B) {EXOS 11};
  inline(#DD,#71,#04) {LD (IX+4),C};
  inline(#DD,#77,#05) {LD (IX+5),A};
end;
```

A cikksorozat következő részeiben a VIDEO: lapok használatáról, a grafikáról, és a fájlkezelésről lesz szó.

(folytatjuk...)

A múlt számítógépes játéka



Írta: Kiss László
(Lacika)

Az 1960-as évektől, a '70-es évek közepéig sorra jelentek az amerikai egyetemeken és fontosabb intézményekben, kutatóintézetekben az ún. nagyszámítógépek (angolul mainframe gépeknek nevezik ezeket), melyek nevükkel ellentétben már korántsem voltak akkorák, mint az '50-es évek második generációs számítógépei. Ezek a harmadik generációs számítógépek már integrált áramköröket használtak, operációs rendszerük volt, és képesek voltak a párhuzamos feladatvégrehajtásra. Ezen számítógépek között teremtette meg a kapcsolatot az internet őskének tekinthető ARPANET hálózat. Ennek egyrészt gazdasági okai voltak (az erőforrások megosztása), részint katonai: ha a decentralizált hálózat egyes elemei kiesnek pl. egy bombatámadás következtében, maga a hálózat még működőképes marad. Senki ne gondolja azonban, hogy ezeken a gépeken csak „komoly” programok futottak! Több, azóta műfajteremtőnek tekinthető játék készült ezeken a gépeken, melyeket elsősorban diákok készítettek. Ezen játékok közül sok elveszett, egyrészt maguk a programírók nemtörődömsége miatt, részben a tanárok szigora miatt. Jó néhány viszont a hálózaton egyetemről-egyetemre „vándorolt”.

Nem is gondolnánk, hogy ezek programok némelyike az eredeti – vagy csaknem eredeti formában - Enterprise-on is futtathatók! Nézzünk meg ezek közül néhányat!

Az egyik legkorábbi mainframe gépre készült játék az 1969-es Sumer című program, mely később Hamurabi címen vált ismertté. Ebben egy uralko-

dó szerepébe kellett belélnünk magunkat, akinek évenkénti ciklusban kellett földet vásárolnia / eladnia, élelmeznie a lakosságot, megadni, mekkora területen vetünk. Ha jól csináljuk a dolgunkat, a lakosság nem éhez, sőt egyre többen érkeznek városunkba. A program Altair BASIC-ben és IS-DOS alatt Microsoft BASIC-ben is futtatható.

A játék a birodalomépítő játékok őse: sokan sokféleképpen egészítették ki. Megjelent benne az adóztatás, építkezés. A Feudal Overlord című 1984-es, jelentősen kibővített Spectrum változatot már némi karakteres grafika is „színesítette”, mely futtatható Geco Software-es emulátorán.

Az Apollo-11 űrhajósainak 1969. július 20-i holdra szállása ihlette meg Jim Storer középiskolai diákot, amikor megírta a Lunar Lander című programját. Az eredetileg FOCAL nyelven írt játékban épségben le kell szállnunk a Holdra. A program szöveges formában tájékoztatta a játékost a pillanatnyi sebességéről, magasságáról és a meglévő üzemanyagról. Ezt követően meg kellett adni, hogy a következő időciklusban mennyi üzemanyagot kívánunk elégetni. A játékból később BASIC változatot készített David Ahl, amit aztán sokan sokféleképpen módosítottak. Altair BASIC-ben mi is kipróbálhatjuk egy változatát.

A Star Trek univerzum egymástól függetlenül több embert is annyira magával ragadott, hogy játékot készítsen belőle. Az első filmsorozat 1966-1969 között készült, így még jóval a mozifilm megjelenése előtt is készült belőle játék. Az 1972-ben Don Daglow változata a szimulációt egyfajta történetmeséléssel vegyítette. Később számtalan változata született játéknak, Microsoft BASIC-ben mi is kipró-



bálhatunk pár változatot. Ezek közül az egyikben 1971-es évszám olvasható, a készítője pedig Bill Peterson.

Valószínűleg az egyik első stratégiai játék a Civilwar című program, amely az Amerikai polgárháború véres eseményeit eleveníti fel. A rendkívül egyszerű játékmenetű programot egyszerűen akár két játékos is játszhatja egymás ellen. Ebben a rendelkezésre álló erőforrások elosztása után csak a stratégiát kellett kiválasztani. Az első változata 1968-ban készült, a Microsoft BASIC-ben futtatható változatban 1973-as évszám olvasható.

A szöveges kalandjátékok előzményének tekinthető Gregory Yob 1972-ben készült Wumpus - más néven Hunt the Wumpus - című játéka. A mai szemmel már játéknak is alig tekinthető programban egy dodekaéder alakú labirintusban kellett vadásztunk egy Wumpus nevű szörnyre. Minden szobából három ajtó nyílik, a program az egyes szobákat sorszámmal azonosítja. Miután a gép közli, hogy melyik szobában tartózkodunk, felteszi a kérdést: „Shoot or Move?” Válaszként az M vagy S karaktert kell megadni majd a szoba számát. Óvatosnak kell lennünk, mert ha a szörnyel egy szobába kerülünk, felébred, ami ránk nézve elég káros következménnyel jár. A labirintus két termében gödör van, amibe beleesve veszítünk. Két termében pedig óriás denevér rejtőzik, amely felkap és a labirintus véletlen-

szerűen kiválasztott szobájába dob le (lehet, hogy épp az egyik gödörbe). Ha valamelyik közeli (szomszédos) szobában veszélyt érzünk (a Wumpust, árkot vagy denevért), a gép közli velünk, de azt nem, hogy melyikben. Ha rájövünk, melyik szobában bujkál a Wumpus nyilat kell lőni az adott helységbe. Ha tévedünk, a nyílvevő visszapatlanhat és minket találhat el... A program futtatható Altair BASIC-ből és Microsoft BASIC-ből is.

Az ARPANET hálózat kiépítést végző programozók közé tartozott William Crowther. Feleségével, Patriciával, aki ugyancsak programozó volt, szabadidejük nagy részét a barlangászatnak szentelték. Crowther másik szenvedélye a D&D (Dungeons and Dragons) szerepjátékok voltak, melyek 1974-ben indultak hódító útjukra. Amikor a Crowther házaspár magánélete válságba került, William elhatározta, hogy ír a lányának egy játékprogramot, mely ötvözi két szenvedélyét. Az 1975-1976-ban készített program nemes egyszerűséggel az Adventure címet kapta, azonban Colossal Cave Adventure-ként is ismert. A program szöveges leírást adott a helyszínről, és a játékos körüli eseményekről. (A CRT monitorok csak a '70-es évek közepétől terjedtek el szélesebb körben, így a korábbi játékok szükségszerűen lényegesen kevesebb információt közöltek a játékosal a nyomtatón keresztül). A különlegességet az jelentette, hogy a játékos a természetes nyelv szerkezetét utánzó igéből és főnévből álló parancsokkal irányíthatta a játékmenetet. A játék célja nem az ellenfelek legyőzése, hanem egy veszélyekkel teli barlang feltérképezése, elrejtett kincsek felkutatása.

Crowther idővel beleunt a program fejlesztésébe és a gépén hagyta a félkész játékot, mely ezután intézetről intézetre vándorolt. Don Woods, a Stanford Egyetem MI laboratóriumában szintén rábukkant a játékra és annyira megtetszett neki, hogy elhatározta folytatja a fejlesztését. Felvette a kapcsolatot Crowther-rel aki jóváhagyta Woods elképzeléseit. A Tolkien-rajongó Woods kedvenc regényei (A babó, Gyűrű ura) motívumaival egészítette ki, a játékban így nemsokára nyüzsgög-

Arpanet

tek a trollok, törpék, sárkányok és felbukkant egy hatalmas vulkán is. Életszerűbbek, olvasmányosabbak lettek a helyszínek leírásai, a játékos egy interaktív regényben érezhette magát. Az Adventure egy új műfajt teremtett, melyet stílusosan kalandjátéknak neveztek el. Az IBM ráharapott a programra, és hamarosan minden PC-hez, az MS-DOS 1.0 operációs rendszer mellé adott egy Adventure-változatot. A játék 1981-ben The Original Adventure címmel hivatalosan is megjelent. CP/M operációs rendszer alatt futtatható változatokat Mike Gotez készített a programból 1980-1982 között. A többszám azért indokolt, mert a játéknak addigra - a számtalan berhelés miatt - többféle (bővített) változata létezett. (Az egyes változatokat az elérhető pontszám alapján lehet megkülönböztetni a SCORE parancs kiadása után.) Pontot természetesen a kincsek megtalálásáért kapunk, illetve ha vissza is visszük őket a házba (ami előtt a kaland kezdődik) - hiszen egyszerre csak annyi tárgy lehet nálunk, amennyit elbírnunk. A DROP parancssal lerakott tárgyak az adott helyszínen maradnak, később újra felvehetjük ezeket.

A játék végeztével (beleértve azt a lehetőséget is, ha elhaláloznánk), a program a pontszámunk függvényében értékeli teljesítményünket. Természetesen legfőbb célunk a maximális pontszám elérésével az „Adventurer Grandmaster” titulus elérése.

A játékot a Level 9 Computing később kiadta Spectrumra, C64-re, CPC-re, MSX-re és Enterprise-ra is, ugyanilyen „minimalista” kivitelben, Colossal Adventure címmel, mely az Adventure Quest-tel és a Dungeon Adventure-rel a Middle Earth-trilógiát alkotja.

A legelső kereskedelmi forgalomba került kalandjáték azonban nem az Adventure, hanem a floridai születésű Scott Adams Adventureland-ja volt. A BASIC nyelven írt játék 1978-ban jelent meg, TRS-80-ra, majd Apple II gépekre. Bár a játék története, kivitelezése és így a szövegértelmező rutinja nagyon primitív volt, a játék unikumnak számított, így sikeres lett. Scott Adams a sikernek köszönhetően hamarosan létrehozta feleségével, Alexisszel saját cégét az Adventure Internationalt. Az Adventureland-et a következő években további 12 kalandjáték követte, melyek majdnem mind klasszikussokká váltak.

Az Adventureland egy az egyben az Adventure receptjét utánozza: kalandorunknak 13 mitológiai kincset (Blue Ox, Jeweled Fruit, Pot of Rubies, Diamond Ring, Diamond Bracelet, Magic Mirror, Gold Crown, Thick Persian



Rug, Firestone, Golden Net, Golden Fish, Royal Honey, Dragon Eggs) kell összegyűjtenie - így megszerezve a maximálisan elérhető 100 pontot -, miközben logikai feladványokat kell megoldania, tárgyakat használnia, szereplőket kell kicseleznie (tehát ebben a játékban sem a harc dominál). Az Adventure-hoz hasonlóan itt sem bírunk el akármennyi tárgyat, így a megszerzendő kincseket itt is egy helyre kell gyűjtenünk.

Az otthoni mikrogepek elterjedése után Adams játéka a Mysterious Adventures-sorozatot készítő Brian Howarth - igen jó minőségű (nem BASIC) - átírtaiban is megjelentek, többségük grafikát is kapott. Ezek közül a Spectrum-változatok futtathatók Geco Software-es emulátorán: Pirate Island, Secret Mission, Voodoo Castle, Savage Island 1, Savage Island 2, The Sorcerer of Claymorgue Castle.

Az egyik első játékcégét, az Infocom-ot 1979 júniusában alapította néhány egyetemista diák, Dave Lebling, Marc Blank, Joel Berez, professzoruk, Albert Vezza vezetésével. Az új cégnek köszönhetően ismerhette meg a nagyközönség az „interactive fiction” kifejezést, amit mi szöveges kalandjátéknak mondanánk. Első játékukat még 1977-ben kezdték fejleszteni a MIT PDP-10-es mainframe gépen (otthoni mikroszámítógépek még nem is léteztek). A Nagy Föld Alatti

Birodalomban játszódó kaland oly hatalmas lett, hogy három részben adták ki: 1980-ban napvilágot látott a Zork 1: The Great Underground Empire, 1981-ben a Zork 2: The Wizard of Frobozz és a trilógia befejező darabja a Zork 3: The Dungeon Master 1982-ben.

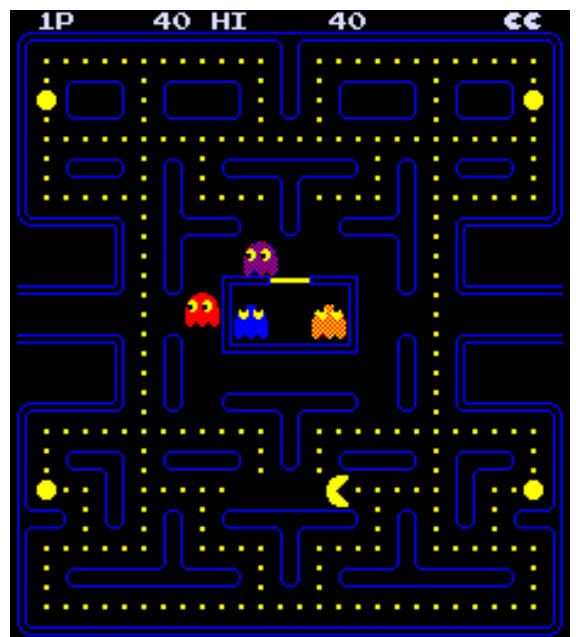
A cég megalakulásától kezdve hirdette az „ezer szó többet ér, mint egy kép” elvét, melynek hirdetésekben is hangot adott. Véleményük szerint az irodalmi szintű szövegek és egy jó szövegértelmezőnek köszönhetően a játékos képzelete keltette életre a tájakat, szereplőket. A gépek erősen korlátozott memóriáját pedig „fellesleges képekre pazarolni”. (Igaz, a Zork írásakor erre még nem is lett volna nagyon lehetőségük). Való igaz, a cég nem csak a levegőbe beszélt! Mark Blank és Dave Lebling a Zork megírásakor nem kisebb célt tűzött ki maga elé, mint hogy megalkossák az addigi legjobb szövegértelmezőt. Véleményük szerint „egy kalandjáték éppen annyira jó, mint a parserje (szövegértelmezője)”. Ez olyannyira jól sikerült, hogy az Infocom játékaik szövegértelmezője legendásan „okosabb” volt, mit az akár később megjelenő más kalandjátékoké. A legfanatikusabb kalandjátékosnak is „demoralizáló” lehet, ha találgatnia kell, milyen szóösszetételt fogad el a program. Az Infocom játékaik viszont sokkal kevésbé állnak a játékos kreativitásának útjába. A csak szöveg alapú játékok

lehetővé tettek egy érdekes programozási újítást: a Joel Berez és Marc Blank által kifejlesztett „virtuális számítógép” - melyet a programozók Z-Machine névvel illettek - lehetővé tette, a platformok közötti egyszerű átjárhatóságot. A cég játékaik Z-kódban futottak, mindössze a játékot futtató virtuális gépet kellett a hardware-hez külön-külön elkészíteni.

A cég játékaik így gyorsan megjelenhettek a ,80-as évek elején elterjedő 8-bites rendszereken, többek között CP/M alatt futtatható változatban is, amely az Enterprise tulajdonosoknak is örömdetes tény...

A Zork trilógia le sem tagadhatja, hogy az Adventure ihlette: A kalandok helyszíne a Nagy Föld Alatti Birodalom, ahová egy elhagyatott erdőszerű házikóban felfedezett csapóajtón át lehetett lejutni. A játék kezelése hagyományos módon történik. Az ige-főnévi kapcsolatokon túl azonban egészen bonyolult mondatstruktúrákat is képes értelmezni (pl.: PUT TORCH AND DIAMOND IN BASKET), persze nem szükséges feltétlenül ilyen „cirkalmasan” fogalmazni.

Hogy némi animációt is lássunk a monitoron, műfajt kell váltanunk: az egyik legismertebb arcade játék a PacMan. Igaz, Pacman klónból



rengeteg van, 2012-ben azonban Simon Owen készítette SAM Coupéra és Spectrumra Pacman emulátort. A SAM Coupé verziót SyX konvertálta át CPC-re, amit pedig Geco Ep-re... Az Emulátor „titulus” szó szerint értendő: a program az eredeti arcade ROM felhasználásával készült (a játékautomatában ugyanis szintén Z80-as processzor dolgozott)! Ebből már sejthető, hogy a program megtévesztésig hasonló az arcade játékhoz, Látványban, hangokban, játszhatóságban ugyanazt az élményt adja, mint az eredeti változat!



Boot szektorológia



Írta: Németh Zoltán
(Zozosoft)

Így néz ki az EXDOS boot szektor:

00-02	EBH, FEH, 90H IBM (Ugrás a betöltő kódra)
03-10	„EXDOS 1.0” rendszer (formázó software)
11-12	512 byte / szektor
13	szektor/cluster
14-15	boot szektorok száma (általában 1)
16	FAT példányok száma (általában 2)
17-18	főkönyvtár bejegyzések száma
19-20	lemez szektorainak száma
21	formázási típusbájt (F8h-FFh)
22-23	egy FAT példány szektorainak száma
24-25	szektor/sáv
26-27	fejek száma
28-29	rejtett szektorok száma
Innentől csak EXDOS (vagy VT-DOS) lemezeken:	
30	C9H RET, az MSX-DOS BOOT program belépési pontja
31-63	0 nullázás
64-69	„VOL_ID” azonosító string
70	0 UNDEL jelző-byte
71-74	32 bites lemezazonosító
75-99	0 nullázás
100-511	E5H üres terület

A 03-29 terjedő terület megfelel a FAT szabványban leírtaknak, most inkább a körítéssel foglalkoznánk: A 00-02 címen található egy x86-os ugróutasítás, amire az EXDOS-nak nincs semmi szüksége, de kompatibilitási okokból oda teszi, mivel egyes Microsoft operációs rendszerek - a saját szabványuktól eltérően - ellenőrzik ennek a meglétét is. Ebbe futottak bele az Atarisok, mivel a korai TOS verziók nem írták ezt oda, így a PC-k nem olvasták az Atarin formázott lemezeket.

30-as címen az MSX tulajdonosokra gondoltak, hogy egy véletlen meghajtóban felejtett EP lemez esetén nem okozzon elszállást a bootolási kísérlet, ezért egy RET utasítást tettek erre a belépési pontra. 64-74 bájtokon kerül el az EXDOS saját extrái: 64-69 bájtokon a „VOL_ID” azonosító jelzi, hogy ez egy EP lemez, és tartalmazza a következő 2 adatot. 70-en egy jelző bájt jelzi, hogy a lemez tartalmaz törölt, de helyreállítható

fájlokat. 71-74 bájtokon pedig egy véletlenszerű 32 bites lemezazonosító található, aminek lemezcsere ellenőrzésénél veszi hasznát.

Az EXDOS megalkotása óta eltelt több mint 25 évben újabb szabványok újabb adatokat pakoltak a boot szektorba:

28-31	rejtett szektorok száma (32 bit)
32-35	lemez szektorainak száma (32 bit)
36	fizikai meghajtó szám (PC) (00h floppy, 80h HDD)
37	CHKDSK jelzőbájt (Windows NT alapú rendszerek)
38	kibővített boot szektor jelzőbájt (29h) ez jelzi, hogy a következő paraméterek léteznek
39-42	32 bites lemezazonosító
43-53	11 karakteres lemeznév
54-61	7 karakteres fájlrendszer azonosító „FAT12” vagy „FAT16”
510-511	55h,AAh boot szektor jelzés

Ezeket az újdonságokat is célszerű lesz beépíteni az EP-s formázó programokba, a mai rendszerekkel való jobb kompatibilitás érdekében. Szerencsénkre ezek mind az EXDOS által kihagyott üres területre esnek, így egyszerűen csak ki kell bővíteni az eddigi EP-s boot szektort. Egyedül csak MSX-eseknek kedveskedő RET utasítás esik áldozatul a bővítésnek, de az is csak vinyón, 65535 szektornál többet tartalmazó partició esetén, ez talán túlélhető probléma.

Mint látható pár évvel az EXDOS után a Microsoft is megtalálta a 32 bites lemezazonosítót ezzel annyi a teendő, hogy az EXDOS által is használt 32 bites azonosítót beírni a PC-s helyére is, valamint 38-as bájtra írt 29h-val jelezni, hogy van ilyen érték. A fájlrendszer azonosító esetünkben „FAT12”.

A boot szektorba helyezett lemeznév az tulajdonképpen egy okos ötlet lett volna, hogy ne a főkönyvtárban kelljen keresgélni a lemez nevét. Viszont kipróbáltam MSDOS 6.2 és XP alatt is, hogy ettől függetlenül a főkönyvtárban lévő lemeznevet használja, ha nincs olyan akkor se veszi elő a boot szektorban lévő. Így nyugodt szívvel figyelmen kívül hagyhatjuk ezt a mezőt. Majd egyszer bele lehet rakni a VOL parancsba, hogy ide is írja be. Utolsó teendő még a boot szektor végére elhelyezni az 55h,AAh bájtokat. Nem létszükséglet, de ki tudja mit ellenőriznek egyes kreténebb PC-s rendszerek (ahogy az már pl. az ugró utasításról, vagy a formázó program nevééről kiderült).

```

EXDOS  command interpreter
:help exdos
EXDOS  version 1.0
Enterprise disk system
  8 1985 Intelligent Software Ltd.

commands available:
CD MD RD DEL DIR CLS ERA REM REN VAR
VOL COPY DATE ECHO LOAD MOVE TIME TYPE
CHDIR ERASE EXDOS ISDOS MKDIR MVDIR
PAUSE RMDIR RNDIR FORMAT RENAME
RAMDISK
:

```

És van még egy dolog, a lemez elején EBh. FEh, 90h az egy önmagára ugró JMP utasítás PC-n, ami azt jelenti, hogy ha egy EP-s lemezt felejtünk a meghajtóban, és arról bootol a PC, akkor jól lefagy... végülis kárt nem okoz ez a megoldás, csak bosszantó, hogy resetelni kell. Egyszerű megoldás, ami nem pazarol sok helyet: az EP-s lemez azonosító után pl a 75-76 címekre CDh,19h bájtokat kell írni, majd az ugró utasítást a lemez elején EBh,49h,90h-ra módosítani. Az új utasítás amire ráugrik, az egy INT 19h, ami újraindítja a bootolást, azaz így újra meg újra olvassa a floppyt, azonban amikor észrevesszük a bent felejtett lemezt, és kivesszük, megy tovább rendesen, nem kell resetelni. Ha nem sajnálunk több bájtot elpazarolni az EP-s programunkban, akkor egy hosszabb kóddal az is elérhető, hogy ki se kell venni a lemezt, az tölti be a vinyó boot szektorát (szintén 4Bh-re helyezve):

```

db 31h,0C0h ;xor ax,ax
db 8Eh,0D8h ;mov ds,ax
db 8Eh,0C0h ;mov es,ax
db 0FCh ;cld
db 0B9h ;mov cx,
dw 100h ; 100h
db 0BEh ;mov si,
dw 7C00h ; 7C00h
db 0BFh ;mov di,
dw 8000h ; 8000h
db 0F3h,0A5h ;rep movsw
db 0EAh ;jmp far ptr 800h:62h
dw 62h
dw 800h
db 0B8h ;mov ax,
dw 201h ; 201h
db 0BBh ;mov bx,
dw 7C00h ; 7C00h
db 0BAh ;mov dx,
dw 80h ; 80h
db 0B9h ;mov cx,
dw 1 ; 1
db 0CDh,13h ;int 13h ; DISK - READ SECTORS
INTO MEMORY
db 72h,5 ;jb +5
db 0EAh ;jmp far ptr 0:7C00h
dw 7C00h
dw 0
db 0CDh,19h ;int 19h ; DISK BOOT

```

Az EXDOS 1.4 az EPDOS 1.7 2012-es, és a FAFO 2.5 már a modernizált boot szektort készíti formázáskor.

FILE

HSOFT ezen apró rendszerbővítőjét sokáig nem használtuk, pedig nagyon praktikus program! Valószínűleg ebben az is szerepe játszott, a talán túl tömör dokumentációból nem derült ki, hogyan is lehet használni, pl. BASIC-ből. Az eredeti funkciója az, hogy programból hívva visszaadja a kiválasztott fájl nevét, vagyis bármilyen programból kényelmesen választhatunk ki pl. egy megnyitandó file-t. Később ZozoSoft fejlesztette tovább igazán univerzálissá, így már START-menü programként is könnyen használhatóvá vált.

Meghívása után egy file választó menü jelenik meg a képernyőn, az aktuális meghajtó tartalmával. Kezelése teljesen egyértelmű: a file-ok és alkönyvtárak között a beépített botkormány le-fel irányával lépkedhetünk. SHIFT+fel/le - lapozás, ALT+fel/le - ugrás a lista elejére / végére, ENTER vagy SPACE: file kiválasztása. (A program kezeli az alkönyvtárakat is.) A STOP megnyomásával meggondolhatjuk magunkat (nem választunk ki file-t). Meghajtót a kívánt meghajtó betűjelének megnyomásával válthatunk. A program eredetileg csak A-E meghajtókat kezel, az újabb verziók az EXDOS-tól kérdezi le a létező meghajtókat.

A program meghívása után - eredetileg - egy a 0. lapon lévő 256 byte-os pufferben helyezi el a kiválasztott file nevét (és elérési útvonalát). A programot ebben a formájában az alábbi alakban kell meghívni:

FILE word\$(puffer)ösvény/opció

Ahol az ösvényben megadhatunk névszűrőt is (pl. *.com). Az opció lehet: H - rejtett file-ok megjelenítése, S - rendszerfile-ok megjelenítése.

Mivel file kiválasztáshoz ez a legkulturáltabb megoldás (teljesen bolondbiztos, és felhasználóbarát), BASIC programból is érdemes használni, melyre alább láthatunk egy példát:

```

100 ALLOCATE 256
110 CODE A=" "
120 LET C$="FILE ,&CHR$(MOD(A,256))&CHR$(INT(A/256))
130 EXT C$
140 LET L=PEEK(A)
150 LET N$=""
160 FOR I=1 TO L
170 LET N$=N$&CHR$(PEEK(A+I))
180 NEXT
190 PRINT "file=" "&N$&"

```

A kiválasztott file neve az N\$ változóba kerül.

Ha például csak a .com kiterjesztésű file-okat szeretnénk megjeleníteni, a 120-as sor így módosul:

```
120 LET C$="FILE ,&CHR$(MOD(A,256))&CHR$(INT(A/256))&"*.COM"
```

Ha teljesen precíznek akarunk lenni, megszakításkezelésből (HANDLER) hívjuk meg a bővítést, hogy ha nincs a rendszerben akkor se keletkezzen hiba, ekkor hagyományos INPUT-ot használhatunk. Fontos tudni, hogy ha a STOP megnyomásával térünk vissza a bővítőből, az a BASIC programban is megszakítást okoz, amit szintén célszerű lekezeln.

A program eredeti funkcióját egészített ki ZozoSoft: ha paraméter nélkül hívjuk, akkor elindítja az adott programot (EXOS modul betöltés funkcióval). Ha így nem sikerült, akkor megpróbálkozik az EPDOS START parancsával, így BASIC programot is tud indítani (ha van EPDOS a rendszerben). Ebből a programindító FILE-ből készül külön betölthető START program, amit lehet lemezekre tenni. A korábbi programtöltőkkel szemben EXOS kompatibilis (EP64-en is megy), valamint kezeli a HDD-t, SD-t is.

Az irányítást egészítette ki István EnterMice egérkezeléssel. Működik a görgő is, bal gomb kiválaszt, jobb gomb CD .. (egy könyvtárral feljebb lép).

SIDBASIC



Írta: Persa Noel
(Geco)

A SIDBASIC legfrissebb verziója, ami egy szoftveres SID lejátszó, konvertált SID fájlok lejátszására Enterprise 128 gépen, hardveres bővítés nem szükséges.

A szoftver EXOS kompatibilis, ha a gép rendelkezik fájlkezelő eszközzel (lemezvezérlő, SD kártya), akkor az M64 fájlok egy fájlkezelőből választhatók ki, a magnós konfigurációval rendelkezők tölthetik a magnóról a kiválasztott M64 fájlt.

A lejátszást 128KB-os gépen a lejátszás közbeni 8Kb-os blokkonként kicsomagolás teszi lehetővé.

Megkötések: maximum betölthető M64 fájl méret 24320 bájt (5f00h), nincs SYNC, felül-, és alul áteresztő szűrő emuláció, SID digi effektek nem játszhatók le

Kombinált hullámformák nem működnek megfelelően, ilyenkor egyszerűen zaj > fűrés > háromszög > négyszög prioritással történik a konverzió, a valódi hardver ettől eltér.

Vezérlés:

- 1 - zene lejátszás Dave chipen
- 2 - zene lejátszás külső 8bit-es DAC kártyán, ha van a géphez csatlakoztatva
- 3 - raster bar be (csak 50Hz-es sebességű zenéknél működik)
- 4 - raster bar ki
- Esc - fájl választás

```
Volume in drive A: is A:\
Directory of A:\

 15  EXPFIST3  M64      1543
 16  GARFIEL  M64      5949
 17  GHOSTBU  M64      3926
 18  GHOSTSN0 M64      7310
 19  GOLDENA  M64      9181
 20  GOLDENA1 M64      3617
 21  HVSC^1   M64     23909
 22  IKPLUS   M64     16930
 23  IRONLOR  M64     12033
 24  IRONLOR0 M64     2394
 25  IRONLOR1 M64     4342
 26  KRAKOUT  M64     5447
 27  LN2_2    M64     10621
 28  LN2_4    M64     8174
 29  LN2_C    M64     10021
 30  LN3_1    M64     7394
 31  LN3_2    M64     5484
 32  LN3_3    M64     5633
 33  LN3_8    M64     6189
 34  MONTYON  M64     9885
 35  MYTH1    M64     7946
```

Néhány szó az M64 fájlokról:

A formátum leírása: 16384 byte-os blokkokat használ, minden blokk a 25 írható SID regiszter 655 értékét tárolja, az első 655 byte a 0. regiszter, a következő 655 byte az 1. regiszter, és így tovább. Az utolsó 9 byte általában 0, az alábbiak kivételével:

- az utolsó blokkban 3FFE-3FFF pozíciónál a ténylegesen használt megszakítások száma található, ami kevesebb lehet 655-nél

- az első blokkban 3FFD-nél a megszakítás frekvencia - 50 Hz található (így 50-305 Hz tartomány lehetséges), 3FFB-3FFC-nél pedig a megszakítások közötti SID ciklusok pontos száma CIA időzítésnél (CIA számlálóba írt érték + 1). Video megszakításnál ez 0, és az alapértelmezés 312 * 63 (PAL 50 Hz) vagy 262 * 65 (NTSC 60 Hz). A sid.com csak a kerekített Hz értéket veszi figyelembe

A 3-as csatorna regiszter (PWM első 4 bit) eredetileg nem használt 7. bitjének speciális funkciója van: azt jelzi, ha a burkológörbét újra kell indítani a GATE bit egy megszakítás ciklus alatti 1->0->1 átmenete miatt.

Köszönet Varga Istvánnak (IstvanV) az ötletekért, és a következő rutinokért, és alkalmazásokért, nélküle a SIDBASIC nem jöhetett volna létre:

Rutinok:

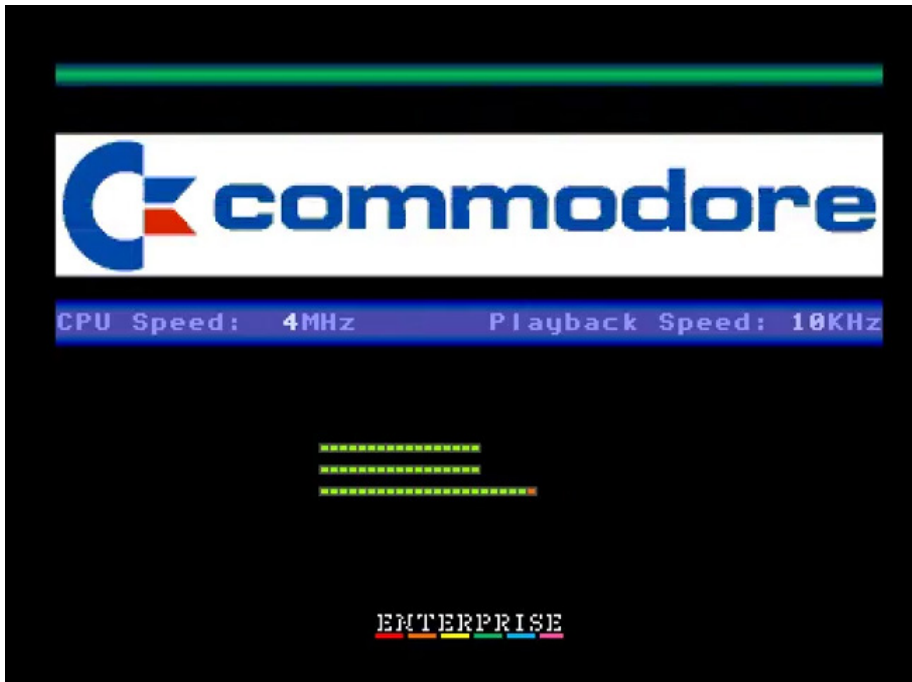
8 KB blokkos kicsomagoló rutin megszakításban Digi lejátszó rutin ideális SID hangminták generálása

Alkalmazások:

- epcompress - Enterprise csomagoló program
- epimgconv - Enterprise képkonvertáló program
- sid_dump - „nyers” SID regiszter adat kinyerő program SID fájlokból
- sid_conv - „nyers” SID regiszter adat konvertáló program M64 formátumba

Használat:

```
sid_dump.exe INFILE OUTFILE
[LENGTH1 [LENGTH2...]]
sid_dump.exe INFILE OUTFILE
[SONGLENGTHS FILENAME]
```



sid_conv INFILE OUTFILE [INTFREQ [BLKSIZE [NOADSRBUG]]]

INTFREQ (default: -1): interrupt frequency, if the value is ≤ 0 then it is read from input file, otherwise the value (Hz) is used given in command line. In case of -2 doubles the frequency of envelope emulation, which increase the size of output file with about 40-70%.

BLKSIZE (power of 2 in range 256-16384, default: 8192): size of a block of compression, size of dictionary is double of block size (-blocksize BLKSIZE -maxoffs BLKSIZE*2). 0 or negative value also sets the default 8K.

NOADSRBUG (0 or 1, default: 0): if this value is not 0 then it resets the counter of SID envelope emulation 15 bit timer at rising edge of GATE bit. Real Hardware works differently, so in this case the conversion can be worse also.

A sid_dump PSID file-t futtat minimális emulált 6502-es környezetben, és a kimenete egyszerű „nyers” formátum, ami a SID regiszterekbe írt értékeket tárolja 50 Hz-es (vagy egyéb frekvenciájú) megszakításonként.

Az ilyen file megfelelő tömörítés után lejátszható a sid.com (lent) használatával, ha van SID kártya (azaz jelenleg csak emulátoron :)). A PSID -> RAW konvertálás egyébként a SIDPLAY programmal és a

sidrecn.lua scripttel is lehetséges, ha a sid_dump valamiért nem működik.

SID zenék például innen tölthetők le:

<http://www.hvsc.c64.org/#download>

Példa konvertálásra:

(az epcompress csak a sid.com-os lejátszáshoz kell, a sid_conv.exe a tömörítetlen formátumot is elfogadja):

```
..\sidconv\sid_dump.exe MUSICIANS/T/Tel_Jeroen/Cybernoid_II.sid
cybnoid2.raw DOCUMENTS/Songlengths.txt MUSICIANS/T/Tel_Jeroen/Cybernoid_II.sid
```

```
Name: Cybernoid II
Author: Jeroen Tel
Released: 1988 Hewson
Video standard: PAL
SID model: MOS6581
IRQ frequency: 50.12 Hz
Done converting track 1: 17343 frames
IRQ frequency: 50.12 Hz
Done converting track 2: 451 frames
```

(a következő lépés a SID.COM-mal való lejátszáshoz szükséges)
epcompress -raw -m0 -9 -blocksize 16384 -maxoffs 32768 cybnoid2.raw cybnoid2.raw

```
Compressing data
100%
```

```
* sid.com (3.37 kB. 106x13 - viewed 2 times.)
* sid.s (10.09 kB - downloaded 1 times.)
* decompress_m0_16K.s (10.27 kB - downloaded 1 times.)
* file.s (31.21 kB - downloaded 1 times.)
* mouse.s (1.47 kB - downloaded 1 times.)
```

A következő lépés a SIDBASIC.COM által támogatott és SID kártya nélkül is lejátszható formátum létrehozása, ami a sid_conv programmal lehetséges. Ennek a formátumnak a leírása:

- az elején 16 byte-os EXOS fejléc található: 00h, 4Fh, tömörített adat méret L, tömörített adat méret H, IRQ frekvencia L, IRQ frekvencia H, megszakítások száma (24 bit, az alsó 8 bit az első), 7 nem használt (mindig 0) byte

- az EXOS fejlécet követi a tömörített adat epcompress -raw -m2 -blocksize 8192 -maxoffs 16384 formátumban

- minden 8K-s blokk legfeljebb 682 megszakítás hosszúságú felvételt tartalmaz, csatornánként 4 konvertált „regiszterrel”. Az adat szervezése a blokkon belül hasonló a RAW formátumhoz. Az utolsó blokk végén (1FFE-1FFF) 682 - a blokkban ténylegesen használt megszakítások száma található

- a 0. és 1. csatorna regiszter a 16 bites SID frekvencia, zaj hullámformánál 4 bittel jobbra léptetve

- a 2. regiszter a gyűrű modulációt (7. bit), hullámformát (5-6. bit, 00=háromszög, 01=fűrész, 10=négyszög, 11=zaj), és az aktuális hangerőt (0-4. bit) tartalmazza. 0 hangerő esetén a hullámforma mindig négyszög

- a 3. regiszter a négyszögjel kitöltési tényezője (felső 8 bit), csak nem 0 hangerejű négyszögjelnél, egyébként 0

Példa konvertálásra:

```
..\sidconv\sid_conv.exe cybnoid2.raw cybnoid2.m64
Converting file...
100%
Compressing data
100%
```

ADVENTURE

```

Welcome to the *new* Adventure!      Say "NEWS" to get up-to-date
game details.

Would you like instructions?y

Somewhere nearby is Colossal Cave, where others have found fortunes in
treasure and gold, though it is rumored that some who enter are never
seen again.  Magic is said to work in the cave.  I will be your eyes
and hands.  Direct me with commands of 1 or 2 words.  I should warn
you that I look at only the first six letters of each word.
(Should you get stuck, type "HELP" for some general hints.  For info-
mation on how to end your adventure, etc., type "JHFD".)

      _ _ _
If you have any problems, please contact Mike Goetz at (212) 671-2498.

You are standing at the end of a road before a small brick building.
Around you is a forest.  A small stream flows out of the building and
down a gully.
>|

```

1982 - Michael Goetz kaland, szöveges

Az internet őseinek tekinthető, 1969-ben megszületett ARPANET (Advanced Research Projects Agency) hálózat (eredetileg katonai célú számítógép-hálózat, mely az amerikai egyetemek és kutatóintézetek között teremtett kapcsolatot) kiépítést végző programozók közé tartozott William Crowther. Feleségével, Patriciaival, aki ugyancsak programozó volt, szabadidejük nagy részét a barlangászatnak szentelték. Crowther másik szenvedélye a D&D (Dungeons and Dragons) szerepjátékok voltak, melyek 1974-ben indultak hódító útjukra. Amikor a Crowther házaspár magánélete válságba került, William elhatározta, hogy ír a lányának egy játékprogramot, mely ötvözi két szenvedélyét. Az 1975-1976-ban készített program nemes egyszerűséggel az Adventure címet kapta, azonban Colossal Cave Adventure-ként is ismert. A program szöveges leírást adott a helyszínről, és a játékos körüli eseményekről. A különlegességet az jelentette, hogy a játékos a természetes nyelv szerkezetét utánozó igéből és főnévből álló parancsokkal irányíthatja a játékmenetet. A játék célja nem

az ellenfelek legyőzése, hanem egy veszélyekkel teli barlang feltérképezése, elrejtett kincsek felkutatása. Crowther idővel beleunt a program fejlesztésébe és a gépén hagyta a félkész játékot, mely ezután intézetről intézetre vándorolt. Don Woods, a Stanford Egyetem MI laboratóriumában szintén rábukkant a játékra és annyira megtetszett neki, hogy elhatározta folytatja a fejlesztését. Felvette a kapcsolatot Crowther-rel aki jóváhagyta Woods elképzeléseit. A Tolkien-rajongó Woods kedvenc regényei (A babó, Gyűrű ura) motívumaival egészítette ki, a játékban így nemsokára nyüzsögtek a trollok, törpék, sárkányok és felbukkant egy hatalmas vulkán is. Életszerűbbek, olvasmányosabbak lettek a helyszínek leírásai, a játékos egy interaktív regényben érezhette magát. Az Adventure egy új műfajt teremtett, melyet stílusosan kalandjátéknak neveztek el. Az IBM ráharapott a programra, és hamarosan minden PC-hez, az MS-DOS 1.0 operációs rendszer mellé adott egy Adventure-változatot. A játék 1981-ben The Original Adventure címmel hivatalosan is megjelent. CP/M operációs rendszer alatt futtatható változatokat Mike Goetz készített a programból 1980-1982 között. A többszám azért indokolt, mert a

játéknak addigra - a számtalan berhelés miatt - többféle (bővített) változata létezett. A játékot a Level 9 Computing kiadta Spectrumra, C64-re, CPC-re, MSX-re és Enterprise-ra is, ugyanilyen „minimalista” kivitelben, Colossal Adventure címmel, mely az Adventure Quest-tel és a Dungeon Adventure-rel a Middle Earth-trilógiát alkotja.

A letölthető csomagban négyféle verzió érhető el:

A02 - Ebben a verzióban 350 pont gyűjthető. Ehhez a változathoz FORTRAN forráskód is van.

B00 - Ebben a verzióban 550 pont gyűjthető. A labirintus így közel kétszer akkora és új szörnyek is vannak benne.

B01 - Ugyanaz az 550 pontos verzió, mint a B00, de ez Z80-as és 8080-as processzoron is futtatható.

B02 - Néhány új helyszínek köszönhetően ebben a változatban 580 pont érhető el.

A játék kezelését segítő parancsok:

SAVE - játékállás kimentése, kilépés a programból. Név megadására nincs lehetőség, egy lemezen egy játékállás lehet.

RESTORE - korábban elmentett játékállás betöltése.

QUIT - kilépés a programból.

INFO - A program kezelésének rövid leírása (csak a B... verziókban).

NEWS - verzió információ (csak a B... verziókban).

SCORE - jelenlegi pontszámunk lekérdezése.

INVENTORY - a játékosnál lévő tárgyak listája (nem bírunk el akármennyi súlyt!).

HELP - pár tipp a játékhoz.

FAST - a helyszínekről csak egy mondatos tömör és velős leírást kapunk (csak a B... verziókban).

BRIEF - az adott helyszín teljes leírását csak akkor kapjuk meg, amikor először járunk ott (csak a B... verziókban). FULL - mindig megkapjuk a helyszín teljes leírását (csak a B... verziókban). LOOK - újra megkapjuk a helyszín leírását.

Pontot természetesen a kincsek megtalálásáért kapunk, illetve ha vissza is visszük őket a házba (ami előtt a ka-

land kezdődik) - hiszen egyszerre csak annyi tárgy lehet nálunk, amennyit elbírnunk. A DROP paranccsal lerakott tárgyak az adott helyszínen maradnak, később újra felvehetjük ezeket. A játék végeztével (beleértve azt a lehetőséget is, ha elhaláloznánk), a program a pontszámunk függvényében értékeli teljesítményünket. Természetesen legfőbb célunk a ma-

ximális pontszám elérésével az „Adventurer Grandmaster” titulus elérése. A legfontosabb tipp a játékhöz: memóriabővítéssel rendelkező gépen RAMDISK-ből érdemes futtatni, mert így elkerülve az állandó diszkrpöge-tést sokkal gyorsabban fut a program! Persze a játékállás mentést kilépés után ne felejtjük átmásolni „valódi” lemezre.

GraCha

- avagy a grafikus karakteres szerkesztő EP-re



Annak idején sokféle játékot csináltam EP-re, amelyek nagyrészt grafikus-pixeles módot használtak. Ezért gondoltam, hogy meg kéne próbálni, mit lehet kihozni az EP-ből a keveset használt, általam grafikus-karakteresnek elnevezett módokban. Több ilyen mód van, sajnos szinte csak az a fajta használható, amelyben 2x4 színűnk van egy karakterben, és híres 16 felbontásunk. Bizonyos karakterkódok az első 4 szint, mások a második 4-et használják. (A z egy karakterben 2 szint, és összesen 4 szín párt használó, nagyobb felbontású karakteres mód is szép amúgy, de azért olyat jó pár program használt, bár jól és szépen csak Geco Panicmanja használta ki szerintem: http://www.ep128.hu/Ep_Games/Leiras/Panic_Man.htm) Elhatároztam tehát hogy írok egy editort, mert ugye semmiféle tool nem létezik EP-re, amelyben ez a mód kihasználható lenne. Az is célom

volt, hogy a végeredmény basic-ben használható legyen. Tehát az editor amellettt hogy BASIC-ben készült (bár a sebessége miatt csak emulátorban használható), az elkészült grafikát BASIC programként tudja kimenteni (pályakirajzoló, pálya adatok, karakter adatok).

Mivel az időm nem végtelen, ezért ez is korlátozó tényező volt, de éppen ezért próbáltam olyan programot írni, amelyik a fapadossága mellett hatékonyan használható.

A legnehezebb feladat az volt, hogy kitaláljam, hogy lehet legoptimálisan használhatóvá tenni ezt a karakteres módot. A 4 pixel szélesség kevés, ezért az editor 2x2 karakteres blokkokra épül, ezekből lehet pályát építeni és karaktert is 2x2-t lehet egyszerre editálni. Persze az 1 karakteres editálás is lehetséges.

A 2x4 szín is eléggé korlátozó tényező, erre azt találtam ki, hogy külön

paletta editor van, ahol 4 színű kis színcsoportokat tudunk szerkeszteni (de beépítve is van sok), és ezekkel színezni a képernyőt, soronként. A soronkénti színezés persze poke-okkal van, mert a basic és az Exos ezt alpból nem tudja, azaz nem lehet soronként más a paletta. (Ezt a 2x4 színű módot se támogatja, csak az 1x4 színűt, ezért tehát a videó mód is poke-olva van.) Így tehát elég könnyen be tudjuk színezni a pályáinkat (persze nem csak pályát lehet csinálni, hanem bármi mást is amire alkalmas ez a fajta karakteres stílus.)

A rajzolás munkamenete kb. így nézhet ki: rajzolunk karaktereket, azokból pályát építünk (pontosabban a 2x2-es blokkokból), és beszínezzük. Persze a sorrend nem kötött, lehet, hogy kell egy újabb karakter, és újra kell színezni a pályát is néha.

A háttér színezésére van egy funkció, ami random gradiens-t rajzol. Újra és újra meghívjuk ezt a funkciót, és elég hamar találni megfelelő háttér színeket. Az extra funkció között van egy olyan ami a sorokban a 0-ás szint átmásolja a 5-ösbe. Ennek az a célja hogy a két 4-es színcsoport karaktereinek a háttére ugyanaz lehessen. De érdekesebb persze úgy dolgozni hogy valamelyik színcsoport karaktereibe olyanokat rajzolunk amik teljesen kitöltik, hiszen így több színünk lesz, nem kell a háttérre elhasználni egyet. Ez így talán kicsit zavaros, de aki használja a rendszert, rá fog jönni, hogyan kell használni a programot.

Egy kihagyott bomba jó lehetőség

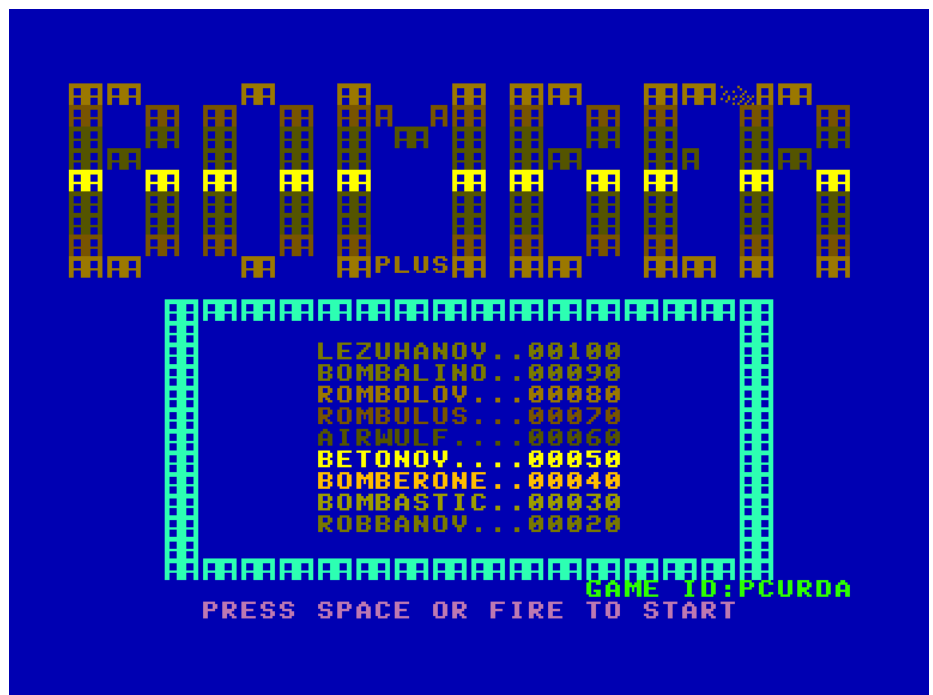


Írta: Bodnár Tamás
(Szipcsu)

A demokazetta játékprogramja, a Bomber jól játszható. Olyan, mint más számítógépek bombázó játéka, azoknál lényegében nem tud se többet, se kevesebbet. Az Enterprise adottságait sem használja ki jobban, kivéve a sztereó hangot.

A demokazetta programjainak célja azon túl, hogy használni lehessen őket, az lehetett, hogy a felhasználó kilitázhassa, módosíthassa, próbálkozhasson, mit és hogyan tud rajta változtatni. A programozók talán szándékosan nem hozták ki a legtöbbet ezekből a programokból, bár a maximálisan „kihajtott” programok látványa és hanghatásai sokat dobhattak volna, hiszen amúgy sem volt kapható túl sok és a gép adottságait is kihasználó program a géphez.

Az első probléma, amivel a felhasználó találkozhat a bombázó játék betöltése után, hogy a billentyűhang nincs kikapcsolva. Ezen könnyen túltehetjük magunkat, de sok játszás után előjönnek a bugok. Sok gyakorlás után végre le tudjuk dönteni az összes házát, ezen felvidulva a végén folyamatosan nyomva tartjuk a szóközt, és a semmi-be dobáljuk egymás után a bombákat, melyek zuhanásának hangja akkor sem hallgat el, ha a bomba már leért. Ha sok bombát dobálunk így le, megtelik a hangpuffer, és a játék megáll arra az időre, amíg fel nem szabadul. Ez egyszerűen orvosolható a programban a megfelelő helyen kiadott



CLEAR QUEUE n utasítással, vagy ha a bombazuhanást kiadó SOUND paramétereire beszúrunk egy INTERRUPT-ot is.

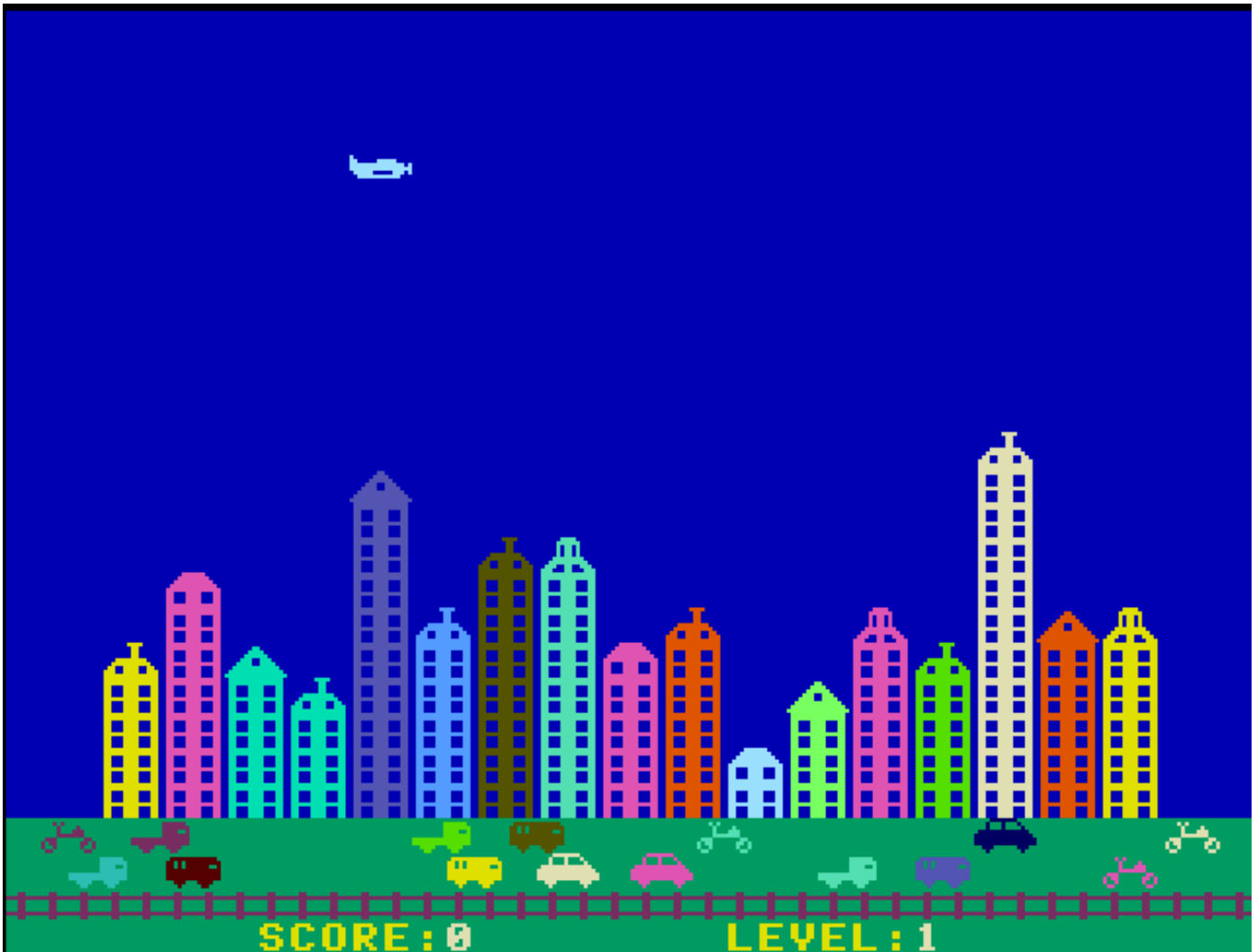
A nagy izgalmak közben talán nem tűnik fel, hogy ha már sűröljük egy háznak a tetejét, és akkor pottyantjuk rá a bombát, a repülő orra (első karaktere) eltűnik, mert a bomba, majd a szóköz felülírja, és amíg a ház omlik, addig „orratlan” a repülő. Nem csak ilyen izgalmas helyzetben, hanem minden bomba ledobásakor megfigyelhető, hogy a repülő orra egy pillanatra eltűnik, ami nem túl esztétikus. Ezen valószínű könnyű segíteni, ha a bomba kirajzolását végző ciklus egy karakterrel lejjebb kezdi a számolást, mint ahol a repülő van...

Ha egy magasabb épületet találunk el, és teljesen leomlik, az omlás eltart egy kis ideig. Előfordul, hogy ennyi idő alatt a repülő motorhangját megszóllaltató SOUND utasítás teljesen lecseng. Ezen is könnyű segíteni, ha

a DURATION utáni számot megnöveljük.

Nem tudni, szándékosan hagyták-e ki a program írói a RANDOMIZE parancsot a program elejéről. Ez lehet szándékos is, ezért minden induláskor a program ugyanúgy állítja be a házak magasságát, ugyanazokat a pályákat kapjuk. Ez lehet jó is azoknak, akik gyakorolni szeretnének, de változatosabb lehetett volna a játék, ha mindig más a pálya.

Kicsit talán meglepő, hogy a játéket a program a 0-ás csatornára írással tölti ki. Írhatna a 102-es csatornára is, így a játék elején a házak kirajzolása gyorsabb is lehetne. A házak kirajzolását más módon is lehetne gyorsítani, így a várakozás kikapcsolása (OUT 191,12) és a megszakítások tiltása látványosan megnövelhette volna a sebességet. Annyi „előnye” lehet a 0-ás csatorna használatának, hogy ha a kezdő felhasználó a STOP-pal megállítja a játékot, kedve szerint rajzolgat



hat a pályára akár házakat, repülőket, a CONTINUE után pedig meglepődhet, hogy műalkotását a program teljesen figyelmen kívül hagyja. Így hiába törölünk le szóközzel a házak tetejéből, azok ott maradnak, és az általunk kirajzolt repülők sem szállnak be a bombázásba.

Viszont, ha a 102-es csatornát használná a program a rajzolásra, kicsit gyorsabb lenne, és így gond nélkül befért volna még egy szín használata. Mivel a házak magasságát a program tömbben tárolja (nem a képernyőről olvassa le), több szint is lehetne használni.

De még a 0-ás csatorna használatával is lehetett volna még egy szint is használni a házak kirajzolásához.

Arról nem is beszélve, hogy a programot szinte jelentéktelen többletmunkával meg lehetett volna csinálni úgy, hogy attribútum képernyőn fusson, így egyszerre akár 16 szint is lehetett

volna használni a képernyőn, ami rögtön más benyomást tett volna pl. a Centrum Áruházakban azokra, akik látják a programot.

A hangok is rendkívül egyszerűek. Amennyire szélesek a lehetőségek, annyira nem használ ki belőle a program semmit. Mindössze a bomba zuhanásának hangjához használ burkológörbét, illetve még a repülő hangjához is, pedig erre nem is lett volna szükség. Talán érthető, hogy nyitva hagyták a lehetőséget a felhasználó számára, hogy kísérletezéssel a hangokat is módosítsa. A Dave lehetőségei sokkal változatosabb robbanóhangokat és bombahangot tettek volna lehetővé a 3+1 csatornán, a burkolóról, a szűrőkről és gyűrűmodulációról nem is beszélve.

Nem létszükséglet, hogy egy játékprogramnak zenéje legyen, ami itt nincs is. Pedig az említett lehetőségeket kihasználva igen erőteljes hangzások felhasználásával is lehetett volna

a programba zenét tenni, ami szintén sokat dobott volna az első benyomáson a Centrum Áruházakban.

Összességében a programba bekerült néhány kisebb bug, és feltűnően minimalisták voltak a programozók. Szinte ugyanazt nyújtja a program, mint bármely más korabeli számítógép bombázó játéka, pedig nyújthatna sokkal többet. Talán meghagyták a lehetőséget a felhasználónak, hogy a bugokat kijavítsa, és a játékot továbbfejlessze? Talán a karakterdefiniáló program is azért került a bombázó mellé (ahogy talán más gépeken is), hogy a repülő alakját ábrázolhassuk tetszés szerint akár hajóra, csirkére, tengeri herkentyűre, a bombát lézerlövedékre, a házakat pedig szemétkupacra, vagy amire csak akarjuk - amit viszont elmenteni és visszatölteni nem tudunk.

Igaz, hogy itt, a nyomtatott újságban nem, de az elektronikus változatban színesben látható majd az összes kép. *(a szerk.)*

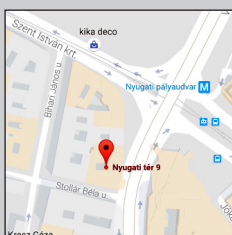
Enterprise Klub - 2017. április 15.



ENTERPRISE KLUB

Egy évben 6 alkalommal

Helyszín:
Skála terem
Budapest (V. ker.)
Nyugati tér 9.
14 órától 19 óráig



Információ: www.enterpriseklub.hu

Ha te is szeretnél Az ENTERPRESS Magazin szerkesztője lenni, küldj cikket, játékleírást, játékismertetőt, vagy bármit amely az Enterprise számítógéppel kapcsolatos!

A cikkeket erre az e-mail címre küldheted:

info@enterprise.news.hu

ENTERPRISE FOREVER

<https://enterpriseforever.com>

ENTERPRESS Magazin - 2017/2-3. március - június

Főszerkesztő: Matusa István

Szerkesztőségi főmunkatárs: Németh Zoltán (Zozosoft)

A csapat: geco, Povi, Kiss László, SzörG, szipucsu, lgb

Design, nyomdai előkészítés: Matusa István

Weboldal: <http://enterprise.news.hu>

E-mail: info@enterprise.news.hu

A lap időszakosan - korlátozott példányszámban - nyomtatott formátumban és elektronikus formában is megjelenik.

ENTERPRESS e-magazinok:

<http://enterprise.news.hu/index.php/magazin>