

ENTERPRESS

Magazin az ENTERPRISE felhasználóknak

2018/4-5. július – október

A SOUND

utasítás rejtelmei

pitch, duration, source, sync



Új buszbővítő
és MSX
Cartridge
Adapter

Készül az
EXDOS 3.0
(FAT16)

ETLAN ENTERPRISE

RAM 64K OR 128K, UP TO 3.9 MEGABYTES

ROM 32K, UP TO 3.9 MEGABYTES

STEREO SOUND, 4 VOICES, 8 OCTAVES

FILE: CES JAN.'84



WITH OBSOLESCENCE BUILT OUT

INTERJÚ INTERJÚ INTERJÚ INTERJÚ INTERJÚ INTERJÚ

Az Enterprise elbűvölt képeségeivel és könnyű bővíthetőségével

Interjú Maciej Gruszeckivel



Készítette: Bodnár Tamás
(Szipucsu)

Maciej Gruszecki (aki az EP fórumban „pear” néven szerepel) 2015-ben csatlakozott az Enterprise Forever csapatához és azóta sok figyelemreméltó hardveres fejlesztéssel örvendeztetett meg minket. Interjúnkban elmondja, hogyan ismerkedett meg az Enterprise számítógéppel és sok érdekes dolog is mesél még.

Hogyan kerültél kapcsolatba az Enterprise számítógéppel? Volt-e más számítógéped az Enterprise előtt?

Egészen öt évvel ezelőttig nem is tudtam, hogy létezik ilyen számítógép. Azt hittem, az Enterprise csak a Star Trek űrhajó neve.

Legjobb, ha az elején kezdem. Mindig is kíváncsi természetű voltam. Sok dolog érdekelt. Egyik hobbi követtem a másikat: játékkatonák, üvegkupakok, bélyegek, moziszterek, különféle címkék, autós brossúrák a nyugati országokból. Talán kimaradt valami és a sorrend sem pontos, de a legfontosabb, hogy legutoljára az autók voltak érdeklődésem középpontjában. Az autós prospektusokat nehéz volt megszerezni. Kellott hozzá ismerni valakit, aki külföldre utazott és hozott ilyen kincset.

A magazinokhoz könnyebben hozzá lehetett férni. Valamikor egy újságárus közelében laktam, így az újságok fotóit kezdtem böngészni. Később kivágtam ezeket a képeket és a füzetembe tettem. Néhány fotót a „Kaleidoskop Techniki” című, havonta megjelenő, gyerekeknek szóló újságból is kivágtam. Az egyik számban, egy autót ábrázoló kép túlfeloldán egy villogó villanykörte áramkörének a rajza állt. Ez 1983-ban volt, akkor 10 éves voltam. Az elektronika magával ragadott, és szenvedélyem lett mind a mai napig. Összegyűjtöttem minden, elektronikával foglalkozó magazint, ami csak elérhető volt. (Internet nem volt akkor még.) A számítógépekről szóló első hírek ugyanabban a magazinban jelentek meg, ezekre



1986 körül kezdtem odafigyelni. De tudtam, hogy nem engedhetem meg magamnak ezt a hobbit - a házi számítógépek nagyon elérhetetlenek voltak számomra. Még az elektronikai alkatrészek is hozzáférhetetlenek vagy túl drágák voltak. Mindenesetre ez még manapság sem olcsó mulatság.

Így először elkezdtem BASIC-ben programozni tanulni... egy papíron. Ez nem került semmibe. A tudásom szépen bővült, de még mindig nem volt felszerelésem. Tulajdonképpen még azt is tudtam, hogyan építhetek magam számítógépet. Egy közeli boltban egyszer láttam egy összerakható lengyel számítógépet -- a Meritumot. Ez olyan sokba került, hogy jó tízévesi zsebpénzemet kellett volna rá költenem.

Több mint egy évig tartott meggyőzni apukámat - ami végül sikerült -, így vett fel kölcsönt kifejezetten ilyen vásárlásra (ebben az időben ez felért egy féléves fizetéssel). A mágius '88. 8. 8. napon megkaptam első számítógépet, a **Timex TC2048**-at. Ez egy portugál ZX Spectrum

klón volt, az egyetlen, melyet lengyel zlotyiért el lehetett érni. Az összes többi gépet külföldi valutáért lehetett csak kapni. Ezen kívül még piaci árusoktól lehetett a külföldről hozott termékeket megvásárolni.

Sok időt töltöttem a számítógéppel, de nem játszottam. Mivel messze laktam a fővárostól, elég kevés játékhoz tudtam hozzájutni. Inkább saját programokat írtam, a programozást pedig könyvtári könyvekből tanultam. Ugyanabban az évben találkoztam először PC XT-vel az iskolában. PC-n Pascalt kezdtem tanulni.

Később aztán diplomáztam, megismertem a feleségemet, majd dolgozni kezdtem. Megszülettek a gyerekek, így hát a hobbijaim igencsak háttérbe szorultak.

2012-ben meghívtak a speccy.pl fórumra. Itt olvastam először 2013 elején az Enterprise-ről egy olyan topikban, mely a különféle országokban népszerű 8 bites számítógépekkel foglalkozott. Egy hónappal később az eBay-en vettem első Enterprise 128 gépemet.

Mennyire volt népszerű Lengyelországban az Enterprise számítógép? Lehetett kapni nálatok?

Az Enterprise gyakorlatilag ismeretlen és nem volt kapható Lengyelországban. A speccy.pl fórumon egy külön szekciót hoztam létre az Enterprise-os topikoknak és próbáltam népszerűsíteni a gépet, de mostanáig is csak kevés embert érdekel.

Ez a gép egyszerűen elbűvölt engem a képességeivel és a könnyű bővíthetőségével.

Milyen számítógépek voltak elterjedve nálatok a 80-as, 90-es években?

Akkortájt, amikor Nyugat Európában a 8 bites házi számítógépek egyre csak fejlődtek, Lengyelországban - a vasfüggöny mögött - alapvető problémákkal kellett megküzdenünk. Például probléma volt, hogy meg kell venni a vécépapírt, és ez teljesen kiszorította a számítógépet. Egyébként, Magyarországon is hasonló volt a helyzet?

– *Nálunk azért szerencsére ennyire nem volt kritikus a helyzet. Emlékszem, egyszer akkortájt Krakóban, Lengyelországban voltam, tényleg rossz volt a helyzet. Alig tudtak megkínálni minket étellel. (- a szerk., Tutus megjegyzése.)*

Na, szóval a 80-as évek első felében a számítógépeket privát úton importálták Lengyelországba, főleg Németországból. ZX Spectrum, Commodore C64, Atari 800XL, 65XE, Amstrad CPC gépeket adtak kizárólag cégeknek és a katonaságnak. A 90-es években a Berlini Fal lebontása után könnyen lehetett vásárolni 16 bites Amigákat, vagy az itt kevésbé népszerű Atari ST-t. Elkezdődött tehát a PC-s korszak.

Hogyan találtad meg a magyarországi Enterprise Forever fórumot?

Nos, nem én találtam meg a fórumot, hanem a fórum talált meg engem. Az eBay-en láttam az eredeti ZX Spectrum emulátort, de valami irreális áron. Azonban nagyon akartam egyet, mert a ZX Spectrumot már ismertem és

látni akartam, hogyan működik ez az emulátor valójában. Keresgéltni kezdtem a neten. Végül a **Zozo** által feltöltött fényképeket és kapcsolási rajzot találtam meg a ZX Spectrum emulátorról. Aztán kiderült, hogy szinte minden információ elérhető az Enterprise hardveréről... magyarul. A kapcsolási rajz viszont akkor is kapcsolási rajz, így végül valahogy sikerült megfejtenem.

Sokáig nem készítettem semmilyen elektronikát. Nem volt meg hozzá a felszerelésem vagy a megfelelő tervező programok. Így hát megkértem barátomat, Zaxont, hogy rajzoljon nekem és készítsen néhány darabot. Zaxon nagyon elfoglalt, sokféle tartozékot készít ZX Spectrumhoz. Egy évig türelmesen vártam, majd írtam neki: *„Tudod mit? Írd meg nekem, milyen eszközöket használsz és hogy hol lehet nyomtatott áramkört rendelni!”*

Majdnem egy hónapig tartott nyomtatott áramkörök tervezni az emulátorhoz és a buszcsatlakozóhoz. 2015 áprilisában be is üzemeltem a saját ZX Spectrum emulátoromat. Azonnal működött is és majdnem tízszer kevesebbe került, mint az eredeti, melyet az eBay-en találtam. A klónomat bemutattam a speccy.pl fórumon, és a Youtube-ra feltettem egy videót az emulátor működéséről. **Zozo** ezt megtalálta és küldött egy linket az enterpriseforever.com oldalhoz. Egy fórumtársam a speccy.pl-ről korábban már regisztrált a magyar fórumon és el is mondta, hogy az emulátor az én munkám. Én is elhatároztam, hogy regisztrálok oda - és a „magyaros” kalandozásom ezzel elkezdődött. Miért is ez a „magyaros” kalandozás? Amit csinálok, abból a legtöbb Magyarországra jutott el. Itt a legnépszerűbb az Enterprise.

Civilben mi a foglalkozásod?

Elektromérnök vagyok. Sajnos, amikor befejeztem a középiskolát, majd elektronikai tanulmányaimat az egyetemen, kiderült, hogy az elektronikai ipar majdnem teljesen eltűnt Lengyelországból, mivel a gazdaság átalakult.

Szerencsére ott volt a másik hobbi, a programozás. Így programozóként tudtam elhelyezkedni. Először ipari szoftvereket írtam, főleg az energiaágazatnak. Néhány évvel ezelőtt másik iparra váltottam és kórházi nyilvántartó programokat írtam.

Munka után nem volt sok időm, de a berendezések tervezése és megépítése számomra lazítás. Néha csak lazítani kell. Ennek az időnek egy részét retro számítógépek javításának szentelem. Minden évben egyre csak több és több van belőle, de a nap még mindig csak 24 órából áll.

Neked mi a kedvenc Enterprise-os témád?

Nos, a hardver – ugye nem meglepő? Egy nap programozás után a munkahelyen azt mondtam, nem akarom ezt többet. Valójában a szoftvereket is és az elektronikai rendszereket is számítógéppel készítik. De egy készülék megalkotása sokkal inkább elégedetté tesz engem. Ha kész a mű, hozzá lehet érni, meg lehet fogni. A program mégis csak valami virtuális dolog.

Jelenleg milyen retrós társaságok vannak Lengyelországban?

Számos aktív ZX Spectrum, Commodore, Amiga, Atari csoportunk van. Talán van más is, de a ZX Spectrum kivételével nem kísérem figyelemmel a többi platformot.

Retro számítógépes klubtalálkozókat szoktak tartani? Van-e nálatok hasonló számítógépes újság mint nálunk az Enterpress?

Az évszórán több nagyobb eseményünk is van: *Pixel Heaven*, *RetroKomp*, *Riverwash*, *Retrospekcie*, vagy a fórumunknak, a *specy.pl*-nek a találkozója. Gyakorlatilag minden hónapban van valamiféle találkozó. Újságokból jóval kevesebb van és a játszani vágyókat célozzák meg inkább (*Pixel*, *RetroKomp*, *Komoda & Amiga*).

Honnan jött az ötlet, hogy az Enterprise-hoz hardverkiegészítőket tervezz?

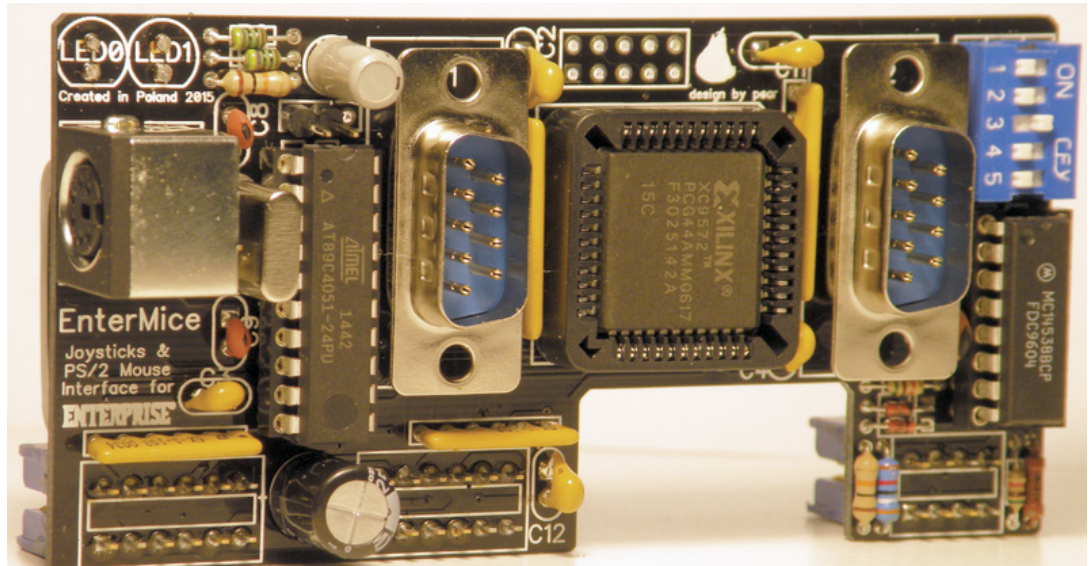
Hajtott a kíváncsiság, és hát a szegénység is. A kemény munkahelyi meló után felüdülés volt ilyenekkel foglalkozni. Ha anno meg tudtam volna venni az eredeti ZX Spectrum emulátort, akkor soha nem készítettem volna el a klónját, vagy bármi mást utána. Felbátorodva az első klón sikerén, elkezdtem nyomtatott áramköröket rajzolni a későbbi bővítésekhez, melyekhez dokumentációt találtam. Ezeket mind magam csináltam, nem akartam álltatni magam, hogy valaha meg fogom tudni venni, amit akarok. Főleg jutányos áron. Így készült a *FlexiBridge and the EXDOS*, *RAMFlashRTC and IDE/CF card series*.

Később kiderült, hogy nem én vagyok az egyetlen, akinek gondot okoz bármit is beszerezni az Enterprise-hoz. Így pár éven belül a hardvereim „pear” logóval elterjedtek egész Európában, Ukrajnától Nagy-Britanniáig, Norvégiától Olaszorszáig.

Az EnterMice nagy siker volt, ezt követte az EPower. Hogyan születtek ezek a bővítések?

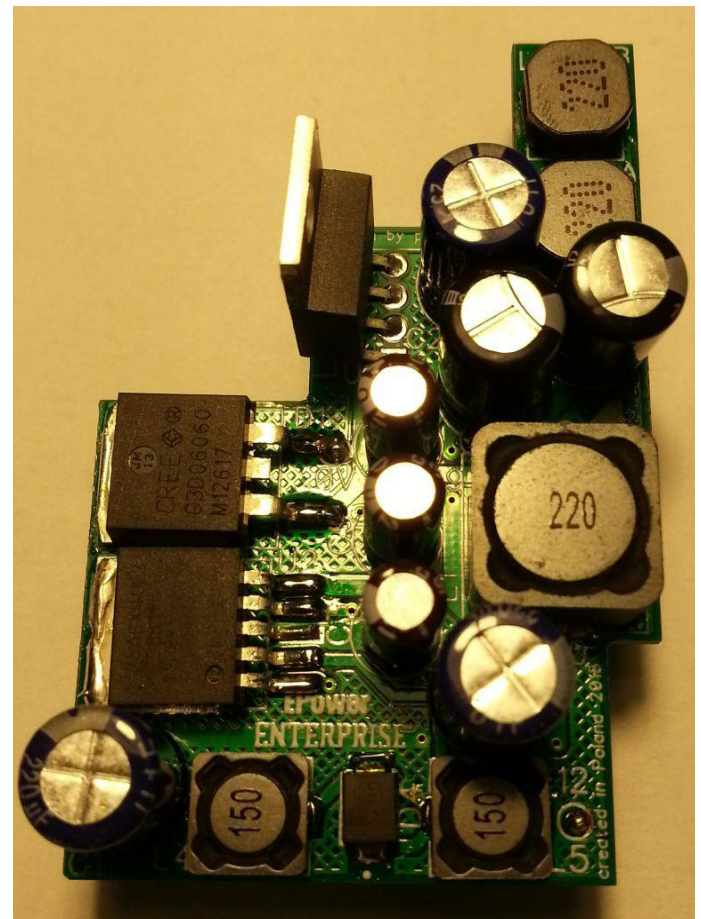
Az ötlet maga a legfontosabb része az egésznek. Ez egy szeletje a tortának.

Az EnterMice főleg Gustavo-nak (GFlorez) köszönhetően készült el. Egy hónappal a ZX Spectrum emulátor bemutatása után, 2015. május 11-én Gustavo írt nekem. Megkérdezte tőlem, nem építenék-e egy egér interface-t az Enterprise-hoz. Ő maga már dolgozott ilyen interface-en, az eredeti BoxSoft egérén. Kicsit féltem, mert évek óta nem terveztem semmit vázlatból scratch. Másrészt pedig szeretem a



kihívásokat, és Gustavo sem adta fel, így hát nekiláttam. Fél évvel később az első működő prototípusa az EnterMice-nak készen volt.

AZ EPower pedig Zozo ötlete volt. Idén január végén ezt írta: „Ez valószínűleg könnyű lesz neked.” Sokkal több önbizalmam volt már, mint az EnterMice-nál, és azt mondtam, miért ne. Több ötletem is volt a kivitelezésre és remélem, végül a legjobbat választottam. A tervezés egy hónap múlva kész lett, de tavaszig várnom kellett. A műhelyem egy fűtetlen garázs, ezért novembertől márciusig csak tervezem az eszközöket, vagy firmware-t írok. Áprilisban le is teszteltem az EPower. Nagyszerűen működött.



Mellesleg kiderült, hogy az Enterprise az új hálózati modullal majdnem fele annyi energiát fogyaszt, mint ahogy azt előtte gondoltuk. Számíthattam erre, de a végeredmény mégis meglepett. Nevettem, hogy ez az első ökológiai eszközöm.

Milyen együtt dolgozni Gflorez-szel, az Enterprise spanyol mágusával?

Gustavo különleges ember nagy szenvedéllyel, de nagyon szerény is – túlságosan is. Ahogy már említettem, az EnterMice ötlete az övé volt, és neki köszönhető, hogy elkészítettem. Ő figyelemmel kísérte az egész tervezési folyamatot, majd a megjelenést és a működést is. Végig motivált engem és kérdeztetett, hogyan haladok. Egyszer projekt-menedzsernek is neveztem, de ő nem engedte, hogy ezt a kifejezést használjam rá.

Gustavo és Zozo is EnterMice teszterek voltak. Nekik köszönhetően ért el a firmware olyan sok funkciót, melyeket ma is használunk. Gustavónak úgy általában véve sok érdekes ötlete van Enterprise-hoz, amiket el is kezd megvalósítani (egy példa: M-Slot adapter az MSX interface-hez).

Mindannyiunk nagy álma a DreamTurboCard Enterprise-hoz. Hogyan állsz jelenleg vele, mikor várhatjuk megjelenését?

Az Enterprise-hoz való „extra hajtóműről” szóló tárgyalásoknak már jó pár év történelme van. Ebbe én az előző év közepén kapcsolódtam be. Nagyon hosszú ideig a megvalósításhoz jöttek ugyan az ötletek, de mindet el is vetették, sokkal inkább, mint az elmúlt fél évben. Végül egy CPLD megoldás mellett döntöttem. Alaposan elmélyedtem a VHDL tanulmányozásában. Életem legelső VHDL implementációja kicsivel az EnterMice előtt ké-

szült. Nem volt hát túl sok tapasztalatom, ez a projekt pedig sokkal összetettebb. A kártya prototípusát márciusban adtam ki. Nem tudtam megvárni a tavaszt, összeraktam a nyomtatott áramkört, mihelyt a hőmérő higanyszála 5 fok közelébe ért a garázsomban. Az első teszt sikeres volt, el is kezdtem készíteni a firmware-t.

Közben azon is gondolkodtam, vágjak-e lyukat, vagy ne az LCD kijelző számára az Enterprise burkolatán. Ugyanekkor zajlott egy eszmecsere a fórumon a külső videóforrások csatlakoztatásáról is. Kitaláltam hát egy OSD modul az LCD kijelző helyett. Most is éppen dolgozom rajta. Jelenleg a lehető legegyszerűbben próbálom a képet elérni, közben épitem a DreamTURboCard-hoz illő modult, melyet már befejeztem.

Sok munka vár még rám, de remélem, jövő év végére a Turbo kártya készen lesz.

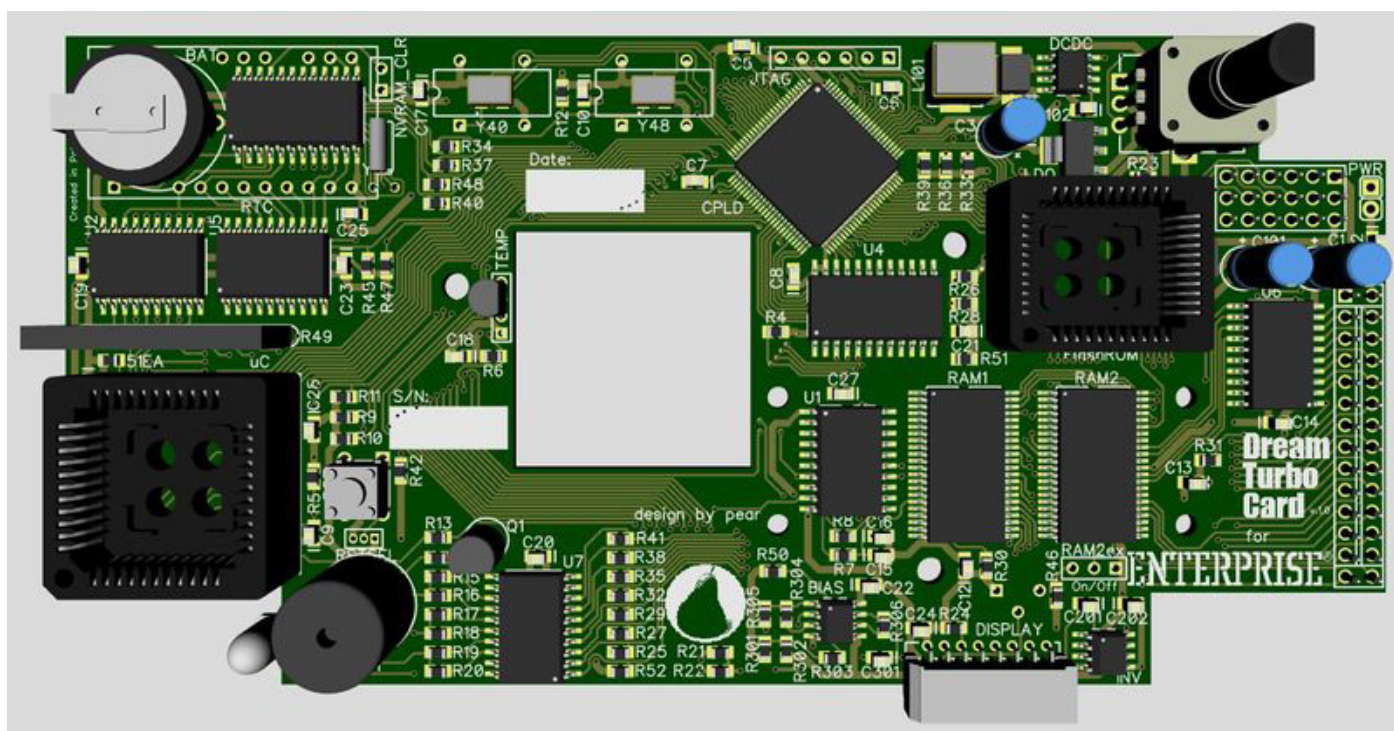
Szerinted milyen fontos hardver kiegészítések hiányoznak még az Enterprise-hoz?

Ezt a kérdést inkább a szoftverkészítőknek kellene feltenni. Tudom azt is, hogy nem minden fejlesztőnek van szüksége az igazi hardverre, sokan inkább az emulátort használják. Ezt meg is értem, mert sokkal kényelmesebb, de nem adja vissza az igazi retro „hőzzáerek, megfogom” érzést.

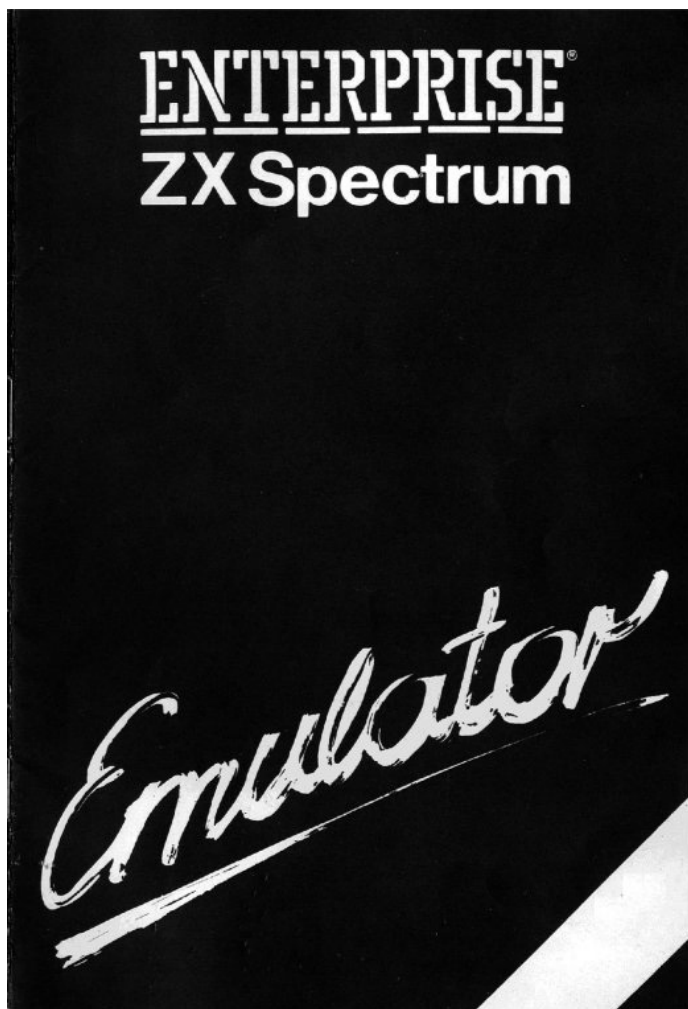
Ha van ötlet, szívesen fogadom. Olyan tartozékot készíthetek, amelyet ti javasoltok. Bár én jobban hiányolom a FAT16 támogatást, mint valami más hardvert, és örülök, hogy Bruce Tanner dolgozik is rajta.

Mik a terveid a jövőre vonatkozóan Enterprise témában?

Végül, hogy legyen időm játszani az Enterprise-ommal.



Halló! Itt az emulátor?



Emulátornak hívják azt az eszközt amely egy másik berendezés működését képes utánozni, annak ellenére, hogy felépítése teljesen különböző. A Spectrum emulátor egy olyan szerkezet amely a Spectrumtól eltérő hardware és software környezetben képes futtatni Spectrum programokat. Az Enterprise esetében ez annyit jelent, hogy egy ún. emulátor kártya csatolásával, ennek tulajdonosa Spectrum programokat is tud futtatni a számítógépén. Minden emulátorral felmerül azonban egy lényeges kérdés: Mennyire tökéletes az utánpótlás?

Amikor az Enterprise-ok hazánkba érkeztek, beharangozták, hogy a Spectrum emulátor hamarosan kapható lesz, még árat is mondtak). Annál is inkább, mivel egy magyar szabadalomról van szó. Eltelt egy év, de sehol sem jelent meg ezzel kapcsolatos érdemleges információ, pedig egyesek még Münchenben is keresték a magyar fejlesztést. Érdekes módon mi rátaláltunk a nem is olyan távoli Zuglóban.

A rendelkezésünkre bocsátott Spectrum emulátor-t egy héten keresztül nyúztuk, az alábbiakban szeretnénk közre adni a szerzett tapasztalatokat. A Spectrum emulátor-t

úgy tervezték, hogy megjelenésében ne térjen el az angol tervezésű import perifériáktól, ezért ugyanúgy néz ki, mint az Enterprise EXDOS lemezvezérlő kártya. A mintegy 1000 betöltött programból 843 futott tökéletesen, a fent maradó programok három részre oszthatóak:

- azok a programok, amelyek már a loader-nél leálltak. (Spectrum emulátor nem tud turbósított file-okat tölteteti, illetve menteni..) Ezeket a programokat normál sebesség-re való konverzió után már be tudja tölteni.

- azok a programok, amelyek működnek, de lelassulva futnak és időnként elszállnak. Ebben az esetben a problémák abból adódnak, hogy ezek a programok nem a standard ROM belépési pontokat használják.

- azok a programok, melyek futásképtelenek, pl. a betöltést követően lefagynak. Ennek - az esetek többségében - az az oka, hogy a Spectrum ROM - jóban van egy terület FFh - val feltöltve, és vannak olyan programok, amelyek ide helyezik a megszakítás-vektorukat. A Spectrum emulátorban ezen a területen aktív kód található. Ezek a programok is futtathatóvá tehetők, némi gépi kódú beavatkozás által. (Természetesen csak saját használatra..)

Ezekon kívül van a Spectrum emulátornak még egy „óriási” hiányossága, nem ismeri a FLASH-t. Eddig leírtuk azt, hogy a Spectrum emulátor mit nem tud, de az igazsághoz hozzá tartozik az is, hogy mennyivel tud többet a Spectrumnál. Nos, lehetőség van a DAVE hangchip, a CENTRONICS printer-port, a soros port, a két joystick port, valamint az RGB video kimenet használatára.

Mi az igazán jó hírt a végére hagytuk! Annak ellenére, hogy a Spectrum emulátort eddig négyszer tervezték át az éppen kapható elektronikai alkatrészek illetve a gyártók igényeinek megfelelően, a tavaszi BNV-n már kapható lesz, es remélhetőleg hozzá járul ahhoz, hogy ugrásszerűen megnövekedjen a Spectrum felhasználók népes tábora.

*A Spectrum örök és elpusztíthatatlan!
Spectrum Világ 8 / Enter Face 1*

Ismét az EMULÁTOR-ról

A Sinclair ZX Spectrum számítógépről mindenkinek megvan a saját pozitív tapasztalata, ezért ezt nem kell különösebben részleteznünk. Az EMULÁTOR egy olyan eszköz, amely egy másik berendezés működését képes utánozni, annak ellenére, hogy felépítése teljesen más.

Az ENTERPRISE gépre készült Spectrum EMULÁTOR majdnem „SPECTRUMMÁ” változtatja a gépet.

Ezúttal az EMULÁTOR folyamatos használata közben szerzett tapasztalatainkat szeretnénk megosztani Olvasóinkkal, hiszen az okos más kárán (is) tanul.

Az első problémák a csatlakoztatásnál jelentkeznek. A SYSTEM BUS BRIDGE-dzsel viszonylag könnyen tudjuk „összedugni”, mert nemigen tud elcsúszna a kártyáról, viszont könnyen elfordulhat, hogy a BRIDGE és a gép közötti csatlakozással megszenvedünk.

Biztos módszer nincs, de idővel majd belejövünk! Nagyon vigyázzunk, MINDIG FESZÜLTSÉGMENTESÍTÜK A GÉPET, mielőtt „dugunk” vagy „lehúzzunk”, mert ha ezt elmulasztjuk, mindent tönkretelhetünk!!!

Ha esetleg nem sikerül a pontos csatlakoztatás, legfeljebb nem jelentkezik be az EMULÁTOR, vagy „se kép se hang” játékot űz velünk a gép, de hibát nem okozunk.

Az EMULÁTOR BASIC-ben teljesen SPECTRUM kompatibilis, kivéve a FLASH utasítást, de ha egy programban ilyet talál, nem zavarja meg, hanem átugorja. Ez a játékoknál nem túl érdekes, de az kifejezetten kellemetlen, ha a szerkesztő sorban „elvész” a kurzor!! Ha legalább inverz volna! Az „F” billentyűkre „kirakott” HELP funkció a kezdő EMULÁTOR használóknak jelent nagy segítséget, hiszen nem kell mindig a leírás után kaparászni.



Gépi kódban a programok kb. 85-90 százaléka fut. A „kiakadás” oka lehet pl. „illegális” belépési cím használata, saját loader, vagy néha „spéci” hangeffektek. Ha az EMULÁTOR-nak nem tetszik valami, akkor vagy egy karakter méretű kis fehér négyzetet tesz ki a bal vagy a jobb sarokba, vagy egyszerűen lefagy. Néhány program erről vagy nem tud, vagy dafke csak azért is fut, miközben vígan virít a négyzet valamelyik felső sarokban. Találkoztunk több olyan programmal is, amelyek nem jeleztek hibát, de a játék közben a harmadik-negyedik pálya után lefagytak.

Az ilyen Programok elsősorban a többrészes, ún. „utántöltős” játékok, de gyakran előfordul ilyen „lemerevedés” az egyrésztes programok esetében is. Sajnálatosan a régi (1983-84 kiadású) programok, amelyek jórészt BASIC-ben íródtak, esetenként egyáltalán nem működnek az EMULÁTOR KÁRTYÁVAL.

Vannak olyan programok, melyeket ha MULTIFACE-szel előzetesen SPECTRUMON „átrántjuk”, felélednek halottaikból, és kifogástalanul működnek, mert a MULTIFACE-szel történő lementéskor a „krakkolás” nyomai eltűnnek.

A SPECTRUM programok sajátossága, hogy a „jugoszláv illetőségű crackerek”, a MULTIFACE tulajdonosok, valamint a rengeteg másolás és magánerőből történő feltérés hatására egy programból igen sokféle változat létezik.

Könnyen előfordulhat, hogy sikerül egy működő verziót találnunk valamely használhatatlannak vélt programból.

Találkoztunk egy érdekes jelenséggel is, melynek nem értjük az okát. Gyakran előfordul, hogy az alsó sor a „C” billentyűtől kezdődően „elhal”. A dolog érdekessége, hogy időnként a felettük lévő billentyűk pl. az az „M”-nél a „K”, az „N”-nél a „J”, a „SPACE” helyett az „ENTER”, stb.) kezdtek el jól, máskor pedig „tudathadásosan” működni. Eme jelenséget nem egy EMULÁTOR-nál tapasztaltuk, tehát az kizárható, hogy a mi példányunkban, vagy a gépünkben volna a hiba! Megint érdekes, hogy ez csak a gépi-kódú programok futtatásakor jött elő!

Vonjuk le a tanulságot: ha valamelyik billentyűnk nem akar működni, akkor próbálkozzunk a környező billentyűkkel, esetleg a SHIFT billentyűvel együtt nyomuk meg őket. Ha a makacs kis programunk még enne ellenére sem áll kötélnék, úgy ajánljuk a „KOMPLEXY SYSTEM” által bevezetett módszert, nevezetesen azt, hogy tenyérfelületünkkel óvatosan az elérhető billentyűkre nehezedünk, vagyis több billentyűt nyomunk le egyszerre. Sok programot sikertelt már ilyen módszerrel elindítani. Az ok az, hogy az indítószó valóban több billentyű együttes lenyomása szükséges! Az itt ismertetett trükkök ellenére mégis elég sok programtól esünk el, pl. FIGHTER PILOT, MERCENARY, stb.

A programoknak van egy része, melyek a használhatatlanságig lelassulnak, pl. „WHAM! THE MUSICBOX”. Némely software éppenséggel gyorsul, ami nem volna baj, ha csak a játék sebessége változna, ám nagyon zavaró lehet, ha az irányítás beállításánál nem csak az aktuális irány, hanem az összes opció egy billentyűre definiálódik a felgyorsult billentyűzet lekérdezés miatt. Ha, már végképp nem tudunk olyan rövid ideig lenyomni (pöccintetni) egy billentyűt, hogy csak egy helyre történjen a definiálás akkor - ha erre lehetősé van a menüben - válasszuk ki valamelyik, billentyűzettel párhuzamos botkormányillesztőt.

Ilyenek lehetnek:

CURSOR	5, 6, 7, 8, a tűz gomb sokféle lehet, pl. SHIFT, SPACE, 0, 9, M, stb
AGF / PROTEK	Az előzőhöz hasonló, de az irányok más kombinációban lettek összeállítva.
SINCLAIR 1	1, 2, 3, 4, tűz: 5
SINCLAIR 2	6, 7, 8, 9, tűz: 0
INTERFACE II.	ld. SINCLAIR 2

Az egyéb Interface-ek (pl. KEMPSTON) nem billentyűzet-párhuzamosak, ezért ne válasszuk ezeket, mert a kiválasztást követően a klaviatúra nem él, és nincs lehetőségünk módosítani.

Sajnos az EMULÁTOR használatával elesünk mind a belső, mind a külső botkormányok, valamint a tényleg jól használható EXDOS-kártya használatától is. Értesítéseink szerint kifejlesztés alatt áll egy KEMPSTON típusú botkormány illesztő, az EMULÁTOR-t felhasználók számára, de amíg megjelenik - ha nem kósza a híresztelés -, addig is a nem túl strapabíró billentet kell „gyötörnünk”. Nagy kár, hogy az EXDOS-t nem használhatjuk, pedig milyen klassz, is volna, ha a SPECTRUM programokat lemezzről is tölthetnénk!!!

A SOUND utasítás rejtelsei I.

pitch, duration, source, sync – kicsit bővebben



Írta: Bodnár Tamás
(Szipucusu)

Milyen számokat lehet beírni a pitch, duration és a többi paraméter után, és mit jelentenek ezek a számok? Megpróbáljuk a teljesség igényével átnézni ezt a témát. Köszönöm Varga Istvánnak (IstvanV), hogy segítette a témát kivesézni, és kiegészítéseivel, pontosításaival valóban a dolgok mélyére áshattunk.

A SOUND szó hallatán az átlagembernek a fesztivál jut az eszébe. Viszont egy Enterprise-osnak teljesen mást jelent. Maga az angol SOUND (magyarul: hang) szó a latin sonus (szintén: hang) szóra vezethető vissza. Nem csak az angol nyelv, hanem a magyar is bővülködik latin eredetű szavakban, így ugyanebből a szóból származik pl. a szonáta, a szuperszónikus vagy a szonár szavunk is.

SOUND PITCH x

Az angol pitch szó több, egymástól eltérő dolgot is jelent, így pl. a sportpályán kívül a hangmagasságot is. Ha a hangmagasságot meg szeretnénk változtatni, a PITCH paramétert használhatjuk. A legmélyebb hanghoz a 0-át adhatjuk meg: SOUND PITCH 0 (ez egy H hang), és már morajlik is valami nagyon mély hang. A SOUND PITCH 127 a legmagasabb hangot adja (ez elméletileg egy fiz hang lenne), illetve adná, de annyira magas, hogy nem is halljuk már.

A SOUND PITCH 1 a legmélyebb C hangot jelöli. Ha egygyel megnöveljük a szám értékét, egy félhanggal magasabb hangot kapunk. Így egy oktávval magasabb hanghoz 12-vel kell a pitch értékét növelni, a következő oktáv C hangja tehát $1+12=13$. A következő oktávbeli C hang $13+12=25$, az azután 37, az azután 49 és így tovább. A SOUND PITCH 37 a normál C hang, a 38 a cisz, a 39 D, a 40 disz, a 41 E, a 42 F, és így tovább. (Zeneszerek tudják, hogy az E és az F hang között csak félhangnyi, azaz kiszszekund az eltérés, ahogy a H és az azt követő C között is, ezért nincs a zongorán fekete billentyű az E-F és a H-C hangok között. Így a 41 és 42 ill. a 48 és 49 különbsége 1, míg pl. a 37 és 39 különbsége 2.) Kb. 90-nél nagyobb értékeket nem érdemes megadni, mert olyan magas a frekvenciájuk, hogy nem hallatszanak jól. Másrészt, a hangmagasság ilyen magas hangoknál már nem olyan pontos, aminek okára rövidesen rátérünk.

D ^b	E ^b	G ^b	A ^b	H ^b			
C [#]	D [#]	F [#]	G [#]	A [#]			
C	D	E	F	G	A	H	C

1	2	3	4	5	6	7	8	9	10	11	12	13
13	14	15	16	17	18	19	20	21	22	23	24	25
25	26	27	28	29	30	31	32	33	34	35	36	37
37	38	39	40	41	42	43	44	45	46	47	48	49
49	50	51	52	53	54	55	56	57	58	59	60	61
61	62	63	64	65	66	67	68	69	70	71	72	73
73	74	75	76	77	78	79	80	81	82	83	84	85
85	86	87	88	89	90	91	92	93	94	95	96	97
97	98	99	100	101	102	103	104	105	106	107	108	109
109	110	111	112	113	114	115	116	117	118	119	120	121
121	122	123	124	125	126	127						



Milyen zenei hangoknak felelnek meg az egyes SOUND PITCH értékek?

Nem csak egész számokat adhatunk meg hangmagasságnak, így a SOUND PITCH 41.5 egy negyedhang az E és az F között. Megadhatunk akár 41.1-et, 41.2-t, tehát egész finom hangmagasság különbségek lehetségesek. A legmélyebb oktávban akár 0.03125 PITCH értékbeli különbség már eltérést okoz a frekvenciában (más kérdés, hogy az emberi fül ebből mit hall), a magasabb hangoknál kevésbé finoman állítható a hangmagasság. Így pl. a normál A hang (PITCH 46) környékén 0.05 körüli pitch értékbeli különbség jelent eltérést a frekvenciában. A két oktávval feljebb lévő A hang (PITCH 70) környékén „csak” 0.2-vel növelve lesz magasabb a hang. PITCH 100 felett pedig már a tört hangmagasság-értékeknek nincs is értelme: csak akkor változik a hangmagasságuk, ha 2-vel, 3-mal növeljük az értéküket, így a tényleges zenei hangok frekvenciája sem érhető el. (Ezért is csak elméletileg fiz hang a legmagasabb, a 127-es.) De ilyen magas hangokat úgyse használunk, mert nem is hallatszanak már. (Ennek az egésznek az oka, hogy a DAVE 125000 / N Hz frekvenciájú négyszögjelet tud előállítani, ahol N 2 és 4096 közötti egész lehet. A PITCH pedig elvileg $12 * \log_2(125000 / N / 440) + 46$, de a SOUND: eszköz nem egészen pontos, lineáris interpolációt használ.)

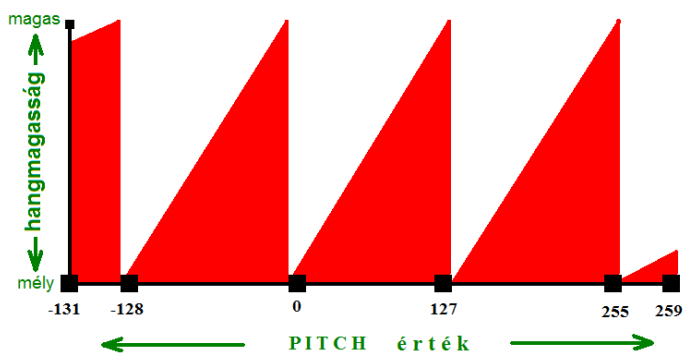
Itt kell megemlíteni, hogy a legtöbb PC-s zeneszerkesztő program, így a midi szerkesztők és az FL Studio sem teszi lehetővé (egykönnyen) a kromatikus skálánál finomabb skála használatát (mikrotonális rendszerek), de az EP basicjébe szinte minden eszköznél könnyebben írhatunk be ilyen zenét! Igaz, az ilyen zene nem túl gyakori, inkább a keleti dallamvilágra jellemző.

Megadhatunk PITCH értékek nullánál kisebb értéket is, egészen -131-ig. A -1-től kezdődően azonban nem mélyebb hang lesz a 0-nál, hanem újra a legmagasabb (127-tel jelzett) hang jön. 127-nél is megadhatunk nagyobb értéket, egészen 259-ig. A 128 szintén nem magasabb a 127-nél, hanem a PITCH 0 értékkel azonos. Ha -131-nél kisebb, vagy 259-nél nagyobb értéket adunk a PITCH után, hibaüzenetet kapunk.

Ezért ha egy 0 és 127 közé eső pitch értékhez 128-at hozzáadunk vagy elveszünk belőle, ugyanazt a hangmagasságot kapjuk, pl. a SOUND PITCH 41 ugyanúgy szól, mint a SOUND PITCH 41+128 vagy a SOUND PITCH 41-128. A basic olyan jó hozzánk, hogy a legmélyebb 4 pitch érték (SOUND PITCH 0, SOUND PITCH 1, SOUND PITCH 2, SOUND PITCH 3) úgy is elérhető, ha 128-at, és úgy is, ha $2 \cdot 128$ -at adunk hozzá, pl.

SOUND PITCH 2 = SOUND PITCH 2+128 = SOUND PITCH 2+128+128.

Tehát a 256, 257, 258, 259 „kóbor” pitch értékek még hozzácsapódtak a mezőny végéhez, ki tudja, miért. A mezőny elejére pedig három hallhatatlan magas hang sorolt be. A -128 tehát a 0-val azonos magasságú, a legmélyebb hang. Ez alá még bevonult a -129, -130, -131, melyek magassága a 127, 126, 125 pitch értékkel azonos. (Mint tudjuk, nullától lefele a számok „visszafele mennek”. Vagyis, ha az abszolút értéküket nézzük, minél nagyobb egy szám abszolút értéke 0 alatt, annál mélyebb a hang, míg a pozitív számok jól megszokott tartományában minél nagyobb egy szám, annál magasabb hangot jelöl.)



SOUND DURATION n

Az angol duration szó latin eredetű (latinul: duratio), a durus (kemény, szilárd, tartós) szóra nyúlik vissza, időtartamot jelent. Az eredeti latin szó tovább él más szavakban is (pl. olasz: durare, német dauern: tart valameddig – utóbbiból származik a magyar dauer szó, mely az időbeli tartósságra utal).

Mint tudjuk, a hang hosszát a DURATION paraméterrel adhatjuk meg. Az n értéke a másodperc 50-ed részét jelenti, tehát a legrövidebb (1-es hosszúságú) hang 1/50 másodpercig szól. Egy másodperces hanghoz a SOUND DURATION 50-et kell megadni (ez az alapértelmezés is),

két másodpercig szóló hanghoz SOUND DURATION 100 kell. Tehát, ahány másodperc hosszú hangot akarunk, azt meg kell szorozni 50-nel.

Meg lehet adni tört értéket is a DURATION paraméterében, de a tizedespont utáni számoknak nincs hatása, hiszen az EXOS hang vezérlése 50Hz-en működik.

A leghosszabb hang, ami megadható, az a SOUND DURATION 65536, ez 1316.34 másodpercig szól (ami majdnem 22 perc), de még senkinek nem volt kedve kivárni és letesztelni. Ha ennél is nagyobb számot adunk meg, nem a hang lesz hosszabb, hanem újra előlről kezdődik minden: a 65537 a DURATION 1-nek felel meg, a 65538 a DURATION 2-nek, és így tovább, egészen 131072-ig, ami újra 65536-nak felel meg, majd a 131073 újra az 1-es duration értéket jelöli. (A gép tehát elfogad 16 biten nem ábrázolható értékeket is, de 16 bitesre csonkítja.) Így a következő utasításokban ugyanannyi lesz a hang hossza:

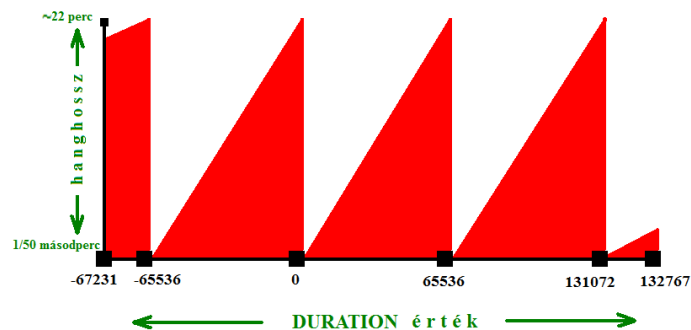
SOUND DURATION 1

SOUND DURATION 1+65536

SOUND DURATION 1+65536+65536 (azaz: 131073)

A legnagyobb megadható szám 132767. Ez $132767 - 2 \cdot 65536 = 1695$ duration értékkel azonos, ami kb. 34 másodperc.

A SOUND DURATION 0 nem 0 hosszúságú hangot eredményez, hanem a leghosszabbat, 65536 hosszúságú. Ez azért van, mert ha DURATION 1-től lefelé megyünk a negatív számok birodalma felé, újra a leghosszabb felől haladunk a rövidebb hangok felé. Így a DURATION -1 a DURATION 65535-öt, a DURATION -2 a DURATION 65534-et jelenti, és így tovább. A -65535 ugyanaz a hanghossz, mint az 1. A legkisebb megadható a SOUND DURATION -67231, ami kb. 34 másodperccel rövidebb a lehetséges leghosszabb hangnál. Így ha valamit elszámolnánk basic programunkban, és megszólal egy hang, mely nem akar elhallgatni, akkor valószínűleg 0-át vagy negatív értéket sikerült beállítani a hang hosszának. Ilyen tragédia esetén a CLEAR SOUND utasítás segít, vagy a reset gomb.



Házi feladat: a gyári Enterprise induláskor ezt írja ki: „131072 bytes in system”. Ez a szám milyen hanghosszt jelöl?

SOUND SOURCE n

A source szó jelentése eredet, forrás. Mindenki tudja, hogy az Enterprise hangchipje három négyszögjelcsatornát (tisztá hangot) és egy zajcsatornát képes

kezelni. Ez azt jelenti, hogy egyszerre négy különböző hang szólaltatható meg, ebből az egyik zaj, a másik három zenei hang is lehet. Így az egyik négyszögcsatornán szólhat egy szólam, a másik négyszögcsatornán egy kíséret, a harmadikon például basszus, a zajcsatornán pedig például dobokra emlékeztető hangzások. A három négyszögcsatorna azonosítója: 0, 1, 2, a zajcsatornái pedig 3. A SOURCE paramétereként ezt az azonosítót kell megadni. Alapértelmezésként a SOUND SOURCE 0 szól.

Ha egymás után meg akarunk szólaltatni pl. egy C és egy E hangot, az a következő basic programmal lehetséges:

```
100 SOUND PITCH 37
110 SOUND PITCH 41
```

Ha nem egymás után, hanem egyszerre akarjuk megszólaltatni a C és E hangot, akkor az egyik hangot egy másik csatornára kell tenni, például így:

```
100 SOUND PITCH 37
110 SOUND PITCH 41,SOURCE 1
```

Ha a zajcsatornát is használni akarjuk a C és az E hang kiadása közben, ezt kell megadnunk:

```
120 SOUND SOURCE 3
```

A SOURCE 3 paraméter használata sistergő hangot eredményez, ami leginkább a tévében az adáskimaradásra emlékeztet. Fontos még megemlíteni, hogy a zajcsatorna hangmagasságát nem tudjuk állítani a PITCH paraméterrel (tehát a SOUND SOURCE 3,PITCH x nem változtatja meg a hangmagasságot, akármennyit is írunk az x helyére). Van azonban egy másik lehetőség a zajcsatorna magasságának megváltoztatására, erről a STYLE paraméternél lesz majd szó. Egyébként ez a sistergés, amit a zajcsatorna kiad, fixen 31.25 kHz magasságú. (Azért pont ennyi, mert a Dave hanggenerátorai 250 kHz-es órajellel működnek, a négyszögjelek frekvenciája pedig csak olyan érték lehet, ami ezt páros egészszel osztja. $250/8$ pont 31.25.)

Érdekes, hogy a SOURCE paraméter után megadhatunk 0 és 3 közöttitől eltérő számot is. A legnagyobb megadható szám a SOUND SOURCE 255, a legkisebb a SOUND SOURCE -256. Sajnos ezek a számok nem jelölnek eddig felfedezetlen csatornákat, hanem a három csatorna ismétlődik. Így pl. ha 4-et adunk meg, az újra a 0-s csatorna lesz, az 5 az 1-es csatorna, és így tovább. Mindegyik csatornára hivatkozhatunk úgy is, ha a SOUND SOURCE értékéhez hozzáadunk 4-et, 8-at, 12-t, stb., azaz négy egész számú többszörösét. Így pl. a zajcsatorna megszólaltatható a SOUND SOURCE 3, a SOUND SOURCE 7 és a SOUND SOURCE 11 kiadásával is, és így tovább, végül a SOUND SOURCE 255 is a zajcsatornát használja. Gyakorlatilag a megszólaltatandó csatornához a következő képlet is használható:

SOUND SOURCE $n+(4*m)$

ahol n a csatorna tényleges száma, m pedig tetszőleges egész szám, melynek 64-nél kisebbnek kell lennie, de nem lehet kisebb -64-nél.

Érdekes, hogy mivel a 0-ás csatornára a 4-es számmal is hivatkozhatunk, lehetne a csatornák számozása 1-től 4-ig, ahol a 3-as a zajcsatorna, a 4-es pedig tiszta hang. Így pont akkora számokkal hivatkozhatunk a csatornákra, ahány csatorna ténylegesen van. Technikai okok miatt célszerűbb 0 és 3 közötti számokkal hivatkozni a csatornákra, de basic programozásához az 1-4 számozási rendszer is megfelelne.

Meg lehet adni a SOUND SOURCE paraméterének törtet is, de a tizedespont utáni részeket a gép nem veszi figyelembe természetesen.

Ha azt akarjuk, hogy valamelyik csatornán a hang hallgasson el, a CLEAR QUEUE n parancsot használhatjuk, ahol n az elnémitani kívánt csatorna száma.

SOUND SYNC 1 / 2 / 3

A sync a synchronize (szinkronizál) szó rövidítése. Mindenki hallott már a basic lassúságáról. Ha több csatornán szólaltatunk meg zenét, előfordulhat, hogy mire egyik SOUND utasításról a másikra lép a program, eltelik egy picike idő ahhoz, hogy a két csatornán a hang ne egyszerre szóljon meg, hanem akár icipici csúszással. Főleg akkor, ha a két SOUND utasítás között valami műveletet is végez a gép, pl.

```
90 LET A=41:LET B=127
100 SOUND PITCH 41
110 LET A=A+12:LET B=255-B
120 SOUND PITCH A,LEFT B,RIGHT 200,SOURCE 1
```

A 0-ás csatornán a hang előbb fog megszólalni, mint az 1-es csatornán. Hogy egyszerre szóljon meg a két csatorna, a SYNC paramétert kell használni az egyszerre megszólaltatandó SOUND utasításokban. A fenti program így kiegészítve:

```
100 SOUND PITCH 41,SYNC 1
110 LET A=A+12:LET B=255-B
120 SOUND PITCH A,LEFT B,RIGHT 200,SOURCE 1,
SYNC 1
```

Így a 100-as sor hangja nem fog megszólalni egész addig, amíg a 120-as sorhoz nem ér a program. Ha hozzáadjuk a programhoz pl. a 115 WAIT DELAY 8 sort, akkor is megvárja a 120-as sort, és csak akkor szól meg hang. Egy adott SOUND utasításban a SYNC paraméter utáni szám azt jelöli, hogy hány másik hanggal kell egyszerre megszólaltatnunk az adott hangot. Két hang egyszerre megszólaltatva:

```
SOUND PITCH 49,SYNC 1
SOUND PITCH 44,SOURCE 1,SYNC 1
```

Három hang egyszerre megszólaltatva:

```
SOUND PITCH 49,SYNC 2
```

SOUND PITCH 44,SOURCE 1,SYNC 2
SOUND PITCH 41,SOURCE 2,SYNC 2

Négy hang egyszerre megszólaltatva:

SOUND PITCH 49,SYNC 3
SOUND PITCH 44,SOURCE 1,SYNC 3
SOUND PITCH 41,SOURCE 2,SYNC 3
SOUND SOURCE 3,SYNC 3

Vagy, úgy is megközelíthetjük, hogy ahány hangot összehangolunk, annyi mínusz 1-et kell a SYNC-ben megadnunk. Így 3-nál nagyobb számot nem is érdemes megadni a SYNC után, hiszen egy adott csatornával egyi-

dejüleg maximum 3 másik csatorna szólalhat meg, mivel ennél több csatorna nincs. Mint látható, az összes összehangolt csatormánál ugyanazt a számot kell megadni SYNC után.

1-nél kisebb számot nem érdemes megadni, hiszen ki akarna nulla másik csatornával szinkronizálni? Ettől még megadhatunk nullát (ez az alapértelmezett), ilyenkor nincs szinkronizálás. 3-nál nagyobb számot is megadhatunk, a 4 újra a 0-ás értéket jelöli, az 5 az 1-et, a 6 a 2-öt, a 7 a 3-at, a 8 megint a 0-t, és így tovább. A legnagyobb megadható szám 255. Negatív számokat is megadhatunk egészen -256-ig, és törteket is. Ennek értelmét chilei kutatók még keresik.

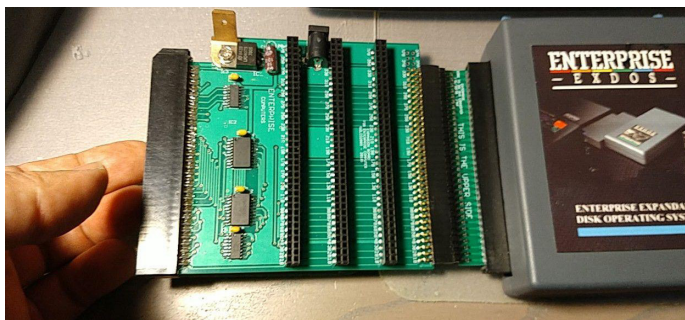
Új buszbővítő és M-Slot (MSX cartridge adapter) by Gflorez

Gflorez műhelyében készültek az alábbi hardver kiegészítők Enterprise-hoz. Új, modern IC-kkel és IDC csatlakozókkal felszerelt **buszbővítő kártya**. Ezzel egy nagy probléma oldódott meg, hiszen élcsatlakozókat már nem nagyon gyártanak.

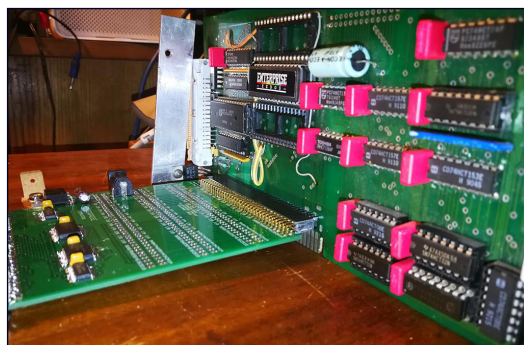
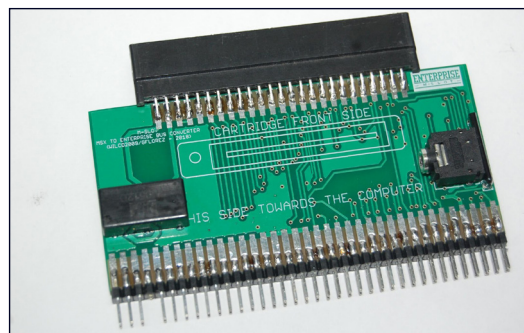
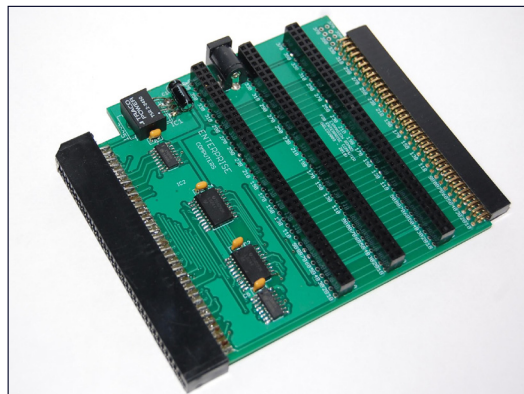
M-Slot kártya, mely lehetővé teszi az MSX cartridge-ek csatlakoztatását.

Ha valakinek kevés lenne a buszbővítőn a három IDC csatlakozó, könnyedén csatlakoztathat egy újabb buszbővítőt az elsőhöz.

Gondolva a régi hardver eszközökre, készült egy ún. **SIL to EDGE csatlakozó** is, mellyel könnyedén illeszthetünk régi kártyákat az új buszbővítőbe.



Például így csatlakozik egy EXDOS kártya ...



... így pedig egy MICRETEAM EXDOS kártya.

Microsoft Basic



Írta: **KISS LÁSZLÓ**
(Lacika)

Mostani cikkünk tárgya a Golden Baton ismertetőjének egyik félmondata, nevezetesen: „**A korabeli BASIC nyelvű játékok...**”. Először is azt érdemes körüljárni, mit értünk „korabelinek”? Az első otthoni felhasználásra szánt (megfizethető) számítógépek 1977-ben jelentek meg pár hónap eltéréssel: az Apple II (június), TRS-80 Model I (augusztus), és a Commodore PET (október). Ezek közül a TRS-80 számítógépek futották be a legnagyobb „karriert”, elsősorban 600 dolláros kezdőárának köszönhetően (szemben az Apple II 1298 dolláros árával). Az előbbi árak természetesen csak a 4 KB-os kiserelésre értendőek (monitorral), később a 16K-s változatok terjedtek el. A három gép paraméterei teljesen eltérőek voltak, eltérő processzorok dolgoztak bennük, egy valamiben azonban nagyon is hasonlítottak: mindháromban (a TRS-80 gépcsaládban 1978-tól) a Microsoft BASIC valamely változata működött, akárcsak később - igaz csonkítva(!) - a VIC20-ban és C64-ben is, illetve még később - bővítve - Plus-4-en. A TRS-80 Model II számítógépet közvetve mi magyarok is jól ismerhetjük: a magyar gyártású HT-1080Z iskolaszámítógép ennek - kicsit megkésett (1983-as (!)) - klónja.

Mindhárom gép erősen korlátozott grafikai képességekkel bírt, így pont alkalmasak voltak a korábban nagyszámú BASIC-játékok futtatására (ezekből már kaphattunk ízelítőt a 2017/2-3. számban). Emellett kézenfekvőnek tűnt - Az Adventure sikerén felbuzdulva - szöveges kalandjátékokkal enyhíteni a játszani vágyó amatőrök programéhségét. Amellett, hogy sok szöveges BASIC játék került kereskedelmi forgalomba (mai szemmel nézve megdöbbentő árakon) újságok, könyvek is közöltek - emberfeletti munkával - begépelhető játékokat. A leggrandiózusabb ilyen jellegű könyv talán az 1985-ben a Virgin Books kiadásában megjelent Castles & Kingdoms című könyv, amely 15 begépelhető szöveges kalandjátékot tartalmaz C64-re. Talán nem nehéz belátni, hogy a 15 játék megoldásánál lényegesen nagyobb kihívást jelenthetett azok (hibamentes) begépelése: a közismerten olvashatatlan Microsoft BASIC programlisták - a magyarázatokkal és „körítéssel” együtt 178 oldalt tölthettek meg. (Valószínűleg olcsóbb lett volna a programokat lemezen kiadni, és a vevő is jobban járt volna...). Emellett több könyv is megjelent kimondottan szöveges kalandjátékok írásának mikéntjéről. Ebből mi magyarok is kaphattunk ízelítőt - és többen kedvet -, amikor Frank DaCosta, 1982-es „Writing

BASIC Adventure program for the TRS-80” című könyve 1986-ban(!) magyar fordításban is megjelent „A kalandprogram írásának rejtelsei” címmel. Mint az az eredeti címből is látszik a könyv TRS-80-hoz készült, de tökéletesen használható volt HT-1080Z-hez is. Ennek első példaprogramja volt az idehaza nagy karriert befutott Kardhalak és Kincsek program, melynek több, a hazai piacon fellelhető gépre készült, többé - de inkább kevésbé működő maszek verziója (Ep-re már van működő változat!). A könyv magyarországi megjelenése egybeesik hazánkban a szöveges kalandjátékok írásának afféle kisebbfajta „népmozgalommá” válásának kezdeteivel, amiből sajnos a Spectrum kimaradt.

Nem úgy a második iskolaszámítógép, azaz a Plus-4! Tihor Miklós első kalandjátékai a Sárkányölő, a Gengszter még C64-re készültek, ez utóbbi a Commodore Egyesületi lap 1986/4. számában jelent meg, begépelhető formában. A Hős lovag lett az első kereskedelmi forgalomba került magyar nyelvű kalandjáték: C16-ra jelent meg a Novotrade kiadásában (250 Forintért). A játék egy 1980-ban megjelent, eredetileg TRS-80 játék, a Dragonquest alapján készülhetett. (A Castles & Kingdoms című könyvben jelent meg C64 változat belőle, megerősítendő, hogy a C64-es szöveges BASIC kalandjátékok jelentős része még a nagy elődökön készült.) Ezen korai kalandjátékok általános jellemzője - a szűkös (jellemzően 16K, de vannak alapkiépítésű VIC20-ra készült 3,5K-s kalandjátékok is...) memóriakapacitásra tekintettel, hogy nem túl bőbeszédűek, mondhatni „távirati stílusban” kommunikálnak a játékkal. Mindazonáltal lenyűgöző, milyen ötletes játékokat tudtak írni alig 16K-ban, egészen „elvetemült” programozási stílusban... Az eleve C64-re készült kalandjátékok már valamivel közlékenyebbek, pl. Olessák Róbert első - még BASIC-ben készült - kalandjátéka, a Brekk! - Avagy a békává varázsolt királylány.

Ezzel a bevezetővel el is értünk a mostani írás témájához: a Microsoft BASIC változatainak és az IS-BASIC összehasonlítása, hogyan is lehetne az említett játékokat futtatni Ep-n? Azt a felhasználói kézikönyvből tudjuk, hogy „Az Enterprise legtöbbször az ANSI X3J2/82-17 Draft Proposal for Standard BASIC-nak megfelelően jár el.” (akármit is jelentsen ez), a futtatni kívánt szöveges kalandjátékok lehetőségei pedig alig mutatnak túl a Minimal BASIC-en. Vegyük sorra az eltéréseket, mit kell módosítani, ha már pl. CBM prg Studio programmal szöveges állományba tudtuk menteni az eredeti programlistát!

Szintaxis

A Microsoft BASIC „leglátványosabb” tulajdonsága, hogy az egyes nyelvi elemek között álló szóköz elválasztó karakter(ek) elhagyható(k). A legtöbb programban el is hagyták a byte-spórolás jegyében (ez a program olvashatóságát persze nagymértékben rontja).

C64-en a legutolsó programsor sorszáma max. 63999 lehet, Ep-n 9999. C64-en RENUMBER nincs, a CBM prg Studio-ban viszont van.

Egy programsor IS-BASIC-ben 254 karakter hosszú lehet, C64-en 80 karakter. Ez több helyen a program egyszerűsítését teszi lehetővé. C64-en ha a THEN után több utasítás áll(na), ami már nem fér bele a 80 karakterbe, kényszerűségből egymás után ugyanazt a feltételt vizsgálva végzik el a műveleteket (vagy GOTO-t használnak), a 80 karakternél többet tároló szöveges változók is csak több sorban kaphatnak értéket (összefűzéssel), ezek mind összevonhatók, akárcsak a rövidebb DATA sorok.

A közhiedelemmel ellentétben Ep-n is egy programsorban több utasítást is írhatunk, kivéve a blokkszervező utasításokat (FOR - NEXT, DO - LOOP, DEF - END DEF, WHEN - END WHEN, HANDLER - END HANDLER), valamint a NUMERIC, STRING, CONTINUE, IMAGE, ON GOTO, ON GOSUB, REM, END utasításokat, melyek minden esetben külön sorban írandók! Figyelni kell még a GOSUB használatára: ez minden esetben a sor utolsó utasítása legyen, mert az ezután lévő utasításokat már nem hajtja végre az interpreter!

Némi hajhullásra adhat még okot a Microsoft BASIC-ben „közkedvelt” IF ... THEN NEXT: A=A+1 típusú szerkezet, ahol a feltétel teljesülése esetén a NEXT hajtódik végre, egyéb esetben a kettőspont után álló további utasítás(ok).

Változók

A Microsoft BASIC-ben a változók neveinek első karaktere csak betű lehet (A-Z), ezt tetszőleges számú alfa-numerikus karakter követheti (0-9, A-Z), de csak az első két karakter szignifikáns. A változónév nem tartalmazhat BASIC alapszót (pl. a VALI, WORD nem megengedett). A változó nevekkel tehát nem lesz sok gondunk IS-BASIC-ben bármi jó, ami Microsoft BASIC-ben jó, mindössze az egész típusú változókat jelölő % jelet kell a végéről kivenni.

Nagyobb gond, hogy míg Microsoft BASIC-ben megengedett, hogy egy numerikus változó és egy tömbváltozó ugyanazt a nevet viselje, az IS-BASIC-ben nem. Márpedig úgy tűnik a korabeli programozók ebből sportot csináltak, rendszeresen találkozhatunk READ O(O) típusú - amúgy nagyon csúnya - értékadásokkal. Itt értelem-szerűen az egyik változót át kell nevezni.

Anno, akik más gépeken kezdték az ismerkedést a BASIC nyelvvel, rendszeresen szidták az IS-BASIC-et, amiért a változókat - nagyon helyesen - első használat előtt deklarálni kell. Microsoft BASIC-ben nem kell, egy újonnan bevezetett változó automatikusan 0 értéket vagy üres stringet vesz fel. Sőt, a maximum 10 elemű tömbök deklarálása is elhagyható! Ezt ki is használták a programkészítők, a 0 kezdőértékkel rendelkező változók - a memóriaspórolás okán - következetesen nincsenek deklarálva. Talán meglepő, de a Microsoft BASIC ezen „szolgáltatása” felderíthetetlen és így igen gyakori hibát okoz a játékokban. A C64-es kalandjátékok legalább fele(!) tartalmaz a játékmenetet befolyásoló hibát. Ha a program egy esemény megtörténtét pl. O(30)=1000 feltétellel vizsgálná,

de helyette O(30)=1000 került a programba, az adott esemény sosem fog bekövetkezni (mert az 1000 változó értéke mindig 0 lesz). Ez néhány esetben nem a megoldás egyszerűsödését okozza, hanem épp ellenkezőleg, a játék teljesíthetetlen lesz. Hogy az ilyen hiba mennyire felderíthetetlen, jól példázza, hogy 'A Hős lovag' úgy került kereskedelmi forgalomba, hogy megoldhatatlan... Az ilyen hibákat az IS-BASIC szerencsére jelzi.

Kevesen tudják talán, hogy IS-BASIC-ben az értékadások összevonhatók, egyetlen LET utasítással egyszerre több változónak is adhatunk értéket: LET, A,B,C,D=0.

Stringek

Microsoft BASIC-ben a stringek végét jelző idézőjel elhagyható, IS-BASIC-ben nem.

Microsoft BASIC-ben egy szöveges változó maximum 255 karaktert tárolhat. Ep-n 254-et, ha azt a STRING utasítással deklaráljuk, egyébként max. 132 karaktert. Ez nem szokott gondot jelenteni, mert ilyen hosszú sort Microsoft BASIC-ben nem lehet beírni. Szöveges tömbváltozónál gond lehet viszont a mérettel, mert a Microsoft BASIC másként tárolja a szövegfűzér típusú változókat: a változóterületen csak a string hosszát és egy mutatót tárol. A szövegfűzért a program után a string-területen (a memória végétől lefele terjeszkedve) tárolja úgy, hogy a szövegfűzér-változó helyfoglalása ténylegesen a tárolt karakterek számától függ. Ez a módszer egyrészt nagyon gazdaságosan bánik a memóriával, ha Ep-n ugyanúgy DIM utasítással deklarálnánk egy nagyobb méretű tömböt (mert sok helyszín van a játékban), betelt a memória hiba-üzenet kapunk. A STRING utasítást kell használnunk, megadva a maximális hosszt. Másrészt - könnyű belátni - hogy a Microsoft-féle megoldás elképesztően lassítja a szövegfűzér műveleteket: egy 168 szót tároló tömbben a lineáris keresés 14 másodperc C64-en, Ep-n 2 másodperc.

Logikai kifejezések

Az AND művelet természetesen számokra is értelmezhető, de ilyen esetben a művelet mindig bitenként hajtódik végre. Vagyis a

```
IF (A AND 47)<>N THEN ...
```

kifejezés javítandó:

```
IF (A BAND 47)<>N THEN ...
```

A szövegfűzéseken értelmezhető műveletek

C64-en a konkatenáció (összekapcsolás) jele a „+”, Ep-n „&”.

Az általunk ORD néven ismert függvényt (a megadott string-kifejezés első karakterének ASCII kódját adja) a Microsoft BASIC-ben ASC-nek hívják.

A részstring-képzés Ep-n túl logikus és egyszerű, a Microsoft ehelyett rögtön 3 függvényt is bevezetett ugyanerre:

LEFT\$(<string>,<aritmetikai kif.>)

A string-ből az első <kif.> számú karaktert adja.

RIGHT\$(<string>,<aritmetikai kif.>)

A string végéről <kif.> számú karaktert ad.

MID\$(<string>,<kif1>, <kif2>)

A string <kif1> karakterétől <kif2> számú karaktert ad. <kif2> elhagyható, ebben az esetben a string végéig másolja a karaktereket. Úgy tűnik elég nagy zavar lehetett a fejekben e három függvénnyel kapcsolatban, senki ne csodálkozzon, ha pl. MID\$(X\$,LEN(X\$)-N+1) típusú produkciót lát, ez magyarra lefordítva RIGHT(X\$,N).

DATA sorok

Microsoft BASIC-ben a DATA sorokban álló szövegfüzeteket csak akkor kell idézőjelbe tenni, ha vesszőt tartalmaznak. Ezzel szemben IS-BASIC-ben akkor is, ha felkiáltójelet is tartalmaz (mert a '!' a megjegyzés jele is). Microsoft BASIC-ben bájt spórolás célzattal a DATA sorok adatlistáiban a 0 értékeket nem kell kiírni. Vagyis a

DATA 0,1,2,0,0,0,3,4,5,0

és a

DATA ,1,2,,,3,4,5,

ugyanazt az adatlistát jelenti. IS-BASIC-ben az adatlista utolsó elemét viszont mindenképpen ki kell írni, vagyis:

DATA ,1,2,,,3,4,5,0

A C64 BASIC-ben is van POS függvény, de teljesen mást jelent, mint az IS-BASIC-ben: kiszámítja a kurzor helyzetét az aktuális képernyő sorban. A lehetséges értéke 0-255 intervallumba eshet. Ez nem a képernyő 40 karakteres sorában lévő pozíció, hanem annak mértéke, amennyivel a kurzort a sor elejétől elmozgattuk (pl. egymás utáni sortörés nélküli PRINT utasításokkal). A függvény argumentuma lényegtelen. Ez az egyik leghasznavehetetlenebb függvény, bár szöveges kalandjátékokban szokták használni egyfajta nyakatekert szóátvitel megvalósítására. Az EXOS-ban ez alap, így ezen programrészletek egy az egyben törölhetők.

File-műveletek

File műveletek végzésére az OPEN, CLOSE, INPUT#, GET# és a PRINT# utasítások szolgálnak. IS-BASIC-ben ugyanazon elvek mentén kezelhetjük értelemszerűen a file-okat, apróbb szintaktikai eltérésekkel.

A Plus4-es játékok módosítása sem bonyolultabb (grafikát tartalmazó játékokkal most nem foglalkozunk), egyetlen említésre méltó eltérés az IF-THEN szerkezetekben használható ELSE-ág. Ez annyira megtetszett az amatőr programozóknak, hogy több programban egyágyú szelekciókat is „sikerült” megoldani IF-THEN-ELSE használatával.

SE-ONE kártya

(mp3 lejátszó + FM rádió Enterprise-hoz)

Az M-Slot kártya segítségével gond nélkül illeszthető és használató az MSX SE-ONE kártyája! SymbOS alatt mindkét funkció nagyszerűen működik!

Részletek az M-Slot kártyáról az Enterprise Forever fórumon! <http://www.enterpirseforever.com>

A SE-ONE kártyát itt szerezheted be:

<http://www.tmtlogic.com/>



Készül az EXDOS 3.0 (FAT 16)

Bruce Tanner technikai okok miatt kénytelen volt abbahagyni az EP-NET kártyát. Most szintén egy izgalmas és sokak által várt fejlesztésbe kezdett, a FAT 16-os EXDOS 3.0 írásába. Nagyon várjuk! :)

```

IS-BASIC          program  0
262144 bytes in system
226938 bytes unused

ok
:help
EPIDE version 3.0
EXDOS version 3.0
BASIC version 2.1
WP version 2.6 (SUPERWP)
WP version 2.1
ok
:dir l:
Volume in drive L: is FAT16
Directory of L:\

HELLO              7 09-08-18 12:23p
7 bytes in 1 file  1048384K free
ok

```

Arrow of Death Part 1

1983 - Digital Fantasia
kaland, szöveges

Brian Howarth második (és harmadik) játéka ugyanabban a világban, vagyis a Ferrenuil királyságban játszódik, mint a Golden Baton. A dolgok - ha lehet - még rosszabbul mennek mostanában a királyságban: Xerdon a gonosz varázsló rombolást és pusztítást hoz a birodalomra. Természetesen mi vagyunk az egyetlenek, akik megmenthetjük a királyságot a végpusztulástól, vagyis el kell pusztítanunk a varázslót, ehhez azonban egy különleges fegyverre van szükségünk: a Halál Nyilára.. A történet két részből áll: az első részben a mágikus fegyver „alkatrészeit” kell megszerelnünk. A második részben keresni kell egy nyilakészítő mestert, aki elkészíti nekünk a Halál Nyilát, majd el kell jutnunk Xerdon birodalmába, ahol meg kell ölnünk Xerdon-t. Bár a játék két része önmagában értelmetlen, mégis külön került kiadásra (egyenként 6,95 Font áron), a Mysterious Adventures sorozat második és harmadik részét képezve. A játék „motorja” természetesen változatlan maradt, a program kezelésre ugyanúgy történik mint a Golden Baton-ban. Egyetlen „downgrade”, hogy a HELP parancs nem hajlandó segíteni, kár is próbálkozni vele.

A program szókészlete:

AMULET	COOKHOUSE	LEDGE	REMOVE
ARMOUR	COURTYARD	LOOK	ROPE
ARROWHEAD	CUT	MARSH	ROPE
ATTACK	DEBRIS	MEDALLION	RUB
BATON	DRUG	MESSENGER	SCROLL
BEACH	DWARF	MOVE	SERPENT
BED	EXAMINE	NOTE	SHIFT
BEGGAR	FASTEN	ORB	SPECTACLES
BOOK	FEATHERS	PALACE	SWORD
BRANCH	FERRYMAN	PASSAGE	TAKE
BUILDING	FOREST	PILLOW	THROW
CAULDRON	GET	PLAQUE	TIE
CAVE	GIANTS	PLUCK	TOADSTOOLS
CHAIN	HOOK	POISON	TRAPDOOR
CHEST	HUT	POLISH	TREE
CLIFF	KILL	PURSE	TURN
CLIMB	KITCHEN	PUSH	WAIT
COAT	LADDER	PUT	WALL
COINS	LAKE	READ	ZARDRA

Kezdő kalandjátékosoknak a program kimondottan „el-lenjavalt”, jó néhány frusztráló jelenet miatt.

A játék elején a királyi palota udvarán állunk. A lábunk előtt heverő hírnök holtteste semmi jót nem ígér. Ha átkutatjuk a holttestet, egy amulettet találunk nála, amit tisztas hullarabló módjára magunkhoz is veszünk (LOOK MESSENGER, GET AMULET). Az udvaron is érdemes körülnézni, mert egy kötelet találunk (EXAMINE COURTYARD, GET ROPE). Északra a vár kijárata van, de előbb nézzünk körül a várban (W, S). A trónteremben (THRONE ROOM) is a pusztítás nyomai fogadnak. A földön Zardra, az



udvari varázsló fekszik, mellette a sorozat előző részében visszaszerzett arany varázspálcá hever. A varázspálcát most azonban fel sem bírjuk venni, sőt, ha megnézzük közelebbről, megöl halálos sugara. Foglalkozunk inkább az igencsak rossz bőrben lévő varázslóval (LOOK ZARDRA). Csak pár szót értünk motyogásából: „Mágikus Nyil... Pusztítsd el Xerdon-t...”. Ezután sajnos meghal. Keletre (E) a konyhát találjuk (LOOK KITCHEN), ahol egy kampót találunk (GET HOOK). Ezt rögtön a kötélre is kötjük (TIE ROPE, vagy FASTEN ROPE, GET HOOK). Keresgéljünk tovább a kastélyban (W, N, W). Megtaláltuk a király szobáját (KING'S CHAMBER). Elég jól be van rendezve: a szoba közepén, a falon a királyi címer, amelyre a program külön felhívja a figyelmünket. Ha megvizsgáljuk (EXAM COAT) megtudjuk, hogy forgatható. Tegyük is ezt: (TURN COAT), semmi hatása... Ne adjuk fel (TURN COAT, TURN COAT), így kinyitottunk egy titkos átjárót. Ne habozzunk bemenni (GO PASSAGE), a pánclétermen egy kardot találunk (GET SWORD). Visszatérve a király szobájába (W), vessünk közelebbi pillantást az ágyra is (LOOK BED). Teljesen váratlan módon egy párnát találunk benne... Ha megtapogatjuk (EXAMINE PILLOW), érzünk benne valamit. Ha felvágjuk a párnát a karddal (CUT PILLOW), egy erszény hullik ki belőle. Talán mindeki kitalálja, mi van benne (LOOK PURSE): egy kis dugipéncz (GET COINS)! Mielőtt távozunk, a pánclét is vegyük magunkhoz (GET ARMOUR, E, E). Visszajutottunk a várudvarra, most már kimehetünk (N).

A vár előtt egy öreg koldus álldogál, ha megnézzük közelebbről (LOOK BEGGAR), megtudjuk, hogy kinyújtja a kezét, vár valamit. Csak az érméket hajlandó elfogadni (GIVE COINS), cserébe viszont egy üveggömböt és noteszt hagy itt, majd eltűnik (GET ORB, READ NOTE). A noteszbe csak ennyi van írva: „ha minden veszve tűnik, várj!” A szélrózsa minden irányába mehetünk, induljunk mondjuk nyugatra (W, W). Egy kavicsos hegyi út végére értünk, előttünk sziklaszirt magasodik. Látszólag

zsákutcába jutottunk. Megpróbálhatunk ugyan felmászni (GO LEDGE), de valami túl nehéz. Persze, hogy a páncél az. A kötel segítségével egy trükkhöz folyamodunk: a só nehéz páncélba akasztjuk a kampót, felmászunk, majd a kötel végénél fogva felhúzzuk a páncélt is (HOOK ARMOUR, GET ROPE, GO LEDGE, PULL ROPE). Jobban belegondolva, mire valaki rájön erre magától... Felmásztunk ugyan a sziklaszirten, de csak egy eltorlaszolt barlang van előttünk. Innen látszólag csak vissza, lefelé lehet menni. Itt valószínűleg sok hajszálát kitepkedni az az elszánt játékos, aki maga próbálja megoldani a játékot... A játékosok hajzatát megkímélendő, a megoldás (RUB ORB, vagy POLISH ORB), és feltárul a barlang bejárata. Mindez úgy, hogy bármit is vizsgálunk, ami nálunk vagy a helyszínen van, mindenre csak azt írja ki a program, hogy semmi különös nem látunk. Pontosan ezért nem szeretik sokan a szöveges kalandjátékokat... Hát még akkor hogy fognak szitkozódni az „elszánt” játékosok, amikor beszélnek a barlangba és egy óriási kígyó simán megeszi őket! Tegyük meg tehát a szükséges „előkészületeket” (DROP ROPE, GET ARMOUR, WEAR ARMOUR, GO CAVE). Bent „persze” sötét van, lámpánk viszont nincs. Szerencsére az üvegömb - melyről a program annyit mond, hogy „semmi különös” - multifunkcionális szerszám, elég ismét megdörzsölni (RUB ORB) és már világít is. Elég jól fel vagyunk fegyverkezve, essünk neki a kígyónak (KILL SERPENT, vagy ATTACK SERPENT). Elég masszív darab ez a kígyó, nem lesz ez sima menet (KILL SERPENT, KILL SERPENT). Miután sikerült legyőzni a kígyót, felvehetjük a nyílhegyet (GET ARROWHEAD) - ez bizonyosan hasznos „alkatrész” egy nyílveoszóhoz (amíg a kígyó él, nem engedi felvenni). Hagyjuk itt a barlangot és a hegyet is (S, D, E, E, E). Egy sűrű erdőbe jutottunk. Bármerre mászkálunk, vagy a kastélyhoz, a hegyhez, vagy ide, az erdőbe jutunk. Jusson eszünkbe a notesz tanácsa, és tekintünk úgy, hogy „el vagyunk veszve” (WAIT). Újabb nonszensz jelenet következik: egy leszálló köd, egy sziklára repít minket... Csak északra (N) indulhatunk, ahol egy tópartra érünk. Előttünk egy bárka, mellette a révész. Úszni nem tudunk, a révész szolgáltatásait kell igénybe venni. Ha „megvizsgáljuk” a révést (EXAM FERRYMAN), megtudjuk, hogy ő is a markát tartja. Csak az amulettet fogadja el (GIVE AMULET). Beszállhatunk a bárkába, nincs más dolgunk, mint várni (WAIT). Mindenki azt várná - teljesen logikusan -, hogy a révész átvisz a túloldalra, de nem... Megint a köd jön, ami elrepít egy kiszáradt folyómederbe! Csak délre (S) indulhatunk, egy kipusztult erdő felé, itt forduljunk keletre (E). A tisztáson egy leláncolt rabszolgával találkozunk. A karddal elvághatjuk a láncait (CUT CHAIN), így kiszabadítjuk, a rabszolga hálásan követni fog minket (S, S, U). Újabb sziklaszirthez értünk, ahol egy hatalmas szikla eltorlaszolja egy barlangot. Próbáljuk arrébb görgetni (MOVE BOULDER, vagy SHIFT BOULDER). Egyedül nem bíránk megmozdítani a sziklát, de a rabszolga segít nekünk, így feltárul a barlang bejárata (GO CAVE). A barlangban egyetlen irattekercset találunk, ha elolvassuk (READ SCROLL), megtudjuk, hogy az óriás sas egy tollára is szükségünk lesz. Menjünk is innen (N, D, W), ezen a tisztáson furcsa mérgező gombákat látunk. Itt újabb szürreális jelenetnek lehetünk tanúi: az öreg koldus - ki tudja hogy került ide - megeteti a rabszolgát mérges gombával, aki mély állomba szenderül. A koldus megint eltűnik. (Ha a szikla elgurítása előtt

jövünk erre a rabszolgával, az pech!) De legalább megtudtuk, hogy a gombának erős altató hatása van (GET TOADSTOOLS, E, N). Egy konyhát találtunk (GO COOKHOUSE), ahol az óriások egy méretes üstben húslevest főznek (erről az EXAMINE CAULDRON paranccsal is meggyőződhetünk). Ízesítsük meg egy kicsit a levesüket mérgesgombával (DRUG CAULDRON vagy POISON CAULDRON). Most már bemerészkedhetünk az óriások házába (W, W, GO BUILDING), akik közben megették a levest és mélyen alszanak. Így már feljutunk az emeletre (GO LADDER vagy CLIMB LADDER). Az emeleten - kiábrándító módon - mindössze egy farönköt találunk (GET LOG), azonban az épület nyugati részében (W) egy farönk úsztató árok van felállítva. Ezen - a farönkbe kapaszkodva - leúszhatunk (GO FLUME). A farönkön kapaszkodva egy folyóba csúszunk le, melyen viszonylag „kényelmesen” utazhatunk sodrásirányba (WAIT) (más lehetőségünk amúgy sincs). Rövidesen a parthoz sodródunk (GO BEACH). Egy parti sziklafal aljába sodródtunk, másszunk fel (U). A szirt tetején - minő szerencse - éppen egy óriás sas nézelődik. A (LOOK EAGLE) parancs szerint szép tollazata van, ne is habozzunk tovább, rántsuk ki egy tollát (PLUCK FEATHERS). A sas felkap minket és egy zöld legelőre repít. (Barbárbak megpróbálhatják megölni a sast, de nem fog sikerülni, mert elrepül.) Bármerre indulhatunk, de csak északra tartva tudunk kijutni a végtelen zöld mezőről (N, N, N, N, N). Egy kis kunyhóhoz jutottunk, menjünk is be (GO HUT). A kunyhóban egy halott törpét találunk, hullarablásban már jók vagyunk (LOOK DWARF). Egy érmét találunk, viszont nem bírjuk el, valamit le kell rakni (REMOVE ARMOUR, DROP ARMOUR, GET MEDALLION). Keresgéljünk a kunyhóban is (LOOK HUT), találunk egy szemüveget, vegyük fel a jó szolgálatot tett üvegömb helyett (DROP ORB, GET SPECTACLES). Haladjunk tovább (S, N, W, N). Egy felégetett falu romjait találjuk, kotorásszunk kicsit a romok között (LOOK DEBRIS), találunk is egy csapóajtót (GO TRAPDOOR). A pincében egy táblát találunk a falon, ha elolvassuk (READ PLAQUE), megtudhatjuk, hogy ez egy nyílkészítő mester háza (volt). Érdemes lesz tehát kutakodni az itt lévő ládában (OPEN CHEST)! Egy könyvet találunk benne (DROP SWORD, GET BOOK). Ha megpróbáljuk elolvasni, abba a banális problémába ütközünk, hogy túl kicsi betűkkel van nyomtatva. Ezen már könnyen tudunk segíteni (WEAR SPECTACLES, READ BOOK). A könyvből megtudhatjuk, hogy Xerdont csak olyan nyílveoszóval tudjuk elpusztítani, mely a szent fűzfából készült. A szent fűzfa őrzője pedig ezüstre vágyik. Keressük meg ezt a fűzfát (DROP BOOK, GET SWORD, U, S, E, E). Egy mocsarat találtunk. A fűzfa erősen vízigényes, irány tehát a mocsár (GO MARSH, E, N)! Íme tehát a szent fűz, előtte az őrzője. Hiába magyarázzuk neki, hogy nekünk csak egy ág kell, nem enged a közelébe. Annyira azért nem hajthatatlan, mint tudjuk, ezüstre vágyik... (THROW MEDALLION). Ettől mindjárt jobb belátásra tér (CUT BRANCH) - a kard kell hozzá. Van nyílhegyünk, van sastollunk és immár mágikus fűzfavesszőnk is, vagyis minden összetevőnk megvan. A nyílveoszót azonban csak a nyílkészítő mester tudja nekünk elkészíteni. Kalandunk a második részben folytatódik.

Kiss László (Lacika)

dBase II. 2.43 (IS-DOS) – II. rész

2. A dBase II.

Mikroszámítógépes környezetben talán a legelterjedtebb adatbázis-kezelők a dBASE család tagjai. Mi a dBase II CP/M változatát fogjuk használni, ami alapvetően 8 bites számítógépekre készült.

2.1 A dBase II fő jellemzői

- interaktív (párbeszéd) üzemmód - az adatok közvetlenül hozzáférhetők és módosíthatók
- programozható - az adatbázis-állományok elrejtethők a „végfelhasználó” előtt; csak a program által ellenőrzött módon lehet az állományokban bármit módosítani
- önálló, saját nyelve van, amely rendelkezik a magasszintű programozási nyelvek tulajdonságaival

A dBASE II egy relációs adatbázis-kezelő rendszer. Parancsfelépítése angol nyelvű (de ENTERPRISE-ra készült magyar nyelvű változat is). Meglévő adatállományainkat egyszerű parancsokkal bővíthetjük, módosíthatjuk. A szükséges információ másodpercek alatt kikereshető. A program egyszerre több szempont szerint is tud rendezni; egy időben kettő nyitott adatbázis-állományt tud kezelni, ezek tartózkodási helyét nem kell megjegyezni, mert a dBASE által adott hivatkozási nevük segítségével bármikor és bárhol megtalálhatók. Olyan parancsok és függvények állnak rendelkezésünkre, melyek a gyorsabb programfuttatást, tesztelést teszik lehetővé, használatukkal programjaink egyszerűbbek és ügyesebbek lehetnek.

2.2.A dBASE II állománytípusai

A használható, különböző típusú állományok egy részét a felhasználó saját igényei szerint állítja elő, a többit az adatbáziskezelő készíti el egyes dolgok adminisztrálására. (Végeredményben ezek létrejötte is tőlünk indul el, bizonyos parancsok kiadásakor.)

Amikor mi adunk nevet egy új, létrehozandó állománynak, ezt csak a CP/M operációs rendszer szabályait figyelembe véve tehetjük meg (ezek a szabályok megegyeznek az MS-DOS szabályaival): Az ún. elsődleges **állománynév** legfeljebb 8 karakteres lehet, betűvel kezdődjék, betűket és számjegyeket tartalmazhat. A jobb olvashatóság kedvéért az elsődleges névben szokás és megengedett használni az „aláhúzás” karaktert (_). Az elsődleges nevet követheti egy ponttal (.) kezdődő, ezenkívül még legfeljebb 3 karaktert - betűt és számjegyet - tartalmazó ún. **kiterjesztés**, mely általában az állomány tartalmára utaló típusjel. A dBASE II használata során legtöbbször elhagyható, mert a parancsok többsége automatikusan generál az elsődleges név mellé egy alapértelmezés szerinti kiterjesztést. Az alapértelmezés helyett megadhatunk mást is, de ilyenkor ezt az állományra való összes hivatkozáskor fel kell tüntetni. Ekkor az is problémát okozhat, hogy bizonyos parancsok szolgáltatásait nem tudjuk kihasználni, mert azok működésük során csak a szabványos típusjelű állományokat veszik figyelembe. Vannak parancsok, melyek egyszerre több állományra vonatkozhatnak. Ezeknél használhatunk ún. nem egyértelmű állománynéveket, melyek egy állománycsoportot azonosítanak. Ezek képzése követi az operációs rendszerben megszokott szabályokat:

a ,*' karakter tetszőleges számú,

a ,?' egyetlen tetszőleges karaktert helyettesít.

(Az operációs rendszerben szokás ezeket „joker” karaktereknek nevezni.) Például: az aktuális lemezegységen található olyan állományokra, melyeknek neve a „munka” szóval kezdődik, a „munka*.*” névvel hivatkozhatunk, az összes állomány jelölésére a „*.*” név alkalmas. (Ez csak az aktuális könyvtár állományait jelenti!)

Röviden tekintsük át a dBASE II által kezelt állománytípusokat. Tulajdonságait, szerepüket részletesebben a használatuk ismertetésekor vizsgáljuk. (Zárójelben megadjuk az eredeti angol elnevezést is.)

1. Adatbázis-állomány (DATABASE FILE)

.DBF típusjele van

A nyilvántartandó adatokat tartalmazza rekordokban és mezőkben, Legfeljebb 65535 rekordja lehet.

2. Indexállomány (INDEX FILE)

.NDX típusjele van

Logikailag rendezett adatbázis-állomány rekordsorrendjét tartalmazza.

3. Formátumállomány (FORMAT FILE)

.FMT típusjele van.

Jelentések megjelenítéséhez szükséges parancsokat tartalmazza, melyek a felhasználó által definiált képernyőforma létrehozására alkalmasak. Szabványos szöveges (ASCII) állomány.

4. Memóriaállomány (MEMORY FILE)

.MEM típusjele van.

Legfeljebb 64 memóriaváltozó megőrzésére hozható létre, a memóriaváltozó nevét és értékét is tárolja.

5. Parancs-, program- és procedúraállományok (COMMAND, PROGRAM, PROCEDURE FILE)

.CMD típusjelük van

Szöveges állományok olyan dBASE utasítások tárolására, amelyek végrehajtását egyetlen paranccsal elindíthatjuk.

2.3. Az adatbázis-állományok szerkezete

Munkánk az adatbázis-állományok körül forog, ezek tartalmaznak rekordokba (sorokba) és mezőkbe (oszlopokba) szervezve adatainkat. A táblázat felépítését létrehozáskor a felhasználónak kell meghatároznia. Ilyenkor kell megadni, hogy a táblázat egy sora (egy rekord) milyen mezőket tartalmazzon, és pontosan definiálni a megadott mezők tulajdonságait. Adatbázis-állomány létrehozásakor tartsuk szem előtt a méretkorlátozásokat (ld. a függelékben)!

A meződefiniáció részei:

Mezőnév

- a mezők későbbi azonosítására szolgál
- legfeljebb 10 karakter hosszú
- betűvel kezdődik
- betűt, számjegyet és aláhúzás karaktert tartalmazhat
- célszerű értelmes, a tartalomra utaló neveket választani

Mezőtípus	definiáláskor egyetlen betűvel kell jelölni: C = karakteres (character) N = Numerikus (numeric) L = logikai (logic)
Szélesség	a mezőben tárolni kívánt karakterek vagy számjegyek maximális száma. - numerikus mező esetén a tizedespont és a tizedesjegyek is beleszámítandók. Ilyenkor külön kell megadni, hogy a definiált teljes hosszból hány karakteren akarjuk tárolni a tizedesjegyeket. - a logikai típusú mező hossza kötött, megváltoztatni nem lehet, megadni nem kell.

2.4. Adattípusok

A dBASE II 3-féle adattípust ismer, ezeket élesen megkülönbözteti egymástól, a kifejezésekben típuskeveredést nem enged meg. (Mint láttuk, adatbázis-állomány mezője mind a három típust felveheti.)

- **Karakteres:** nyomtatható (billentyűzetről beírható) karakterekből álló szöveg (legfeljebb 254 karakter hosszú lehet). Így gazdaságos tárolni minden olyan adatot, amellyel nem végzünk matematikai műveleteket.
- **Numerikus:** előjeles szám (számjegyeket, tizedespontot, előjelet (+,-) tartalmazhatja. Ha egy adattal később matematikai műveleteket szeretnénk végezni, így kell nyilvántartani. A műveleteket és függvénykiértékeléseket 10 jegyig pontosan végzi el a rendszer.
- **Logikai:** mindig egyetlen karakter, a logikai IGAZ vagy HAMIS érték. Jelölésük:

IGAZ (TRUE/YES) = T,t
HAMIS (FALSE/NO) = F,f

2.5. Állandók és változók

Adataink a munka során lehetnek változatlan értékűek (ezeket nevezzük állandóknak vagy konstansoknak) vagy változók. Az adatbázis-állományokon kívül tárolt adatok szintén háromféle típusúak lehetnek: karakteres, numerikus, logikai. Ha egy karaktert vagy karaktersorozatot (karakterláncot) állandóként akarunk a géppel közölni, ún. határolójelek közé kell tenni. A karakterlánc-határoló háromféle lehet:
felső vessző (apoztróf) 'xxxxxx'
idézőjel „-----”
szögletes zárójel [.....]
Természetes, hogy a határolójel típusa a karakterlánc mindkét végén meg kell, hogy egyezzen, így a másik két határolójel a karakterlánc belsejében is szerepelhet.

A numerikus állandók (számok) a megszokott módon közölhetők a géppel.

Logikai állandók begépelésekor a megfelelő betűt (ld. az adattípusok ismertetésénél) két pont közé kell tenni (pl. ".T.").

A változók végeredményben tároló helyek, amelyekre azonosító (nevük) segítségével hivatkozhatunk, tartalmukat lekérdezhajtuk, módosíthatjuk. A dBASE kétféle változót különböztet meg:

1. változónak tekinti minden nyitott adatbázis összes mezőjét

2. tárolhatunk az adatbázisoktól függetlenül is adatokat az **memóriaváltozóknak**.

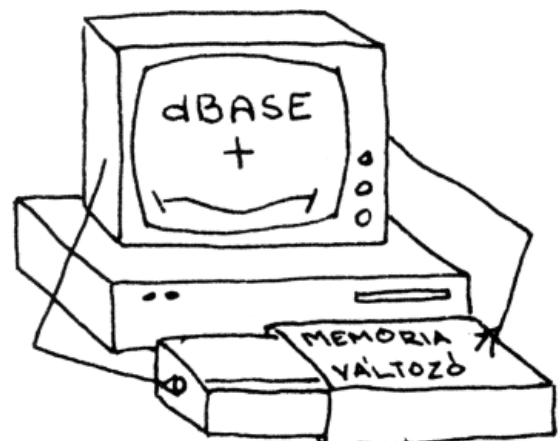
2.6. Memóriaváltozók

Az adatbázisoktól független tároló helyeket - megkülönböztetésül - memóriaváltozóknak nevezzük. Ezeket általában ideiglenes tárolásra használjuk (elsősorban programokban, pl. a felhasználótól való adatbekérésre, a végrehajtás ellenőrzésére). Nevük ugyanolyan szabályok szerint képezhető, mint az adatbázismezőké, legtöbb esetben ugyanott és ugyani használhatók, mint a mezők. Ha egy memóriaváltozó neve megegyezik az aktuális adatbázis-állomány egy mezőjének nevével, akkor e név beírása mindig a mezőre vonatkozik. A memóriaváltozóra úgy tudunk hivatkozni, hogy neve elé írjuk „családnevét”: egy „m” betűt és egy „nyilat” (pl. ha a változónk neve „datum”, ezt az „m->datum” karaktersorozattal tudjuk megnevezni nyilat a kötőjelből és a nagyobb jelből kell összeállítani). Egyszerre legfeljebb 64 memóriaváltozónk lehet. A memóriaváltozók törölhetők vagy későbbi felhasználás céljából lemezes állományba is elmenthetők.

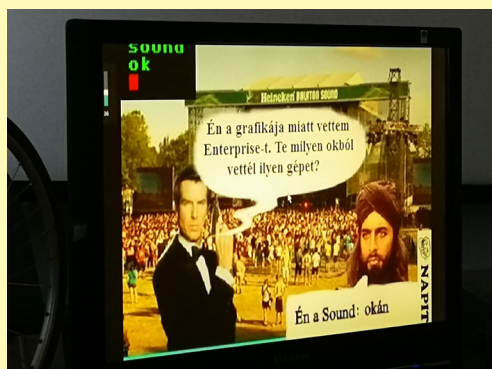
A változók deklarálására külön utasítás nincs. Egy eddig még nem íjok használt memóriaváltozóra először csak olyan környezetben hivatkozhatunk, amelyből egyértelműen kiderül a memóriaváltozó típusa és mérete. Ez az értékadó utasításokkal valószínűleg meg, amelyek mind képesek egy új, adott nevű és típusú memóriaváltozó létrehozására. Ha már létezik egy ugyanilyen nevű, az megszűnik, felülíródik. Egyetlen parancs van, amely nem tud létrehozni új memóriaváltozót (a @...GET parancs), csak egy előzőleg definiált értékét tudja megváltoztatni (típusát, hosszát nem).

Ezzel végére értünk a dBASE II által használt objektumok ismertetésének. Volt szó állományokról, mezőkről, memóriaváltozókról. Láttuk, hogy minden objektumnak nevet kell adnunk, ennek szabályait szintén megismertük. Szeretnénk felhívni a figyelmet arra, hogy egyszer sem mondtuk, hogy bizonyos szavakat nem használhatunk objektumok elnevezésére. Erről nem véletlenül feledkeztünk meg, hanem egyszerűen azért nem említettük, mert ilyen szabály nincs. A dBASE II megengedi, hogy különböző állományainknak, változóinknak olyan nevet adjunk, amely egyébként egy parancsot vagy egy paramétert jelent. Azonban az ilyen nevű objektumok használata félreérthetővé teszi utasításainkat, nemcsak számunkra, de sokszor a dBASE számára is. Ezeket később nem biztos, hogy megfelelően tudjuk használni, néhánynak a további használata teljesen lehetetlenné válhat.

Folytatjuk!



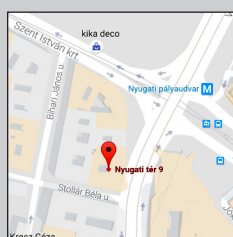
Enterprise Klub 2018. szeptember 8.



ENTERPRISE KLUB

Egy évben 6 alkalommal

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



További 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@enterpress.news.hu

ENTERPRISE FOREVER

<https://enterpriseforever.com>

ENTERPRESS Magazin - 2018/4-5. július-október

Főszerkesztő: Matusa István

Szerkesztőségi főmunkatárs: Németh Zoltán (Zozosoft)

A csapat: geco, Povi, Kiss László, SzörG, szipucsu, lgb, Hajdó Máté †

Design, nyomdai előkészítés: Matusa István

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

E-mail: info@enterpress.news.hu

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

ENTERPRESS e-magazinok:

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