

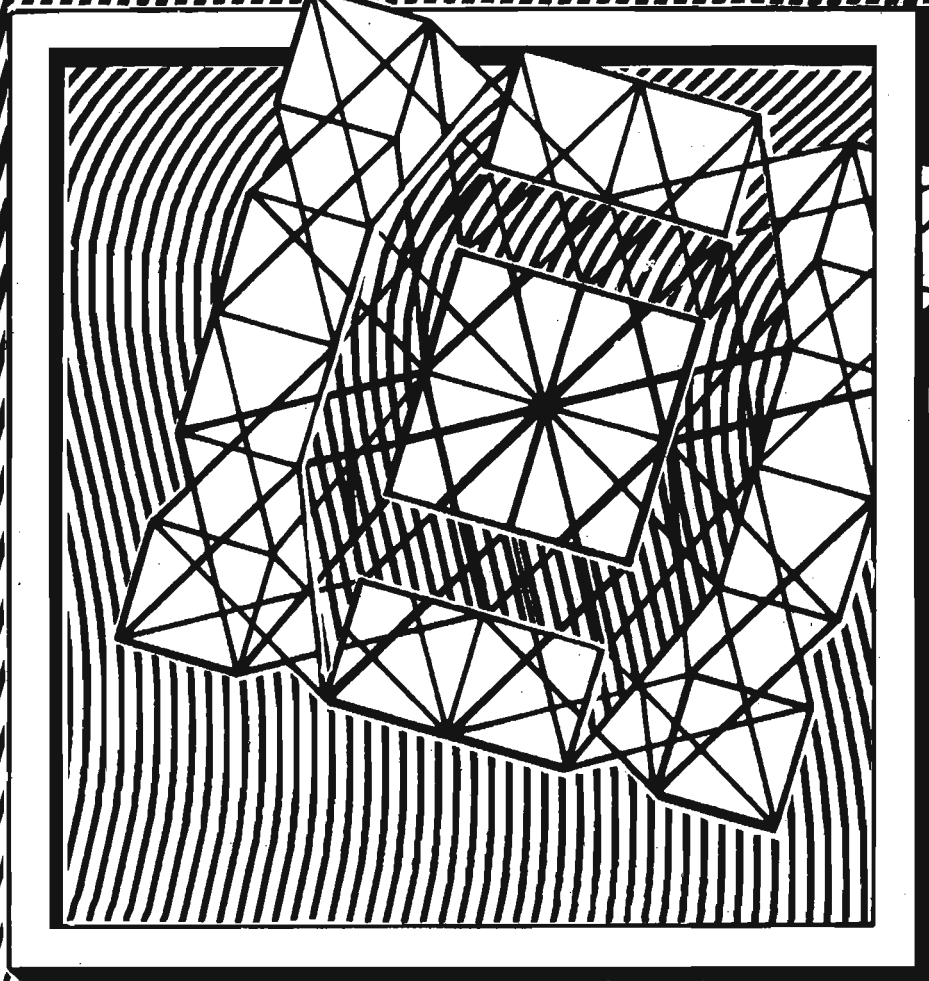


**SORD
AMSTRAD/
/SCHNEDER**

1

88

602



Vážení přátelé klubu SORD !

Srdečně Vás zdravím na začátku druhého ročníku našeho společného ZPRAVODAJE. A protože se začíná čile rozebíhat korespondence zejména s mimopražskými členy, vybral jsem z dopisů to, co by mohlo zajímat i ostatní:

.... co je s deskami pro EM 64 kB a PI 5 ...

to je snad nejčastější dotaz ve Vašich dopisech. Věřte, že nás velmi mrzí, že Vám nemůžeme uspokojivě odpovědět. Největší problémy jsou administrativního rázu. K tomu přistupuje ještě požadavek výrobce (kompletní dokumentace vč. technického popisu a funkčního vzorku). Není mnoho možností, jak tento problém řešit. Zatím vidíme jedinou cestu. V knihovně klubu bude uloženo několik exemplářů dostupné TD, aby si ji zájemci mohli vypůjčit a zajišťovat si paměť vlastní cestou. Stejně chceme postupovat pokud jde o připojení RAM DISKU. Poslední poznámku berte, jako výzvu k zaslání potřebné dokumentace.

.... návrh na schůzku vedoucích klubů SORD z ČSSR...

nám zaslal předseda klubu z Košic Tibor Menyhert. Jeho dopis jsem předal Petrovi Musilovi, ale jistě by bylo zajímavé kdyby se ozvali i předsedové dalších klubů z ostatních měst. PÍŠTE !

.... připojení RAM DISKU ...

a ostatních periférií, jako zapisovače XY 4131 z Laboratorních přístrojů zajímá nejenom Z.Bombíka z Prosečnice, ale řadu dalších. Chceme se touto problematikou v příštích číslech zabývat. Proto uvítáme jakékoli informace z této oblasti.

.... požadavky na různé programy.

Dopisy tohoto druhu předávám V.Zemanovi, který má na starosti programové vybavení. Na jiném místě najdete první programovou nabídku vč. pokynů pro jejich získání.

.... Chyby v tisku programů.

Jak je vidět zařadili jsme se mezi řádné tiskoviny a máme zřejmě i vlastního tiskařského šotka. Ale vážně. Několik z Vás nás upozornilo na chyby v programu MULTIUSER. Omlouváme se za ně Vám všem i autorovi. Chyby vznikly při přepisování. Oč méně chyb by bylo, kdyby jste své příspěvky posílali na kazetě rovnou v TASNORDU, nebo jako záznam programu. (Kazety samozřejmě vrátíme). A nyní opravy:

ř.285 za tab(9) má být chr\$(225)

ř.345 špatná proměnná read XL

ř.1318 chybí "D" v příkazu RDST\$

ř.368 ohlásí se chyba 17 - stejně navěští (nutno zrušit mezeru-).

Na všechny dopisy se snažím odpovědět přímo autorům, pokud mi to čas dovolí. Ostatní doufám najdou odpověď ve ZPRAVODAJI i když se spoždění a nebo jim odpoví moji kolegové, kterým předám ty dotazy, na které nemohu odpovědět sám. Přeji Vám příjemnou zábavu nad naším čtvrtletníkem v roce 1988. Budeme se snažit, aby odpovídal Vašim požadavkům tak, jak jste je uvedli do naší ankety s jejímiž výsledky Vás seznámíme v příštím čísle.

Ing.Petr Čihula

Programová nabídka SORD klubu

Protože do nabídky MIKROBÁZE je možné zařadit pouze původní programy na určitém stupni profesionálního zpracování, hledali jsme vlastní cestu jak zdostupnit další "lidové", nicméně zajímavé a potřebné programy. Toto je první pokus a doufáme, že nezůstane jenom u něho.

Co Vám nabízíme:

MS.DIS program disassembler specializovaný na MONITOR ROM
 =====
 a BF. Tento programový blok dále obsahuje blok dat pro MONITOR ROM, MONITOR ROM i BF a MONITOR BASIC F ROM.

TEXT V tomto bloku jsou programy :
 =====
 ERRORNIK - seznam hlášení Err pro BF
 SOUBOR Z 88 - seznam instrukcí Z 88
 TASWORD - textový editor
 TW-návod - návod k používání TASWORDU

JAZYKY BASIC 6
 =====
 BASIC F - programovací jazyky pro 64 kB
 FALC
 DBASIC - rozšíření BASICU F

MS.MON jednoduchý disassembler
 =====
 kopírovací programy
 zavařič MONSU výkonný, relokovatelný ladící prostředek ze ZX-SPECTRA

MATEMATIKA blok 55 různých matematických programů. Seznam
 =====
 neuvádíme pro nedostatek místa

HUDEBNÍ blok 18 programů demonstující hudební možnosti
 =====
 počítače SORD

HRY blok nejžádanějších her, (např. Města, M.MINER
 =====
 BUDIČ, COBRA, POOYAN, atd.)

ASSEMBLER programy v BI a BG zabývající se assemblerem
 =====

Jak programy získat:

Tuto činnost jsme museli řešit tak, aby náklady pro obě strany byly co nejmenší, stejně, jako nutná administrativa. V knihovně klubu bude od každého bloku uloženo cca 10 nahraných kazet. Ty si bude možno vypůjčit a přehrát. Vypůjční doba bude 14 dní pro pražské zájemce, 1 měsíc pro mimopražské. Kazety je možné vrátit poštou. Doufáme, že se tato služba osvědčí. Přivítáme Vaše připomínky stejně, jako nové programy kterými naši nabídku rozšíříme.

Tuto službu bude zajišťovat stejně, jako půjčování dokumentace z knihovny V.Zeman.

 * Pozn: Ve zpravodaji MIKROBÁZE č.2 bude nabízen programovací *
 * jazyk FORTH pro počítač SORD #5. *

Úpravy v kopírovacím programu MR.COPY

Popisované úpravy se týkají programu:

MR.COPY.CM 03 F050 FD40 0CFC F400

Program obsahuje několik závažných chyb. Stává se že se program zacykli, nebo dokonce zhroutí. Stává se to, když má plný seznam položek a je-li někde mezi nimi typ .DT nebo .LS. Obecně lze říci, že soubory typu .DT a .LS nahravá a kopíruje MR.COPY záhadným způsobem a to ne vždy správně. Některé z těchto chyb lze odstranit celkem snadno několika "pouky".

Program se zhroutí když:

= se v COPY zadá X-Y, přičemž platí, že X>=Y. Za normálních okolností (tj. když X<Y) se kopírují programy X až Y. Zhroutení lze zabránit:

POKE &FA7D,&38,&FA,&18,&CB

Po této úpravě se kopíruje jen X a ostatní je ignorováno. = se vícekrát než Xkrát stiskne nefunkční klávesa (vypíše-li se znovu DISH:). Je to způsobeno tím, že se nevybere návratová adresa a STACK po čase přepíše hlavičky jednotlivých programů a potom i program. Odstranit lze:

POKE &F525,&D8 (JR 0F4E0H na JR 0F4FEH)

Po této úpravě se při stisku špatné klávesy nevypíše znovu DISH: Odstranění dalších chyb:

Soubor typu .SC je hlášen jako .VM a soubor .LS jako .DT. Chybu lze částečně napravit:

POKEW &F564,&534C:POKE &F772,&77 potom se správně hlásí .LS. Napíšeme-li OLD a přepneme-li kurzor do grafického modu, po nahrání je kurzor ve 4-tém nedefinovaném modu a program samozřejmě na naše povely nereaguje. Opravit lze:

POKEW &F4E6,&9136 (SET 0, (HL) na LD(HL),91H - v HL je &701A) Program si pak po nahrání nastaví vždy zvuk, kurzor na CAPS atd. Úpravou: POKEW &F4E6,&0618:POKE &F4EA,0 (CALL ... na JR ...) se ušetří jedna zbytečná návratová adresa.

Další možné úpravy

Změna barvy popředí a pozadí:

POKE &F4A7,&XY (X - fcol, Y - bcol)

Změna délky pauzy před úvodním pískem (pauza mezi programy):

POKEW &F97B,&0E00 - cca 4 sec
&1100 - cca 5 sec
&1F00 - cca 10 sec

Změna délky IBG (písk mezi bloky a za hlavičkou)

POKE &F993,1 256 pulsů
2 512 pulsů standart
3 768 pulsů atd.

Změn by bylo potřeba provést ještě mnoho, ale nelze je již provést pouhým poukovaním nových dat.

Roman Švec

HODINY REÁLNÉHO ČASU

Běh systému M-3 zajišťuje podprogram monitoru uložený od adresy 01DFh. Tento program se mimo jiné dotazuje na obsah systémové proměnné SPRPRC (SPRite PRoCes), kde je uložena adresa odskoku. U BG 5765h, u BFG 8F22h. Na těchto adresách je uložena rutina realizující příkaz MOVE. Protože BASIC-F příkaz MOVE nemá, je adresa odskoku 002E (na instrukci RET). Je tedy možné provádět vlastní kratší programy ve strojovém kódu. Stačí do proměnné SPRPRC (7053h) uložit vstupní bod našeho programu, který musí být ukončen instrukcí RET. Spuštění programu se provede z BASICu příkazem POKEW &7053, vstupní bod programu, zastavení pak POKEW &7053,002E. Program se provádí v závislosti na CTCMS. Na ukázkou uvádíme program hodin reálného času.

Nastavení hodin: POKE &704C, hod
 POKE &704B, min
 POKE &704A, sec

F000	F3	DI	F037	CD5101	CALL 0151
F001	CD040D	CALL 0D04	F03A	3D	LD E, L
F004	111701	LD DE, 0117	F03B	21AF71	LD HL, 71AF
F007	212700	LD HL, 0027	F03E	C630	ADD A, 30
F00A	CD5C13	CALL 135C	F040	77	LD (HL), A
F00D	015000	LD BC, 0050	F041	2B	DEC HL
F010	115F71	LD DE, 715F	F042	7B	LD A, E
F013	2123F0	LD HL, F023	F043	C630	ADD A, 30
F016	EDB0	LDIR	F045	77	LD (HL), A
F018	215370	LD HL, 7053	F046	2B	DEC HL
F01B	3E5F	LD A, 5F	F047	363A	LD (HL), :
F01D	77	LD (HL), A	F049	7D	LD A, L
F01E	23	INC HL	F04A	3D	DEC A
F01F	3E71	LD A, 71	F04B	327871	LD (7178), A
F021	77	LD (HL), A	F04E	216C71	LD HL, 716C
F022	FB	EI	F051	34	INC (HL)
F023	3E4A	LD A, 4A	F052	14	INC D
F025	326C71	LD (716C), A	F053	7A	LD A, D
F028	3EAF	LD A, AF	F054	FE03	CP 03
F02A	327871	LD (7178), A	F056	C26B71	JP NZ, 716B
F02D	1600	LD D, 00	F059	0600	LD B, 00
F02F	3A4A70	LD A, (704A)	F05B	111030	LD DE, 3010
F032	2600	LD H, 00	F05E	21A871	LD HL, 71A8
F034	6F	LD L, A	F061	CD6014	CALL 1460
F035	3E0A	LD A, 0A	F064	C9	RET

Vzhledem k nedostatku místa uvedeme podrobný popis v některém dalším čísle.

P. Bláha
 P. Musil

HLASOVÝ VÝSTUP HV 01 - konstrukční návod

=====

Patříte-li mezi ty, kterým se zdá, že obrazovka a tiskárna, jako jediná standardní zařízení pro styk počítač -> člověk, jsou málo, pak jsou vám určeny následující řádky. Popisují konstrukci poměrně jednoduchého hlasového výstupu (dále jen HV) s dostatečnou srozumitelností. Všeobecný popis HV jsem uvedl v [1]. Kromě příslušných součástek budete potřebovat základní znalosti o IO a několik hodin práce. HV je připojen přes konektor pro tiskárnu, takže v nejhroším případě můžete zničit v počítači příslušný střadač.

Vlastní program nelze pro svou obsáhlou prozatím publikovat, ale bude k dispozici na všech burzách programů klubu SORD 682. IO a v naléhavých případech i přímo u mě.

POKYNY PRO OŽIVOVÁNÍ:

Deska s plošnými spoji není prozatím navržena, takže musíme použít univerzální desku nebo udělat vlastní návrh. Před osazením desky přeměříme všechny pasivní součástky. Kondenzátory s přesností 1% můžeme skládat ze dvou kondenzátorů paralelně. Nemáme-li možnost měřit kapacitu, lze použít nevybírané kondenzátory z řady E12. (Tento postup jsem zvolil i já a výsledek je docela uspokojivý.)

Po osazení pasivních součástek a případných propojek zaletujeme IO, přičemž obvod 4876 letujeme poslední a dodržujeme zásady pro práci s obvody CMOS. Nezapomeneme na několik blokovacích kondenzátorů. V nouzi můžeme nahradit IO 4876 obvodem 7475, který však není pinově kompatibilní (obvod CMOS je použit z důvodu, že má logické úrovně 0 a 5V oproti asi 0.5 a 3V u TTL). Pozn. všechny součástky jsou číslovány podle původní zprávy, takže se nenechte překvapit vysokými čísly.

Napájecí napětí vyvedeme nejlépe ze zdroje počítače na pětikolíkovou zásuvku DIN (zapojení dutinek podle plánu). Zásuvka se bez problémů vejde nad vypínač zdroje. Konektor na připojení k počítači vyrobíme například z konektoru FRB.

Desku ještě jednou prohlédneme a připojíme ke zdroji, nejlépe přes ampérmetr (odběry řádově deset miliampér), poté připojíme desku k počítači přes konektor pro tiskárnu.

Měříme voltmetrem napětí na výstupu D/A převodníku (pin 4 MDAC 08). Spustíme program 1, napětí by mělo být asi -1V. Po opakovaném tisknutí libovolného tlačítka by napětí mělo stále stoupat až asi na -2V, poté skokově klesnout na 0V a opět postupně vystoupat na -1V.

Poté měříme napětí na výstupu váhových odporů (pin 13 obvodu A 273D). Spustíme program 2 a tiskneme libovolné tlačítko. Napětí by mělo z počátečních asi 3V při každém stisknutí o něco stoupnout a po posledním (15.) stisknutí být téměř 5V. Odpor R77 vybereme tak, aby počáteční napětí bylo asi 3.2V a případně ho úplně vypustíme.

Nakonec vyzkoušíme celek - spustíme opět program 2 a připojíme výstup HV k zesilovači - ozve se slabý zvuk, který by měl při každém stisknutí tlačítka zesilovat. Pokud funguje i toto normálně, můžeme nahrát program PREDVAĎ, který si nahraje vlastní program ve strojovém kódu HV01-put a řekne nám základní informace o HV.

OVĽADÁNÍ HV:

HV je nadefinován jako vlastní zařízení s názvem HV:. Prozatím umí zpracovávat pouze foneticky zadané texty.

Háčkované hlásky píšeme malými písmeny, ostatní hlásky písmeny velkými. Dlouhé samohlásky píšeme zdvojeně. Mezi jednotlivými slovy není třeba dělat mezery, protože se i v normální řeči vyslovují dohromady. Chceme-li mezeru přesto udělat, použijeme znak '\' (5Ch) nebo '!' (7Ch) (používá se např. před slovy začínajícími samohláskou nebo pro zdůraznění). Používáme pouze jedno I a chceme-li měkkit, použijeme háčkované písmeno (např. dETIsLIDOI ASTRONOMICKEE IOPSERVATORIE). Místo CH píšeme X, místo X píšeme KS a místo Q KV (např. KSAVEROFSKAAKURATAJSOUXITRAA). Neznámé znaky (tj. vše kromě existujících písmen a znaků \ a !) HV ignoruje.

Příklad ovládání (pro BASIC-F):

```
CLEAR ,&E7FF
OLD "HV01-put"
OPEN "HV:" AS #1
T$="VIISTUP"
PUT #1 "HLASOVII",T$,"PrIPRAVEN"
CLOSE #1
```

Příkaz PRINT je rezervován pro pravopisně psanou prosluvu včetně číslovek a některých speciálních znaků.

Příklad ovládání (pro BASIC-G):

```
CLEAR ,%E7FF
OLD "HV01-put"      (je nutno odstranit autostart)
T$="AHOJJAKSEMAAs"
CALL %E81D,,XCHG(LEN(T$)),VARPTR(T$)+2
```

Struktura programu:

E800-E806 - podprogram pro nadefinování vlastního zařízení
E807-E81C - definice zařízení HV:
E81D-E875 - program pro nastavení CTC, registrů a spuštění
podprogramu pro generování promluvy
E876-E9CF - volné místo
E9D0-E9DF - tabulka časovacích konstant pro CTC
E9E0-E9FF - tabulka adres pro dekódování hláskových dvojic
EA00-ECFF - tabulka 12 periodických vln
ED00-EFFF - tabulka šumové vlny
F000-F7FF - popisy hláskových částí
F800-F9E3 - podprogram pro generování promluvy
F9E6-FCE7 - proměnné a pracovní oblasti
FCE8-FFFF - volné místo (rezervováno pro obsluhu PRINT)

HV využívá pro svoji funkci kanál 0 CTC. Půlsničkové přerušeni od VDP přes kanál 3 CTC pracuje, ale jenom ve zjednodušené formě - obsluha systémových hodin a test na SHIFT+RESET.

PROGRAM 1:

=====

```
10 OUT %50,1
20 FOR N=0 TO 255
30 OUT %40,N
40 REPEAT:UNTIL INKEY$<>""
50 NEXT N
```

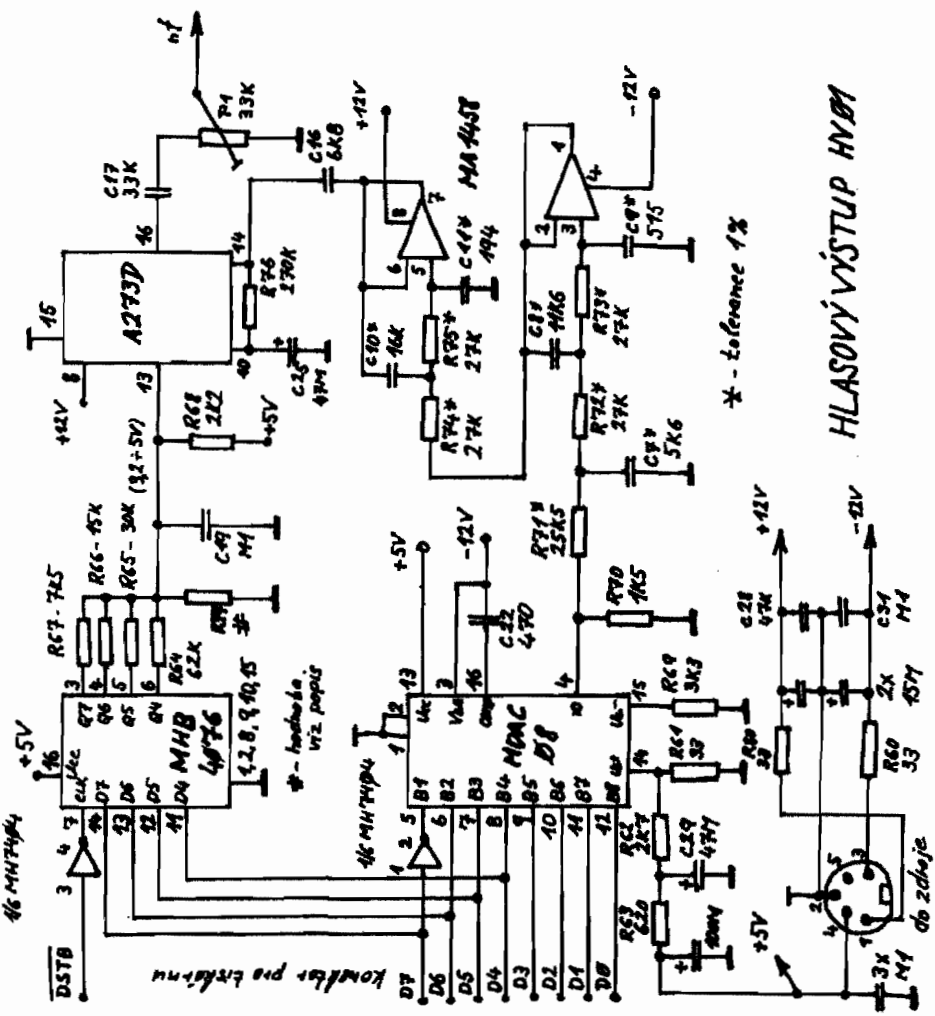
PROGRAM 2:

=====

```
10 FOR N=0 TO 255 STEP 16
20 OUT %50,0
30 OUT %40,N
40 OUT %50,1
50 REPEAT
60 OUT %40,127:OUT %40,128
70 UNTIL INKEY$<>""
80 NEXT N
```

Literatura:

- [1] JELÍNEK Jan: Hlasový výstup HV 01 pro SORD MS, zpravodaj SORD+SCHNEIDER č. 1/87
[2] DVORÁK Pavel: Hlasový výstup pro ASDŘ, zpráva VÚST č. 1718 00 742/1, 16.3.1984



Komplet pro tréningu

* - tolerance 1%

HLASOVÝ VÝSTUP HV01

ob zdoje

KOMPRESIE OBRAZOVKY.

Každý z nás má na disketě nebo na páse nějakou tu hru. Většina z nich začíná malým programkem v Basicu, který zadefinuje pár symbolů, někdy třeba ne, nastaví horní mez paměti na novou hodnotu, i to někdy třeba ne, a pak se vykoná řádek s instrukcí LOAD "OBRAZEK.SCN",42152, tohle, naopak, vždycky. A následuje 16 kilobajtů na obrazovku. Máme-li pásku, počkáme, až poporoste tráva, máme-li disketu, netrvá to sice tak dlouho, ale zas nám je líto těch 17 kB, které užírají drahý diskový prostor jenom proto, aby bylo chvilku na co koukat.

Je tu samozřejmě možnost obrázky jednoduše umazat. Ale příliš to připomíná zvyk chodit do kina až po titulcích. Napsal jsem si tedy program, který obrazovku zkrátí. Jeho výpis je dole. Až si natukáte těch šestnáct řádků, získáte program KOMPRESIE.BAS, který pracuje jednoduše a spolehlivě takto. Nejdříve se optá na jméno obrázku, který si přejete zhustit, pak ho natáhne do obrazovky, a vytvoří kód, kde každou skupinu typu byt,byt,...,byt (n krát) zapíše do dvou bytů v podobě n,byt. Kdyby hrozilo, že místo komprese dojde k dekompresi, že by totiž místo byt1,byt2,byt3,byt4,..., zapisoval 1,byt1,1,byt2, 1,byt3,1,byt4,..., změnil taktiku a zapíše raději 0,n,byt1,byt2,...,bytn. Takto vytvořená data - po vašem souhlasu - uschová s názvem původní obrazovky a na začátek tohoto datového souboru přilepí dekódér. Dekódér je plně relokovatelný. Na vás zbyde už jen maličkost.

Přepíšete ve hře tu řádku, kde původně bylo LOAD "SCREEN.PIC", na: MEMORY m-1:LOAD "SCREEN.PIC",m:CALL m. Hodnotu m pochopitelně zvolíte s ohledem na to, aby se stlačeny soubor do paměti vešel a aby s ničím nekolidoval, například aby nepřepisoval nově definované symboly, pokud jsou. Hodnota m=16000 vyhovuje téměř vždycky.

Je zřejmé, že hodně členité obrázky s mnoha detaily se komprimuje hůře, než obrázek s velkými jednobarevnými plochami. Rekord v nestlačitelnosti jsem zaznamenal u Ghosts and Goblins, kde výsledek má plných 16 kB. Naopak to krásné auto formule F1 z 3D Grand Prix se vejde do 6 kB a SABOTER.PIC má po kompresi všeho všudy 7 kB. Takže posuďte sami.

PROGRAM KOMPRESI.BAS

```
10 MEMORY &1FFD:kam=&2000:IF PEEK(kam)=221 THEN 70
20 FOR dato=0 TO 1
30 READ cod$,soucet:kontrola=0
40 FOR j=0 TO 93:co=VAL("&"MID$(cod$,2*j+1,2)):POKE kam,co:
   kam=kam+1:kontrola=kontrola+co:NEXT
50 IF kontrola<>soucet THEN PRINT "CHYBA V RADKU ";
   150+dato*10;"!":POKE &2000,0:IF dato=0 THEN
   EDIT 150 ELSE EDIT 160
60 NEXT
70 MODE 1:INPUT"Jmeno obrazovky ke kompresi ";jmeno$
80 jm$=jmeno$
90 LOAD jmeno$,&C000:CALL &2000
100 MODE 1:delka=PEEK(&1FFE)+256*PEEK(&1FFF)-&20BF:
   PRINT UPPER$(jm$);" ma nyní delku";delka;"bytu."
110 PRINT"Zapsat na disk?(A/N)";
120 a$=INKEY$:IF a$="" THEN 120
130 IF UPPER$(a$)<>"A" THEN END
140 SAVE jm$,b,&2090,delka
150 DATA DD21BB202100C0E5FDE1FD7E00FDBE01202106004E78FEFF280
   A04237CB528597EB928F1DD7000DD7101DD23DD23E5FDE118D7
   FD7E01FDBE0228D70600DDE5D1131304EDA003FD2378FEFF280
   C7CB52814FD7E00FDBE0120E9DD360000DD,11352
160 DATA 7001D5DDE118A5DD360000DD7001DD360200DD360300ED53FE1
   FC9DD7000DD7101DD360200DD360300DD23DD23DD22FE1FC9F3
   3EC932FFBECDFBE3B3BE1FB112200191100C07E2347B72B087
   E23121310FC18F27EB7CB234FEDB018E900,10425
```

Pozn.

Řádky 40, 50, 150 a 160 jsou přesunuty kvůli formátu textu pod sebe.

ZÁZNAM PROGRAMŮ A DAT NA DISKU

Formátování

Celý disk je rozdělen na 40 stop, které si můžeme představit jako soustředné kružnice. Stopy jsou očíslovány,

vnější stopě přísluší č. 0, vnitřní č. 39. Při formátování rozdělí operační systém každou stopu na 9 sektorů. Každý sektor obsahuje 512 bajtů informací (t.j. 0.5 kB). To znamená, že na disk se vejde $40 \times 9 \times 0.5 = 180$ kB údajů.

Počítače Amstrad CPC přímo umožňují dva typy formátů: Formát DATA je z kapacitních důvodů výhodný, protože pro potřeby operačního systému rezervuje minimální prostor na disku a umožňuje uložit maximum uživatelských údajů. V prvních 4 sektorech je uložen adresář, který zabírá $4 \times 0.5 = 2$ kB prostoru, ve zbylých 178 kB mohou být uloženy uživatelské programy či data. Aby operační systém poznal, o který formát se jedná, jsou při formátování jednotlivé sektory na disku označeny. Pro formát DATA jsou sektory označeny čísly C1H až C9H (příponou H označují hexadecimální tvar čísla).

Pokud počítáme v operačním systému CP/M 2.2 nebo chceme spustit operační systém CP/M plus, musíme užít disk, který je formátován jako SYSTEM. Při tomto formátu jsou celé první dvě stopy rezervovány pro potřebu operačního systému CP/M, první 4 sektory ve třetí stopě jsou rezervovány pro adresář. Tudíž $2 \times 9 \times 0.5 + 4 \times 0.5 = 11$ kB je rezervováno, zbyvajících 169 kB je pro potřeby uživatele. Ve formátu SYSTEM jsou sektory označeny 41H až 49H. Můžeme se setkat i s názvem VENDOR. V tomto formátu jsou stopy a sektory rozděleny a označeny stejně, jako ve formátu SYSTEM, pouze první dvě stopy jsou volné a před použitím je třeba do nich dodatečně zaznamenat systémové údaje.

Operační systémy Amdos a CP/M 2.2 mohou pracovat i s disky, formátovanými jako IBM. Nedoporučuji však používat tento formát, protože dělí stopy pouze na 8 sektorů, 1 stopu vyhrazuje pro systémové účely, 4 sektory vyhrazuje pro adresář. Lze tedy s tímto systémem zaznamenat pouze 154 kB užitečných údajů. Sektory jsou číslovány 1 až 8.

Adresář

Adresář umožňuje operačnímu systému orientaci na disku. Každému souboru je vyhrazena "adresa", která zabírá prostor 32 bajtů. Jednotlivé bajty mají následovný význam:

0. Ukazuje číslo uživatele (USER), který zapsal soubor. Může nabývat hodnot 0 až 15, při obvyklém užívání je nula.

Pokud je soubor vymazán instrukcí ERA, je v prvním bajtu hodnota ESH. Tento údaj říká operačnímu systému, že v případě potřeby může přemazat adresu novými hodnotami.

1-8. Jméno souboru, případně doplněné na konci mezerami.

9-11. Přípona jména souboru (na př. BAS, COM a p.).

12. Pokud je soubor delší než 16 kB, je mu vyhrazeno více adres. Bajt 12 je pořadové číslo adresy.

13-14. Amsdos nevyužívá.

15. Udává délku té části souboru, na kterou ukazuje příslušná adresa. Jako jednotka délky je užíván tzv. logický sektor, který je tvořen 128 bajty informace. Tato jednotka pochází z doby počítačového středověku, kdy z technických důvodů bylo možno na jeden sektor disku uložit pouze 128 bajtů.

16-31. Obsahuje pořadová čísla bloků velikosti 1kB (t.j. 1024 bajtů). Pokud je soubor kratší než 16 kB, jsou neobsazené bloky vyznačeny nulami. Při výpisu adresáře instrukcí CAT se počítá délka souboru jako počet obsazených bloků, nikoliv pomocí údaje v bajtu 15. Pokud je soubor delší než 16 kB, nevejdou se pořadová čísla bloků do jedné adresy a je přidána adresa další, která se od předchozí liší v bajtu 12 a přirozeně v číslech na bajtech 16-31. Případně další adresy jsou přidávány podle potřeby. Bloky jsou počítány tak, že bloky č. 0 a 1 odpovídají adresáři, blok 2 a další patří prostoru na disku za adresářem. Vztah mezi pořadovým číslem bloku BL, stopou ST a sektorem SEK je dán následujícími vzorci:

a) formát SYSTEM $ST = INT((BL*2+18)/9)$
 $SEK = (BL*2) \text{ MOD } 9 + 41H$

b) formát DATA $ST = INT((BL*2)/9)$
 $SEK = (BL*2) \text{ MOD } 9 + C1H$

MOD značí zbytek po celočíselném dělení stejně jako v Basic-u.

V adrese jsou zakódovány ještě dva údaje do místa přípony jména souboru. Je přitom využito toho, že ASCII kód přípony zabírá pouze 7 spodních bitů v každém bajtu (bity 0 až 6) a poslední bit je volný. Pokud bit 7 v bajtu 9 je roven 1, jedná se o tzv. Read Only soubor (znamená pouze čti, často označován R/O), který nemůže být přepsán či smazán. Pokud bit 7 v bajtu 10 je roven 1, jedná se o SYS soubor

(systémový), který se nevypíše po instrukcích CAT a DIR, ale který může používat kterýkoliv znaly uživatel pro výpočty, opravy či jiné manipulace.

Upozorňuji, že při použití operačního systému CP/M plus může být struktura adresáře značně komplikovanější, než zde bylo uvedeno. K tomu dochází tehdy, když užíváme možnost systému CP/M plus chránit soubory hesly nebo označit soubory časovými značkami.

Záhlaví

Soubory, které obsahují programy psané v BASIC-u či ve strojovém kódu (přípony BAS či BIN) obsahují ještě záhlaví, které je obdobné záhlaví souborů na magnetofonových kasetách. Záhlaví je značně neekonomické, zabírá prvních 128 bajtů souboru a obsahuje pouze tyto údaje (obdobně jako u adresáře uvádím pořadová čísla bajtů):

- 0. Číslo souboru.
- 1-8. Jméno souboru.
- 9-11. Přípona jména souboru.
- 18. Typ souboru.
- 21-22. Adresa začátku programu.
- 24-25. Délka souboru.
- 26-27. Adresa, od které program startuje výpočet (může být odlišná od začátku programu).
- 64-65. Délka souboru (totožné s 24-25.).
- 67-68. Kontrolní součet bajtů 0 až 66. Pomocí této hodnoty odlišuje operační systém soubory se záhlavím a bez záhlaví. Pokud součet bajtů 0 až 66 je roven hodnotě uložené na 67-68, jde o soubor s hlavičkou, pokud ne, jde o soubor bez hlavičky (a když u souboru bez hlavičky je čistou náhodou suma bajtů rovna hodnotě na 67-68, máte prostě smůlu a budou se dít věci veliké).

Typ souboru je možno rozlišit podle obsahu jednotlivých bitů v bajtu 18., jak je demonstrováno v následující tabulce:

Typ:	Basic	Binary	ASCII
Bit 1:	0	1	1
Bit 2:	0	0	1

Když bit 0 je roven 1, jedná se o chráněný (protected) soubor, v opačném případě jde o nechráněný soubor. Soubor ASCII má záhlaví jen tehdy, když je chráněný. Programy s příponou CDM, které spolupracují s operačním systémem CP/M, nemají záhlaví nikdy.

RnDr. Svatopluk Kríz

~~~~~

### CP/M PLUS NA DATOVÝCH DISKETÁCH

Počítače Amstrad CPC používají dva hlavní způsoby formátování disket, datový a systémový formát. Systémový formát má dvě rezervované stopy a proto je jeho kapacita využitelná uživatelem o 9 kbyte menší. U systémového disku CP/M Plus je v rezervovaných stopách uložen pouze v prvním sektoru krátký zaváděcí program, jinak jsou rezervované stopy zcela nevyužité. Vlastní systém CP/M Plus je uložen na disku jako soubor C10CPM3.EMS. Prozkoumáme-li podrobněji zaváděcí program pro CP/M Plus, zjistíme že vlastně nedělá nic jiného než že vyhledá na disku soubor s typem .EMS a naládí ho od adresy 0C00h a spustí ho od této adresy. To znamená, že soubor C10CPM3.EMS bychom mohli spouštět jako každý jiný binární soubor. K tomu ho ale musíme vybavit headerem ze kterého AMSDDS pozná kam má soubor naladovat a od které adresy spustit. Tuto jinak komplikovanou práci za nás téměř automaticky vykoná monitor z PYRADEVu. Postup je následující: Naládujeme monitor a relokujeme ho tak aby jeho startovací adresa byla 7100h nebo vyšší, přičemž zvolíme standardní verzi. Pak zvolíme funkci L - čtení z disku, vložíme systémovou disketu a zadáme jméno souboru C10CPM3.EMS. Protože tento soubor nemá header, monitor nám nabídne ládovací adresu C000h.

```
LOAD: C000
EXEC: 0000
LEN: 0000
```

Musíme tedy přepsat adresu C000 uvedenou za LOAD na 0C00 a soubor se načte do paměti. Pak vložíme pracovní disk zformátovaný jako datový, zvolíme funkci W - zápis na disk a

zadáme následující:

From Address (HEX): 0C00  
To Address (HEX): 6BFF  
LOAD Address (HEX): 0C00  
EXEC Address (HEX): 0C00  
Filename :CPM.BIN

Na disk se zapiše soubor CPM.BIN a bude tedy možné z tohoto disku po použití příkazu RUN"CPM spustit CP/M Plus. Vyhody jsou zřejmé:

1. Získáme na disku 9 kbyte navíc.
2. Nemusíme již rozlišovat mezi datovými a systémovými disky a můžeme používat jen datové.
3. Pouhým přenesením souboru CPM.BIN na jakýkoliv disk máme možnost spouštět CP/M Plus z tohoto disku.
4. Zavedení CP/M je o poznání rychlejší.

Ing. František Langmaier

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

---

Sord+Amstrad, technický zpravodaj pro mikroelektroniku a výpočetní techniku. Vydává 602 ZO Svazarmu pro potřeby vlastního aktivu, zodpovědný redaktor ing. Petr Čihula (Sord), ing. Richard Kaucký (Amstrad). Adrese redakce: 602. ZO Svazarmu, Wintrova 8, Praha 6, 160 41. Telefon 32-85-63. Povoleno ÚVTEI pod evidenčním číslem 87 086. Cena: 7.50 Kčs dle ČCÚ č. 1030/202/86 Náklad 500 výtisků Praha, březen 1988

---