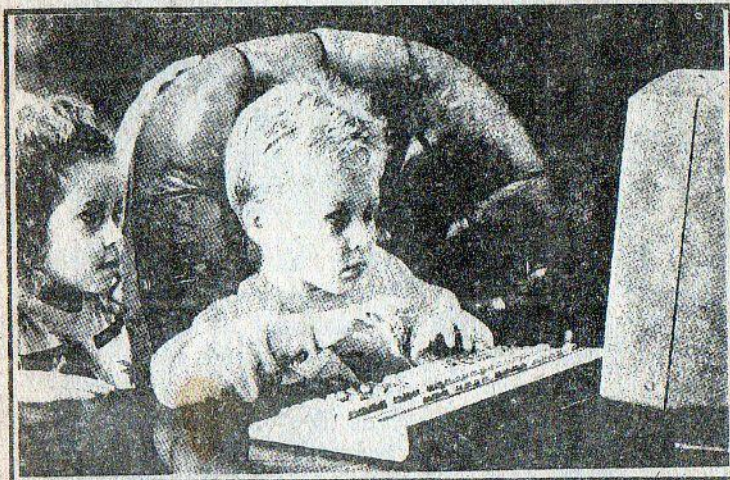


amatérský programátor

#1-93

Jediný soukromý časopis pro amatérské programátory
na počítačích ZX SPECTRUM / DIDAKTİK A KOMPATIBILNÍCH.



- * Změna barev na obrazovce * COPY pro Robotrona
- * Kurz programování strojového jazyka Psát česky
- * BroSoft pro SPD * BETA BASIC 03 (pokračování)
- * a ještě další a další, včetně „AP PLUS„ (PROGRAMY)

na úvod

pohřeb se zatím odkládá

Na závěr mého úvodníku posledního čísla roku 1992 jsem si kladl otázku „Jaký asi budu muset napsat pro lednové číslo?“.

Dnes tedy opět sedím u svého ZKS* a píšu úvodník do tohoto 1. čísla roku 1993. Co napsat? Přiznám se, že vlastně stále ještě nevím. Již dva měsíce jsem na AP ani nepomyslel, neboť jsem vydával celkem 6 katalogů pro jednu reklamní agenturu abych tak získal nějaké peníze na dotaci AP. Díky této zakázce jsem mohl zaplatit ofsetové stroje na kterých se nyní AP tiskne a nakoupit papír. Proto bylo možné vydat toto číslo.

Jak to ale bude s dalším číslem, prozatím nevím. Rád bych se touto cestou na Vás obrátil o pomoc. Pokuste se ve svém okolí přesvědčit nějakou firmu či soukromníka, aby si v AP nechal zveřejnit svou reklamu. Takto získané finance totiž AP mohou dosti pomoci. Ceny za inzerci jsou zveřejněny na jiné stránce AP. Pokud se reklama bude opakovat alespoň 3x zaplatí inzert o 40% méně.

Rád bych také poděkoval všem, kteří reagovali na poslední úvodník tak, jako třeba pan Špirek, který mimo jiné napsal: „Jste asi jediný kdo nechce ze všeho a všude jen vydělávat, tak jako ti ostatní dravci, kterým vlastně ani nejde o to co dělají, ale za kolik to dělají“.

jak bude ápičko vycházet letos

NA
ÚVOD
AP

Abych ušetřil co možná nejvíce, rozhodl jsem se přistoupit na návrh jednoho čtenáře, vydávat AP vždy jednou za dva měsíce v rozsahu dvojčísla. Histo 5,- Kč za poštovné každý měsíc na AP o 24 stranách, dům nyní 7,- Kč každé dva měsíce za AP v rozsahu 44 stran. Proto tedy najdete v každém čísle „AP“ Cena časopisu se tedy zvýší na 15,- Kč a spolu s poplatky na poštovné (42,- Kč) se dostaneme do částky 132,- Kč. Nebude tedy třeba nic doplácet (pokud nám zase pošta nezdraží).
Petr Černý

ap listárna

využití tiskárny
d 100 m

V poslední době se nejvíce dotazujete na obsluhu, využití a zapojení tiskárny D100M, dovážené k nám z Polska. Přesto, že její prodejce k ní dodává stručný slovenský překlad, není mnohým uživatelům obsluha tiskárny zcela jasná. Rád bych zde ještě zdůraznil, že je nutno rozlišovat označení této tiskárny mezi D100M, D100S a pouhé D100.

Tiskárny D100 se k nám dovážely hned po tzv. „revoluci“. V současné době však lze již zakoupit pouze D100M. Mezi oběma je však rozdíl. Pokud vlastníte D100M a chcete k ní použít program pro D100, pak Vám asi tisknout nebude. Naproti tomu má i D100M své záludnosti. Přesto, že se o ní říká, že je kompatibilní se standartem EPSON, není to tak docela pravda. Částečně totiž ignoruje normu IBM a EPSON. Proto se může u některých programů stát, že po volbě

nerozjede a někdy dokonce i tzv. zamrzne (projeví se to tak, že žádné tlačítko na tiskárně nereaguje). Zde jen stačí nechat tisk jako by proběhnout, pak ji vypnout a znovu zapnout a zvolit znovu tisk. U takového programu již příště tiskárnu zapneme teprve až ve chvíli, kdy se rozhodneme, že budeme požadovat tisk.

Pokud tisková rutina programu používá numerické kódy (tiskne ze znakové sady v tiskárně) rutinou „cenz2“, pak by mělo vše probíhat bez problémů. Jestliže ale program tiskne v grafice (např. ART-STUDIO, DESKTOP, TEXT MACHINE aj.) dostáváme se do problémů, neboť žádný z nich není schopen na D100M tisk provést. Je zde totiž rozdíl proti EPSON v kódování grafiky a řádkování. Např. u programu DESKTOP bylo nutno doplnit instrukci LD A, 27 a LD A, 51 ještě další instrukcí LD A, 24 pro správné odřádkování. O úpravě programu Text Machine jsem psal již v 11. čísle roku 1992.

Úpravu tiskové rutiny u ART Studia jsem zatím ještě nestihl udělat. V případě DESKTOPu, věřím, že tuto úpravu provede fa. Proxima a umožní tak i uživatelům této jinak vynikající tiskárny jejich program bez potíží

DOPISY

ČTENÁŘO

AP

používat.

U některých programů je i volba pro jednosměrný nebo obousměrný tisk. U D100M je však marné tuto volbu využívat, neboť se to nastavuje na přepínači DIL K2/7.

-----DIL K2-----

1 2 3 4 5 6 7 8



7-zapnut
jednosměrný
tisk

1 2 3 4 5 6 7 8



7-zapnut
obousměrný
tisk

Protože jsem si vědom, že manuál k tiskárně je v polském jazyce, a ne každý umí polsky, uvádím na jiné straně alespoň částečný popis některých uživatelem zřejmě nejvíce využívaných instrukcí.

**public domain v
tomto roce.**

Zvýšené poštovné se projeví letos také na zasílání programů SPD. Opět se i zde budu snažit snížit náklady na SPD na možné minimum. Protože peaní do 500g se nyní posílá za 7,-Kč + 5,-Kč za doporučenou zásilkou, činí to celkem

12,-Kč. Programy SPD budou v letošním roce proto koncipovány tak, aby se Vám to vyplatilo. Pro majitele D40 budu sestavovat programy tak aby byla pokud možno využita plná kapacita diskety. Tedy již žádné zasílání jednoho jediného programu, ale vždy celé plné diskety.

Pro majitele jednotky D80 a počítačů KOMPACT, mám bohužel smutnou správu: **na disky 3,5" programy SPD poskytovat nemohu, neboť ji nevlastním.** Zatím nemám ani dost financí abych si ji mohl zakoupit. Požádal jsem sice výrobce DIDACTIC a.s. o darování Kompaktu (nebo D80) pro redakční účely, ale dostal jsem doporučení abych si ji zakoupil.

Cena za SPD bude nyní počítána za jednu disketu ve výši 12,-Kč + 5,-Kč na poštovné a k tomu 8,-Kč na amortizaci, **** celkem tedy 20,-Kč. ***

Služba SPD bude opět obnovena od 2. čísla AP (březen 93).

NEPŘEHLEDNÍ

Hvězdička před číslem na Tvé adrese neoznačuje kolik čísel ještě máš předplaceno, ale do kterého čísla máš uhrazeno předplatné!

kurz programování strojového jazyka

pro
AP
píše
PAVEL MACEK

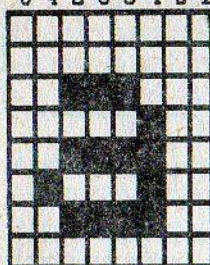
PSÁT
ČESKY

Všichni víme čím se naše písmo liší od jiných. Používá totiž háčky, čárky a kroužek nad písmeny.

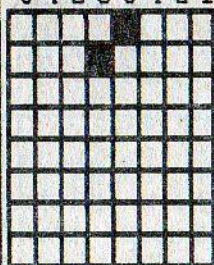
psát česky

Jeden ze způsobů jak psát opravdu česky na počítači je, mít zvlášť vytvořené znaky s čárkami a háčky a „d“. Druhý, který si teď předvedeme, spočívá ve vytvoření čárky, háčku a kroužku, které pak zobrazujeme k k patřičným písmenům.

1
2 6 3 1
8 4 2 6 8 4 2 1



1
2 6 3 1
8 4 2 6 8 4 2 1



Na obrázku je v rastru 8 * 8 bodů znázorněn znak „a“, a vedle něho znak „čárka nahře“. Nyní je možné zobrazit nejprve znak „a“, a potom s pomocí OR zobrazit přimícháním znak „čárka“, na tutéž pozici a tím zobrazíme „á“.

Můžete si zkusit složit program, který by dokázal udělat právě to, co jsme si řekli. Do registrového páru HL ukládejte hodnotu 18442, aby byl znak zobrazován bližší středu obrazovky kvůli přehlednosti. Nejlépe, když potom program budete krokovat nebo trasovat. Pomalejší běh programu umožní lepší přehled o průběhu programu a snadno tak také zpozorujete případnou chybu.

co všechno lze
provádět
se znakem

Kolumbovo vejce prý stálo na špičce, ale zkuste postavit na špičku třeba náš znak „A“. Co je k tomu prakticky potřeba? Stačí obrátit čísla vyjadřující tvar znaku a nebo ještě lépe, číst je obráceně. Zkuste si následující program:

NASTV1	LD HL, 18442	;příprav adresu do obrazovky
	LD DE, ZNAK	;příprav tvar znaku
ZRCX	PUSH HL	;ulož adresu obrazovky
	LD HL, 7	;příprav číslo sedm
	ADD HL, DE	;přičti k adrese tvar znaku
	EX DE, HL	;spočtenou adresu dej do DE
	POP HL	;obnov adresu do obrazovky
	LD B, 8	;pro osm bajtů tvaru
PSS2	LD A, (DE)	;vezmi bajt tvaru
	LD (HL), A	;dej ho do obrazovky
	DEC DE	;sniž na další bajt tvaru
	INC H	;zvyš na další linku
	DJNZ PSS2	;opakuj podle reg. B
	RET	;návrat

Na počátku programu přičteme k adrese čísel tvaru sedm, čímž se ocitáme na posledním bajtu tvaru. Potom během zobrazování postupujeme obráceně až k prvnímu bajtu ale tvar znaku přitom zobrazujeme do obrazovky jako vždy shora dolů. Tak se znak objeví na obrazovce špičkou dolů „V“.

Všimněte si návěští ZRCX, které vyjadřuje funkci tohoto programu. Jedná se totiž o zrcadlo podle vodorovné osy X. Samozřejmě, že je možné zobrazit znak také zrcadlově podle svislé osy Y. O tom bychom se však na znaku „A“ velmi těžko přesvědčili. Vytvořme si proto k tomuto účelu znak „E“:

ZNK2	DEFB 0, 124, 64, 112
	DEFB 64, 64, 124, 0

Potom je možné vyzkoušet následující program:

```

NASTV2 LD HL, 18442 ;příprav adresu do obrazovky
        LD DE, ZNK2 ;příprav adresu tvaru znaku

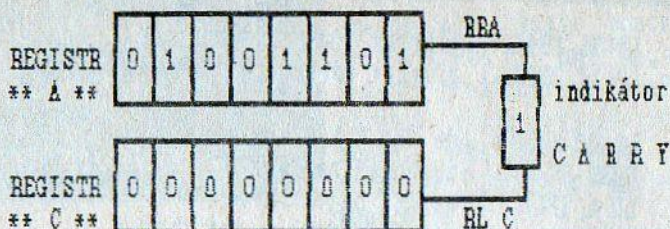
ZRCY LD BC, #0808 ;do reg. B i C dej číslo 8
PSS3 PUSH BC ;uschovej registr BC
      LD A, (DE) ;vezmi bajt tvaru
PSS4 RRA ;rotuj doprava v registru A
      RL C ;odtud rotuj doleva do reg. C
      DJNZ PSS4 ;opakuj podle registru B
      LD (HL), C ;výsledný bajt ulož do obrazovky
      INC DE ;další bajt tvaru
      INC H ;další linka v obrazovce
      POP BC ;obnov registr BC
      DEC C ;sniž registr C
      JR NZ, PSS3 ;opakuj pokud reg. C není nula
      RET ;návrát
  
```

Program pro zrcadlo v ose Y není také nijak složitý. Přímo u návěští ZRCY se tu setkáváme s instrukcí, u níž bylo použito hexadecimálního vyjádření dvoubajtového čísla. Kdo si prostudoval práci s assemblerem, s nímž pracuje, už se setkal s takovým vyjádřením čísel. O hexadecimálních číslech a jejich užití jsme si to podstatné už také řekli. Zbývá jen dodat, že tímto způsobem se hexadecimální čísla píšou do assembleru aby je odlišil od dekadických. Zcela nově se tu setkáváme s instrukcemi rotace, o které si musíme něco povědět.

rotace

znamenná, že rotují jednotlivé bity v bajtu. Děje se tak přes indikátor C (CARRY). Pokud rotujeme doprava, přesune se nejnížší bit do indikátoru CARRY a při další rotaci doprava se přesune z indikátoru CARRY na místo bitu nejvyššího. Na následujícím obrázku máme přímo příklad z programu ZRCY kde rotujeme doprava nejprve registr A a po té doleva registr C. V programu se to děje ale po všech osm bitů každého bajtu tvaru. Bajt se tak přeháze, že nejnížší bit je nejvyšší a tak postupně až k bitu nejvyššímu. Tím je

také zrcadlově obrácen tvar znaku ve svislé ose.



Rotovat je možné doleva i doprava téměř všemi registry. Symbolika je jednoduchá. RR znamená *rotate right* (rotuj doprava) a RL znamená *rotate left* (rotuj doleva). S využitím rotací dokážeme uplatnit další zajímavý efekt. Představte si nějakého siláka jak uhodí do našeho znaku „A“, a ten se obrátí na bok. Takovým silákem je další program:

```

MASTV3 LD HL,18442 ;připrav adresu do obrazovky
LD DE,15616+264 ;připrav adresu tvaru znaku „A„
OBRT PUSH HL ;ulož adresu obrazovky
EX DE,HL ;adresu tvaru do HL
LD DE,ZASOB ;do reg.páru DE adresu zásobníku
LD BC,8 ;délku osm bajtů
LDIR ;přenes do zásobníku
POP HL ;obnov adresu do obrazovky
LD BC,#0808 ;do reg.B i C dej číslo 8
OBT1 PUSH BC ;uschovej registr BC
LD DE,ZASOB ;připrav adresu zásobníku
OBT2 LD A,(DE) ;bajt tvaru do registru A
RLA ;a tam ho rotuj doleva
LD (DE),A ;zrněnu zapiš do zásobníku
RR C ;bit rotuj do reg.C doprava
INC DE ;další bajt tvaru v zásobníku
DJNZ OBT2 ;opakuji podle registru B
LD (HL),C ;složený bajt uloží do obrazovky
INC H ;další linka v obrazovce
POP BC ;obnov registr BC
DEC C ;sniž registr C
JR NZ,OBT1 ;pokud není nula tak opakuji
RET ;návrat
ZASOB DEF 8 ;definováno místo pro zásobník

```


Jak to program dělá? Nejprve si musíme určit místo, kam si připravíme osm bajtů, představujících tvar znaku. Bajty budeme na obrazovce zobrazovat jako vždy shora dolů. Budeme však pro zobrazení potřebovat bajty, které tvoří jednotlivé **mikrosloupce** tvaru znaku. Potom když začneme třeba levým mikrosloupcem, bude znak ležet na pravém boku. Jak si však mikrosloupce přeskládat do bajtů pro zobrazení? Samozřejmě opět s využitím instrukcí rotace. Postupně pro všech osm bajtů tvaru odrotujeme jeden bit a hned tento posun zapíšeme do zásobníku. V registru C se nám složí bajt mikrosloupece, který ukládáme do obrazovky. Při delší rotaci se skládá další mikrosloupec a další až jsou všechny. Jestliže chceme, aby se znak pokládal na opačný bok, stačí místo instrukce BLA dát instrukci RRA a na místo instrukce RR C instrukci BL C. Doporučuji tuto variantu rovněž vyzkoušet.



abeceda v počítači

Možná si někdo z vás všimnul, že v programu obratu znaku na bok je při nastavení registrů DE

na adresu tvaru znaku použito adresy 15616. Takový postřeh je nutno pochválit. Nelze již před vámi tajit, že tvary všech písmen (znaků), které počítač používá jsou zapsány v pevné paměti ROM. Tabulku těchto znaků je možné nalézt v příručkách k počítači tam, kde jsme hledali instrukce strojového kódu ukryté pod čísly. Každý znak má také svůj kód. Vidíme v tabulce, že tabulka znaků začíná mezerou, což je kód 32. Mezeru tvoří osm čísel nula a tyto se nacházejí právě na adrese 15616. Pak následuje dalších osm čísel, vyjadřujících tvar dalšího znaku tabulky. Jsme tedy podle kódu znaku schopni vypočítat adresu tvaru znaku?

```
5 INPUT „Kód znaku: „;kod
10 LET znak=kod-32
20 LET adrzn=15616+(znak*8)
30 PRINT adrzn
```

Příkladem nám může být krátký program v Basicu. Nejprve odečteme rozdíl od počátku k prvnímu znaku. Ten je 32. Tak získáme přehled kolikátý je to znak. Každý znak má osm bajtů na vyjádření svého tvaru, musíme ho tedy znásobit osmkrát. Pak

dění svého tvaru, musíme ho tedy znásobit osmkrát. Pak

už stačí výsledek přičíst k adrese znaků. V assembleru to může vypadat asi takto:

```

ASCR SUB 32      ;odečti 32
      LD H, 0    ;nuluj registr H
      LD L, A    ;kolikátý znak dej do L
      ADD HL, HL ;sčítej reg. páry HL (násobení)
      ADD HL, HL
      ADD HL, HL
      LD DE, 15616 ;do reg. páru DE připrav adresu tvarů
      ADD HL, DE  ;a přičti ji
      EX DE, HL  ;výsledek do reg. páru DE
      RET       ;návrat
  
```

Násobení osmi zde probíhá podobně jako násobení třiceti dvěma, které jsme si detailně popsali. Opakovací smyčka se pro tři opakování nevyplatí používat. Takhle je totiž program kratší než s použitím smyčky DJNZ. Smyčka se vyplácí pouze při násobení větším číslem než 32.

*** Pokud do tohoto programu vstupujeme s kódem ABECEDA znaku v registru A, dostaneme na jeho výstupu ***** v registrovém páru DE adresu tvaru tohoto **** znaku. Tím jsme získali nadvládu nad abecedou v našem počítači. Nutno ještě podotknout, že u novější verze počítače DIDAKTIK GAMA a u DIDAKTIK M jsou znaky, které počítač používá, o něco tučnější než je tomu u starší verze DG a ZX Spectra. Pokud někomu více vyhovují tučnější znaky, stačí aby použil krátký speciální zobrazovací program. Tučné znaky už neradno **ztučňovat**. Proto je následující program spíše pro ty, jejichž počítač používá slabších znaků:

```


MASTV4 LD HL, 18432 ;adresa do obrazovky
      LD DE, 15616+520 ;adresa tvaru znaku „a“
TUC1   LD B, 8      ;pro osm bajtů tvaru
TUC2   LD A, (DE)   ;vezmi bajt tvaru
      LD C, A       ;vlož do registru C
      RR C          ;rotuj reg. C doprava
  
```

pokračování dále

dokončení

OR C	; smíchej s pđv. tvarem v reg. A
LD (HL), A	; vlož bajt do obrazovky
INC DE	; další bajt tvaru
INC H	; další linka na obrazovce
DJNZ TUC2	; opakuj podle registru B
RET	; návrat

Ve svých programech můžete samozřejmě používat různé znakové sady. Většina her má své vlastní znakové sady. Používají při psaní textů na obrazovku znaků, které počítač nemá ve své pevné paměti ROM. Taková vlastní znaková sada, představující tvary znaků, které si sami vytvoříte v nějakém grafickém editoru (např. ART-STUDIO, ARTIST ap.), se ukládá jako sada bajtů (čísel) do paměti RAM a představuje část vašeho programu. Kódy znaků se zachovávají, mění se jejich tvar. Znaků je v tabulce 96, délka znakové sady bude představovat v paměti RAM 96*8=768 bajtů. Jak budeme hledat adresy tvarů znaků ve vlastní znakové sadě? Stejně jako jsme to dělali u znaků z paměti ROM. Stačí jen v programu ASCR vyměnit v instrukci LD DE, 15616 adresu znaků v ROM za adresu naší znakové sady.

V tabulce znaků jste si mohli všimnout, že od kódu 144 začíná uživatelská grafika - UDG -  ABECEDA *****

mohou to být různé obrazce (šipky, stromy, figurky, ap.), které je možné zobrazit v rastru 8 * 8 bodů. Pokud si vytvoříme vlastní znakovou sadu, můžeme od kódu 97 vytvořit uživatelskou grafiku až do kódu 255. Znaková sada potom může mít délku 224*8=1792 bajtů. Program ASCR se v ní bude umět orientovat jenom musíme vědět, pod kterým kódem máme jakou grafiku.

v dalším pokračování si povíme něco o grafických oknech a zobrazení textů.

brosoft

...posílám Vám soubor programů které jsem vytvořil a nebo se ke mně dostaly složitými cestami pirátských kopíí.

Programy jsou určeny pro SPD vracet

a protože některé jsou dosti složité, uvádím jejich popis.

air lines

Tento program je mým prvním pokusem o hru. Jde o manažer letecké společnosti.

Jako začínající podnikatel máte vydělat 5,000.000,- \$ (\$-čechodolar). Hra se ovládá joystickem typu kempston a tlačítkem „ENTER„ se potvrzuje nabídka. Program je zkompilován HiBasicem a proto vám vřele **nedoporučuji** s ním provádět žádné experimenty.

Funkce jednotlivých položek;

LETADLA -po potvrzení této nabídky se objeví podnabídka;

-NAKUP: zde si můžete koupit letadlo

-OPRAVY: opravy letadel nutno provádět po každém

letu, jinak letadlo nelze použít až do konce měsíce

-KONEC: návrat zpět do předchozí nabídky,

BANKA - opět se objeví další podnabídka;

-PUJCIT: půjčená částka může být max. 1,500.000,-\$

-VRATIT: vrácení dané částky

-SPLATKA: určení výše částky, kterou budete každý měsíc

-KONEC: návrat

BEKLAMA - reklama je možná v televizi, rozhlasu nebo v novinách. Množství reklamy ovlivňuje velikost částky, která se přičítá na váš vrub.

POSADKY - jako obvykle lze zaměstnance přijímat, propouštět nebo jim měnit plat

CENY - ceny letenek v turistické a první třídě

TRASA - vyberete si kam a čím chcete letět

INFO - tady dostanete některé důležité informace

KONEC - toto bude asi vaše nejoblíbenější nabídka.

Ještě pár poznámek k herní taktice; doporučuji aby jste všechny údaje udržovali přibližně na 50%. Protože být podnikatelem v zemi zaslíbené není žádný med, mají-li

zaměstnanci dovoleno létat pouze jeden let měsíčně.

Podobné opatření je také u letadel. Pokud letadlo neopravíte, musíte si počkat do konce měsíce.

K absolvování letu potřebujete následující komponenty:

pilot.....1 kus

letuška.....1 KUS!

letadlo.....1 kus

peníze.....minimálně

na zaplacení paliva(1\$=1litr)

DÍLNA
ČTENÁŘO
AP

Při výběru letadla je důležitý počet. Oprava letadla stojí 1/4 jeho ceny. Pokud piloti nebo letušky mají malý plat, začnou dávat výpověď, což se rovná 2-měsíční plat. Jestliže má letadlo více než 1000 letových hodin, spadne. Přes mou snahu hru udělat co nejvěrohodněji, jsem se nevyhmul chybám a proto vás žádám, aby jste je omluvili.

vého souboru.
SEZNAM - jsou v něm uloženy dvě „adresy,“ Hru znovu spustíte tlačítkem BREAK. Spojení přerušíte CAPS SHIFT. Po dokončení se nedovíte žádný tajný kód, takže si ho nemůžete zapisovat.

DÍLNA
ČTENÁŘ
AP

shades

computer world

Toto je má druhá nelogická hra, naprogramovaná ve stylu hry PODRAZ. Jako předchozí se i tato hra ovládá z menu.

Protože nabídka je asi ze 145.4645575456353651532563729% pro normálního smrtelníka naprosto nepochopitelná, opět celé menu popíšu:

POSLAT - tohoto příkazu použijete, chcete-li něco někam poslat

SPOJENÍ - můžete navázat spojení s dalšími uživateli Síť

VYMAZAT - doporučuji tento příkaz vyzkoušet hned po nahrání hry (soubor zmizí v nenávratnu)

PROHLÍŽET - tuto nabídku volte chcete-li zjistit obsah textů

Tento program již nepochází z mé dílny, a proto se autorovi (crackerovi, který ho upravil na BT100) již předem omlouvám. Program umožňuje tisk obrázků na „BT100“, ve 3 formátech podle ukázky na další straně. Protože nabídka je v angličtině, stručně popíšu co k čemu slouží:

0-SHADE upraví obrázek do žádané podoby,

1-BLACK/WHITE černobílý mód,
2-SHADED pseudobarevný mód,
3-ISOLATE tvorba vlastních „barev,“ (rastrů).

4-SHOW SCREEN ukáže obrázek a čeká na stisk klávesy, kde „M,“ je návrat na menu a „C,“ pro COPY na tiskárnu.

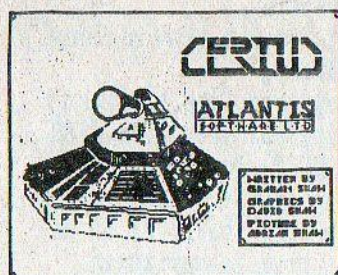
5-GET ORIGINAL nastaví originál obrázku.

6-LOAD UDG můžete si přihrát vlastní UDG plnicích vzorů.

7-SAVE SCREEN uloží obrázek na kazetu.

8-LOAD SCREEN nahraje obrázek bez hlavičky („na divoko,“)

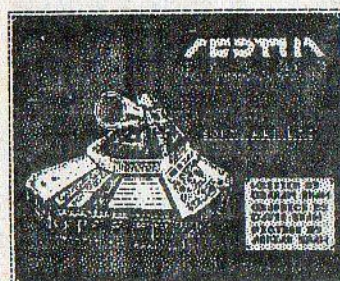
DÍLNA
ČTENÁŘ
AP



obr. 1: normálně



obr. 2: černobíle



obr. 3: pseudobarevně

9-PRINT u této položky moje experimenty nedosáhly valného účinku (kdo ví na co to vlastně je...)

Chcete-li vytisknout pseudobarevný obrázek, doporučuji použít následující postup:

- *8* nahrát obrázek
 - *5* ukaž originál
 - *2* pseudobarevný mód
 - *0* stínování
 - *4* ukaž výsledek
 - *C* kopie na tiskárnu
- dále buď „C“, další kopie, nebo „M“, návrat na menu.

Jestliže vám bude tiskárna rozhazovat, upravte program takto:

```
3001 IF INKEY$="C", OR INKEY$="c," THEN LPRINT CHR$ 27;"J,";
RANDOMIZE USR 64002: LPRINT
''''': FOR N=20 TO 55 STEP 3:
BEEP .1, N: NEXT N
```

Tisk se přepne na jednosměrný zpomalí se, ale bude přesnější.

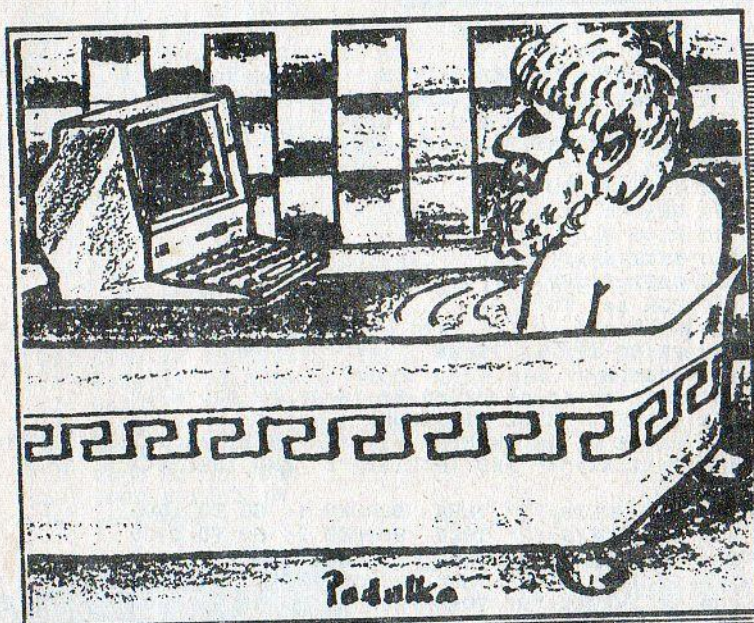


Jiří Brossman

Všechny tyto programy, které nám Jirka poslal, jsou připraveny do SPD na kazetu i dosketu. Zařazeno je ještě 19 znakových sad (fontů) a ukázkový program na jejich použití. Majitelé disketové jednotky D40 je najdou na disketě označené jako „SPD9-různé/užitkové“.

ap #1-93 plus
programy

- ♦ Uchovaný text
- ♦ CLS
- ♦ Schéma
- ♦ Ladička
- ♦ Stručný návod na D100M
- ♦ BETA BASIC 03



Pokud chcete také přispět svým vlastním programem do této programové přílohy, zašlete tištěný výpis, který však musí být čitelný, kontrastní (výrazný), prostě co nejčerněji vytištěn, aby mohl být použit pro tisk.

ladíčka

Široké využití Spectra je jistě dnes již každému známo. Pokud vlastníte Didaktik M, který má dosti silný zvuk

nebo máte jiné přídavné zařízení pro zesílení zvuku a současně hrajete na kytaru, pak Vám jistě přijde vhod tento program. Pro svou potřebu si ho napsal náš čtenář J. Kotvald z Jiříkova.

```

[38] 1 BORDER 0: PAPER 7: CLS : INK 9: BEEP .5,0
[DE] 2 PRINT '"ZADEJ POSUN TONU": INPUT "-10 +10 " ;la
d
[D7] 10 CLS : PRINT "LADICKA KYTARY"' "LADENI : ZAKLAD + " ;
lad: RESTORE
[OE] 20 FOR a=1 TO 8
[6A] 30 READ b
[4B] 40 PLOT 0,b: DRAW 250,0
[8D] 50 NEXT a
[3C] 100 DATA 94,99,107,115,123,124,131,132
[64] 110 FOR a=1 TO 6
[E8] 120 READ a$
[93] 130 PRINT AT a+4,29;a$
[E7] 140 NEXT a
[2A] 200 DATA "E-1","A-2","D-3","G-4","H-5","E-6"
[A4] 210 PRINT '"ZVOL SI STRUNU 1 - 6 "'
[0C] 230 PAUSE 0: IF INKEY$<>"1" AND INKEY$<>"2" AND INKEY$
<>"3" AND INKEY$<>"4" AND INKEY$<>"5" AND INKEY$<>"6" THEN
GO TO 230
[34] 240 IF INKEY$="1" THEN BORDER 1: GO TO 1000
[A7] 250 IF INKEY$="2" THEN BORDER 2: GO TO 1100
[1A] 260 IF INKEY$="3" THEN BORDER 3: GO TO 1200
[8D] 270 IF INKEY$="4" THEN BORDER 4: GO TO 1300
[FF] 280 IF INKEY$="5" THEN BORDER 5: GO TO 1400
[71] 290 IF INKEY$="6" THEN BORDER 6: GO TO 1500
[63] 605 RETURN
[56] 1000 GO SUB 5000
[38] 1010 LET ton=-20
[65] 1015 PRINT AT 18,6;"1"
[56] 1020 GO TO 6000
[BB] 1100 GO SUB 5000
[9C] 1110 LET ton=-15
[CB] 1115 PRINT AT 18,6;"2"
[BB] 1120 GO TO 6000
[1F] 1200 GO SUB 5000
[F6] 1210 LET ton=-10

```




```

[30] 1215 PRINT AT 18,6;"3"
[1F] 1220 GO TO 6000
[84] 1300 GO SUB 5000
[29] 1310 LET ton=-5
[96] 1315 PRINT AT 18,6;"4"
[84] 1320 GO TO 6000
[EB] 1400 GO SUB 5000
[85] 1410 LET ton=-1
[FB] 1415 PRINT AT 18,6;"5"
[EB] 1420 GO TO 6000
[4C] 1500 GO SUB 5000
[C1] 1510 LET ton=4
[60] 1515 PRINT AT 18,6;"6"
[4C] 1520 GO TO 6000
[ED] 5000 CLS : INPUT "ZADEJ DOBU TRVANI [sec] - ";sec
[08] 5010 PRINT "STISKEM TLACITKA "; FLASH 1;"SPACE"; FLASH
0;" ZAZNI""ZADANY TON "" FLASH 1;"Z"; FLASH 0;" SNIZI TO
O 1 OKTAVU"" FLASH 1;"X"; FLASH 0;" SNIZI TON O 2 OKTAVY""
FLASH 1;"C"; FLASH 0;" ZVYSI TON O 1 OKTAVU"
[2A] 5020 PRINT ""TLACITKEM "; FLASH 1;"K"; FLASH 0;" SE VR
ACIS NA START""TLACITKEM "; FLASH 1;"L"; FLASH 0;" VOLIS N
ABIDKU STRUN"
[CO] 5030 PRINT AT 18,0;"LADIS .STRUUN"
[D1] 5050 RETURN
[97] 6000 PAUSE /0: IF INKEY$<>"1" AND INKEY$<>"k" AND INKEY$
<>" " AND INKEY$<>"z" AND INKEY$<>"x" AND INKEY$<>"c" THEN
GO TO 6000
[A0] 6010> IF INKEY$=" " THEN BEEP sec,ton+l ad
[B7] 6020 IF INKEY$="z" THEN BEEP sec,ton-12+l ad
[CE] 6030 IF INKEY$="x" THEN BEEP sec,ton-24+l ad
[B2] 6040 IF INKEY$="c" THEN BEEP sec,ton+12+l ad
[14] 6050 IF INKEY$="k" THEN RUN
[90] 6055 IF INKEY$="1" THEN GO TO 10
[1A] 6060 GO TO 6000
[2B] 9000 BEEP 3,15
[7A] 9050 RUN

```

V Mostě existuje již delší
dobu **GAMA KLUB D40 / D80**
Pro zájemce uvádím adresu:

GAMA KLUB D-40
J. Dušek a spol.
Česká 625/377
434 01 M O S T



po uzávěrce

Z GAMA KLUBU v Mostě jsem obdržel disketu do SPD na které je několik textových editorů D-TEXT pro různé tiskárny - D100, PET, GMC, EPSON a BT100 - dále známý grafický editor ARTIST II pro K6304 s rozhraním centronics a ART STUDIO pro D40. To vše tedy máte možnost získat již od března za poplatek 20,-Kč v síti PUBLIC DOMAIN.

Další nový program je od Zdenka Gebauera **SCREEN LOADER 3** pro tzv. „vykrádání obrázků“, z her jak na kazetě tak i na disketě, a **TEST ZNALOSTÍ**, který z trochou nadsázky lze nazvat testem inteligence (Vaší nebo autora?)

Pro uživatele programu **TEXT MACHINE** připravuji diskovou verzi knihovny fontů a semigrafiky. Disketa bude stát celkem 60,-Kč včetně manuálu. Získáním této diskety se z tohoto programu stane skutečné **DESK TOP PUBLISHING!**

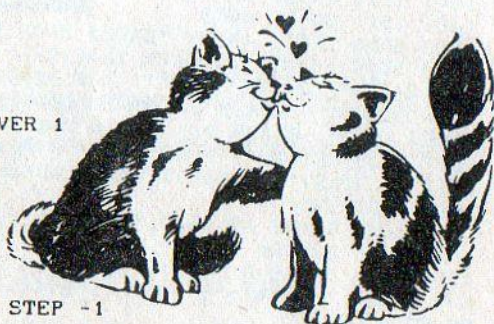
efektní els

David Wenzel nám přinesl do redakce další svou drobotinu, která tentokrát prová-

dí efektní „přemazání„ obrazovky.

Je to sice Basicový program, ale pokud by se zkompileval jistě by vznikla docela zajímavá rutina

```
[2B] 2 OVER 0
[3S] 3 BORDER 0
[3E] 4 INPUT ""
[FO] 5 FOR b=0 TO 1
[E1] 7 IF b=1 THEN OVER 1
[5F] 10 FOR a=0 TO 255
[4D] 15 PLOT 255,175
[64] 20 DRAW -a, -175
[CB] 21 PLOT 0,0
[OA] 22 DRAW a,175
[74] 25 NEXT a
[7A] 30 NEXT b
[6C] 35 FOR a=175 TO 0 STEP -1
[7E] 40 FOR b=0 TO 255
[86] 45 IF POINT (b,a) THEN PLOT b,a: OUT 254,b: OUT 254,
a
[8E] 50 NEXT b
[92] 55 NEXT a
[DO] 60 GO TO 40
```



schémy

Tento program jsem objevil v redakčním archivu a tak si již vůbec nejsem jist, zda nám ho poslal sám autor nebo někdo jiný.

Program upravil náš čtenář Julius Szarka mladší v roce 1986. Až se s programem do-

konale seznámíte a zapamatujete si jeho ovládání, pak získáte celkem dobrý program na kreslení různých schémat.

Pokud si program ještě doplníte tiskovou rutinou své tiskárny, uděláte na něm jistě mnoho užitečné práce.

[2C] Ø>REN *****
 ***** Program: Schemy Ing. Peter Ko
 zdon, 1986 ula 18/1986 *****

 ***** Opravil: Julius Szarka ml. *****
 28. IX. 1986

```
[F8] 5 POKE 23658,8: POKE 23609,10
[00] 10 CLEAR 44999: RESTORE
[93] 15 FOR I=60000 TO 60074: READ Q: POKE I,Q: NEXT I: FO
R I=60972 TO 60995: READ Q: POKE I,Q: NEXT I: FOR I=64009 TO
64067: READ Q: POKE I,Q: NEXT I: CLS
[10] 60 PLOT 0,0: DRAW 0,175: DRAW 255,0: DRAW 0,-175: DR
W -255,0
[AD] 65 LET X=255/2: LET Y=175/2: PLOT X,Y
[FB] 69 PRINT #0;" I=INSTRUKCIE"
[FB] 70 IF INKEY$<>" " THEN LET A$=INKEY$: GO TO 72
[97] 71 BORDER 4: LET A$=INKEY$: IF A$="" THEN GO TO 70
[83] 72 BORDER 7
[A7] 75 IF A$="8" THEN IF X<250 THEN LET X=X+1: PLOT X,Y
[50] 76 IF A$="5" THEN IF X>5 THEN LET X=X-1: PLOT X,Y
[55] 77 IF A$="6" THEN IF Y>5 THEN LET Y=Y-1: PLOT X,Y
[50] 78 IF A$="7" THEN IF Y<170 THEN LET Y=Y+1: PLOT X,Y
[36] 79 IF A$="1" THEN IF X>5 THEN LET X=X-1: PLOT X,Y:
PLOT OVER 1;X+1,Y
[3B] 80 IF A$="2" THEN IF Y>5 THEN LET Y=Y-1: PLOT X,Y:
PLOT OVER 1;X,Y+1
[45] 81 IF A$="3" THEN IF Y<170 THEN LET Y=Y+1: PLOT X,Y
: PLOT OVER 1;X,Y-1
[93] 82 IF A$="4" THEN IF X<250 THEN LET X=X+1: PLOT X,Y
: PLOT OVER 1;X-1,Y
[16] 83 IF A$="U" THEN RUN 50
[12] 84 IF A$="T" THEN GO SUB 1000
[0C] 85 IF A$="R" THEN GO SUB 1200
[99] 86 IF A$="C" THEN GO SUB 1400
[00] 87 IF A$="D" THEN GO SUB 1500
[6F] 88 IF A$="L" THEN GO SUB 1600
[0C] 89 IF A$="S" THEN GO SUB 1700
[0F] 90 IF A$="J" THEN GO SUB 1710
[FB] 91 IF A$="P" THEN GO SUB 1720
[EB] 92 IF A$="I" THEN GO TO 3000
[6F] 93 IF A$="Z" THEN COPY
[4B] 94 IF A$="0" THEN RANDOMIZE USR 64009
[02] 95 IF A$="B" THEN GO SUB 4000
[EE] 96 IF A$="G" THEN GO SUB 4020
[19] 100 GO TO 70
[72] 1000 IF X<9 OR X>246 OR Y<9 OR Y>166 THEN GO SUB 9999:
RETURN
[2C] 1001 CIRCLE X,Y,0
[6B] 1002 LET L=USR 60972
[BA] 1003 IF L=56 THEN GO TO 1010
[04] 1004 IF L=55 THEN GO TO 1020
[CF] 1005 IF L=54 THEN GO TO 1030
[09] 1006 IF L=53 THEN GO TO 1040
[AC] 1007 GO TO 1002
[4F] 1010 PLOT X,Y: DRAW 6,6: PLOT X,Y: DRAW 6,-6: PLOT X,Y:
```




```

DRAW 0,3: DRAW 0,-6
[09] 1012 LET X=X+6: LET Y=Y+6: PLOT X,Y
[FD] 1014 RETURN
[59] 1020 PLOT X,Y: DRAW 6,6: PLOT X,Y: DRAW -6,6: PLOT X,Y:
DRAW 3,0: DRAW -6,0
[13] 1022 LET X=X+6: LET Y=Y+6: PLOT X,Y
[08] 1024 RETURN
[00] 1030 PLOT X,Y: DRAW 6,-6: PLOT X,Y: DRAW -6,-6: PLOT X,
Y: DRAW 3,0: DRAW -6,0
[22] 1032 LET X=X-6: LET Y=Y-6: PLOT X,Y
[12] 1034 RETURN
[F3] 1040 PLOT X,Y: DRAW -6,-6: PLOT X,Y: DRAW -6,-6: PLOT X
,Y: DRAW 0,3: DRAW 0,-6
[64] 1042 LET X=X-6: LET Y=Y+6: PLOT X,Y: RETURN
[80] 1200 IF X<11 OR Y<11 OR X>244 OR Y>164 THEN GO SUB 999
9: RETURN
[33] 1201 LET L=USR 60972
[40] 1202 IF L=56 THEN GO TO 1210
[53] 1203 IF L=53 THEN GO TO 1220
[63] 1204 IF L=55 THEN GO TO 1230
[60] 1205 IF L=54 THEN GO TO 1240
[30] 1206 GO TO 1201
[45] 1210 DRAW 0,-2: DRAW 10,0: DRAW 0,4: DRAW -10,0: DRAW 0
,-2
[68] 1212 LET X=X+11: PLOT X,Y
[6] 1214 RETURN
[4F] 1220 DRAW 0,-2: DRAW -10,0: DRAW 0,4: DRAW 10,0: DRAW 0
,-2
[74] 1222 LET X=X-11: PLOT X,Y
[00] 1224 RETURN
[2B] 1230 DRAW 2,0: DRAW 0,10: DRAW -4,0: DRAW 0,-10: DRAW 2
,0
[7E] 1232 LET Y=Y+11: PLOT X,Y
[DA] 1234 RETURN
[35] 1240 DRAW 2,0: DRAW 0,-10: DRAW -4,0: DRAW 0,10: DRAW 2
,0
[80] 1242 LET Y=Y-11: PLOT X,Y
[64] 1244 RETURN
[04] 1400 IF X<6 OR Y<6 OR X>249 OR Y>169 THEN GO SUB 9999:
RETURN
[FC] 1401 LET L=USR 60972
[E1] 1402 IF L=56 THEN GO TO 1410
[E7] 1403 IF L=53 THEN GO TO 1420
[F7] 1404 IF L=55 THEN GO TO 1430
[01] 1405 IF L=54 THEN GO TO 1440
[01] 1406 GO TO 1401
[5] 1410 DRAW 0,-4: DRAW 0,8
[F7] 1412 LET X=X+2: PLOT X,Y
[79] 1414 DRAW 0,-4: DRAW 0,8
[F9] 1416 LET X=X+1: PLOT X,Y
[93] 1418 RETURN
[7F] 1420 DRAW 0,-4: DRAW 0,8
[03] 1422 LET X=X-2: PLOT X,Y
[83] 1424 DRAW 0,-4: DRAW 0,8
[05] 1426 LET X=X-1: PLOT X,Y
[F0] 1428 RETURN

```





```

[89] 1430 DRAW -4,0: DRAW 8,0
[6D] 1432 LET Y=Y+2: PLOT X,Y
[8D] 1434 DRAW +4,0: DRAW 8,0
[6F] 1436 LET Y=Y+1: PLOT X,Y
[A7] 1438 RETURN
[93] 1440 DRAW -4,0: DRAW 8,0
[19] 1442 LET Y=Y-2: PLOT X,Y
[97] 1444 DRAW -4,0: DRAW 8,0
[1B] 1446 LET Y=Y-1: PLOT X,Y
[81] 1448 RETURN
[65] 1500 IF X<8 OR X>247 OR X>167 THEN GO SUB 9999: RETURN

[60] 1501 LET L=USR 60972
[AA] 1502 IF L=56 THEN GO TO 1510
[80] 1503 IF L=53 THEN GO TO 1520
[BE] 1504 IF L=54 THEN GO TO 1530
[CD] 1505 IF L=55 THEN GO TO 1540
[9A] 1506 GO TO 1501
[F7] 1510 DRAW 0,3: DRAW 0,-6: DRAW 3,3: DRAW -3,3
[7C] 1511 LET X=X+4: PLOT X,Y: DRAW 0,3: DRAW 0,-6
[59] 1512 LET X=X+1: PLOT X,Y
[F2] 1513 RETURN
[01] 1520 DRAW 0,3: DRAW 0,-6: DRAW -3,3: DRAW 3,3
[88] 1521 LET X=X-4: PLOT X,Y: DRAW 0,3: DRAW 0,-6
[ES] 1522 LET X=X-1: PLOT X,Y
[FC] 1523 RETURN
[00] 1530 DRAW 3,0: DRAW -6,0: DRAW 3,-3: DRAW 3,3
[14] 1531 LET Y=Y-4: PLOT X,Y: DRAW 3,0: DRAW -6,0
[F1] 1532 LET Y=Y-1: PLOT X,Y
[06] 1533 RETURN
[16] 1540 DRAW 3,0: DRAW -6,0: DRAW 3,3: DRAW 3,-3
[10] 1541 LET Y=Y+4: PLOT X,Y: DRAW 3,0: DRAW -6,0
[FA] 1542 LET Y=Y+1: PLOT X,Y
[11] 1543 RETURN
[9D] 1600 IF X<12 OR Y<12 OR X>243 OR Y>163 THEN GO SUB 999
9: RETURN
[E5] 1601 LET I=USR 60972
[95] 1602 IF I=56 THEN GO TO 1610
[98] 1603 IF I=53 THEN GO TO 1620
[6B] 1604 IF I=55 THEN GO TO 1630
[B5] 1605 IF I=54 THEN GO TO 1640
[65] 1606 GO TO 1601
[H0] 1610 LET X=X+1: PLOT X,Y: DRAW 0,-10: DRAW 10,0: DRAW 0
,10: DRAW -10,0: DRAW 0,-10
[78] 1611 CIRCLE X+13,Y,2
[7E] 1612 LET X=X+13: PLOT X,Y
[57] 1613 RETURN
[FB] 1620 LET X=X-10: PLOT X,Y: DRAW 0,-10: DRAW -10,0: DRAW
0,10: DRAW 10,0: DRAW 0,-10
[84] 1621 CIRCLE X-13,Y,2
[90] 1622 LET X=X-16: PLOT X,Y
[61] 1623 RETURN
[CF] 1630 LET Y=Y+10: PLOT X,Y: DRAW 10,0: DRAW 0,10: DRAW -
10,0: DRAW 0,-10: DRAW 10,0
[8E] 1631 CIRCLE X,Y-13,2
[9A] 1632 LET Y=Y+16: PLOT X,Y
[6B] 1633 RETURN

```



```

[DB] 1640 LET y=y-10: PLOT x,y: DRAW 10,0: DRAW 0,-10: DRAW
-10,0: DRAW 0,10: DRAW 10,0
[98] 1641 CIRCLE x,y-13,2
[16] 1642 LET y=y-16: PLOT x,y
[75] 1643 RETURN
[9A] 1700 INPUT "Menoz: "; LINE m$: IF M$="" OR LEN M$>10 THEN
GO TO 58
[E7] 1701 SAVE m$SCREEN$
[68] 1702 GO TO 69
[9A] 1710 LOAD ""SCREEN$
[71] 1711 GO TO 69
[25] 1720 POKE 60029,1: POKE 60066,32: LET r=1: LET s=1: POK
E 60062,r: POKE 60064,s: POKE 60041,1: PRINT : RANDOMIZE USR
60000
[01] 1723 LET I=USR 60972: BEEP .1,60: POKE 60029,0: PRINT :
RANDOMIZE USR 60000: POKE 60029,1
[78] 1725 IF I=56 THEN IF S<31 THEN LET S=S+1: GO TO 1750
[23] 1726 IF I=53 THEN IF S>0 THEN LET S=S-1: GO TO 1750
[27] 1727 IF I=56 THEN IF R>0 THEN LET R=R-1: GO TO 1750
[68] 1728 IF I=54 THEN IF R<22 THEN LET R=R+1: GO TO 1750
[96] 1729 IF I=14 THEN GO TO 1780
[76] 1730 IF I=13 THEN GO TO 70
[07] 1750 POKE 60062,r: POKE 60064,s: PRINT : RANDOMIZE USR
60000: GO TO 1723
[81] 1780 POKE 60029,0: POKE 60041,0: LET I=USR 60972: POKE
60066,1: PRINT : RANDOMIZE USR 60000: PAUSE 30: LET I=1: GO
TO 1720
[0F] 3000 RANDOMIZE USR 64828: FOR I=1 TO 23: RANDOMIZE USR
3190: BEEP .005,-15: NEXT I: CLS: PRINT "Instrukcie": PRIN
T "====="
[71] 3003 PRINT "Kreslenie ciary-5,6,7,8": PRINT "Pohyb bodu
m-1,2,3,4": PRINT ""Odpor-R, potom smer (5 az 8)"
[AA] 3014 PRINT "Kondenzator-C, potom smer": PRINT "Dioda-D,
potom smer"
[6B] 3018 PRINT "Tranzistor-T, potom smer""Logicky clen-L,
potom smer""Nahratie na NG-S""Nahratie z NG-J"
[16] 3026 PRINT "U-CLS""Popis-P; Zam-G; Prepoj-B""COPY-2;
Inverzia-0": PRINT "Pri popise obrazku pohybujeme.""svetly
bodom pomocou tlacitok""5,6,7,8.""Ked chceme zapisat znak
na pozii""Ciu kde sa prave nachadzame"
[14] 3040 PRINT "stlacime SYMBOL SHIFT a potom""klavesu s p
ozadavanyim symbolom."
[14] 3044 PRINT "Navrat z popisu je po stlaceni"" ENTER"
;0; " NAVRAT = LUBOVOLNE TLACITKO"
[31] 3048 PAUSE 0: FOR I=1 TO 25: RANDOMIZE USR 3190: BEEP .
005,-2: NEXT I: CLS: RANDOMIZE USR 64848: GO TO 58
[1B] 4000 CIRCLE X,Y,1: RETURN
[EB] 4020 DRAW -4,0: DRAW 8,0: RETURN
[97] 9900 DATA 195,153,234,245,197,213,229,62,16,215,62,0,21
5,62,17,215,62,7,215,62,18,215
[BC] 9902 DATA 62,0,215,62,19,215,62,1,215,62,20,215,62,0,21
5,62,21,215,62,0,215,62
[1B] 9904 DATA 22,215,120,215,121,215,122,215,225,209,193,24
1,201,245,197,213,229,6,0,14,0,22,32

```



```

[1E] 9906 DATA 205,99,234,225,209,193,241,201,205,142,2,123,
254,255,32,248,205,142,2,32,251,22,0,205,30,3,48,244,79,175,
71,201
[2A] 9910 DATA 33,0,64,6,24,197,6,0,126,238,255,119,35,16,24
9,193,16,243,201
[22] 9911 DATA 245,197,213,229,17,200,175,33,0,64,1,0,24,237
,176,225,209,193,241,201
[23] 9912 DATA 245,197,213,229,17,0,64,33,200,175,1,0,24,237
,176,225,209,193,241,201
[1E] 9998 CLEAR : SAVE "SCHEMY" LINE 1: VERIFY "": RUN
[81] 9999 BORDER 0: BEEP .1,13: EOFOR 7: RETURN

```

uschovaný text

Poslední program je strojní rutina, která je určena začátečníkům v assembleru, ale i všem ostatním šouralům.

Rutina provádí výpis stále stejného textu na obrazovku. Pokud rutinu vylistujete nějakým editorem, text tam nebude a vlastně bude.

Až přijdete na tu jednoduchou fintu, možná Vás napadne, že by se tímto způsobem dal zatajit text ve vlastním strojáku před různými šouraly. Zkuste to...

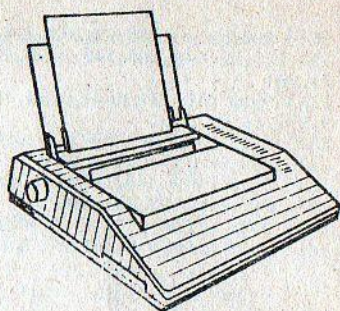
```

[CD] 10 REM uschovany text
[FF] 20 RESTORE
[0C] 30 LET s=0
[D9] 40 FOR i=65400 TO 65436
[A6] 50 READ a: POKE i,a
[E7] 60 LET s=s+a
[A9] 70 NEXT i
[8E] 80 IF s<>6313 THEN PRINT "chyba v datech": STOP
[C9] 85>PRINT "Kde je v rutine ulozen""tento text ?"
[A9] 90 RANDOMIZE USR 65400
[D1] 95 CLEAR 98303
[72] 100 DATA 33,135,255,6,22,126,238,255
[05] 110 DATA 215,35,16,249,24,242,201,177
[10] 120 DATA 190,179,186,171,186,179,223,181
[42] 130 DATA 172,182,223,222,223,190,175,173
[A2] 140 DATA 182,179,223,222,223

```



**polska
drukarka
d 100 m**



STRUČNÝ PŘEKLAD NÁVODU

ŽÁDOST
ČTENÁŘŮ
AP

HT CHR\$ (9); TABULACE SLOUPCOVÁ

Tento kód vykoná tabulaci podle prvního zadání a pak již automaticky nastavuje tabulaci na každý osmý znak.

LF CHR\$ (10); ODŘÁDKOVÁNÍ

Po zadání kódu LF jsou data v buferu tiskárny smazána a papír je posunut o jeden řádek. Platí i v případě, že shromážděná data vyjadřují prázdný řádek. Kód LF také ruší široký tisk, který byl zadán kódem SO a ESC SO. Délka posunu papíru, je nastavena pomocí kódu ESC 2, ESC 3.

CR CHR\$ (13); NÁVRAT VOZÍKU

Kód vymaže data z bufe-tiskárny. Pokud je zaveden signál AUTO FEED XT (PIN 14) ve stavu „LOW“, kód CR způsobí

posun papíru o jeden řádek.

Kód CR s posuvem papíru o jeden řádek zároveň zruší široký tisk zadáný kódem SO. V případě, kdy signál AUTO FEED XT je ve stavu „LOW“ a nebo přepínač DIL K1.4 je na pozici ON, je vykonání kódu CR zajištěno přes LF.

SO CHR\$ (14); ŠIROKÝ TISK

Po zadání tohoto kódu tiskne tiskárna v tom samém řádku znaky jako znaky široké. Kód SO je rušen kódy LF, DC4, VT, FF, CAN a CR. V jednom řádku lze kombinovat tisk znaků širokých i normálních. (Údaje o generátoru „nr2“ jsem vynechal, neboť ve většině tiskáren není zabudován).

SI CHR\$ (15); ZHUŠTĚNÝ TISK
Po zadání tohoto kódu jsou znaky tištěny **zhuštěně**;

10 LPRINT CHR\$ (15); "Mode"

20 LPRINT CHR\$ (18); "Mode"

10 = Mode | 20 = Mode

Kód SI je rušen kódem DC2. Jestliže s kódem SI zdstane zadán kód SO, bude tiskárna tisknout **široké znaky zhuštěně**

DC2 CHR\$ (18); ZRUŠENÍ ŠIROKÉHO TISKU

Kód zruší široký tisk, který byl zadán kódem SI.

DC4 CHR\$ (20); ZRUŠENÍ ŠIROKÉHO TISKU

Kód DC4 ruší široký tisk zadany kódem SO.

CAN CHR\$ (24); NULOVÁNÍ
V buferu jsou zrušena data z řádku, ve kterém byl kód uveden;

10 LPRINT "AHOJ!";

20 LPRINT "DOBŘÍ"; CHR\$ (24);

30 LPRINT "LIDÉ"

AHOJ! LIDÉ

DEL CHR\$ (127); DELETE
Smaže v buferu tiskárny poslední znak;

10 LPRINT "ČTENÁŘ";

20 LPRINT CHR\$ (127); "EM"

ČTENÁŘEM

ESC SO ŠIROKÝ TISK
CHR\$ (27); CHR\$ (14);
Stejně jako SO

ESC SI ZHUŠTĚNÝ TISK
CHR\$ (27); CHR\$ (15);
Stejně jako SI

ESC G CHR\$ (27); "G";
Tisk znaků s dvojitou intenzitou (tzv. Doublestrike). Před druhým tiskem těch samých znaků se papír posune automaticky o 1/144 (o 1/2 středu jehly). Tím se dá zlepšit kvalita tisku znaků.

ESC H CHR\$ (27); "H";
Zruší dvojitý tisk znaků zadany kódem ESC G.

ESC J CHR\$ (27); "J";
CHR\$ (n);
Tento kód udává o kolik mikrořádků se papír posune po vytištění jedno řádku.

ESC S O HORNÍ INDEX
CHR\$ (27); "S"; CHR\$ (0);
Provede tisk znaků v horní polovině řádku ve velikosti 1,6mm

10 LPRINT CHR\$ (27); "E";
20 LPRINT "Y-AX"; CHR\$ (27); "F";
30 LPRINT CHR\$ (27); "S"; CHR\$ (0); CHR\$ (15);
40 LPRINT "2";
50 LPRINT CHR\$ (27); "T"; CHR\$ (18);

----- Y-AX²

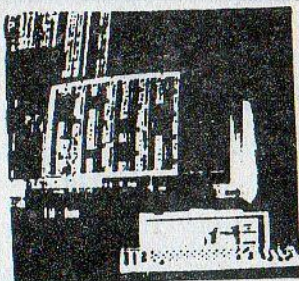
62, 136, 211, 127, 42, 28, [606]
 92, 237, 75, 79, 92, 9, [584]
 1, 19, 91, 112, 43, 113, [379]
 201, 254, 31, 202, 111, 91, [890]
 254, 13, 202, 198, 91, 254, [1012]
 128, 218, 205, 91, 214, 165, [1021]
 210, 16, 12, 79, 62, 27, [406]
 205, 205, 91, 62, 75, 205, [843]
 205, 91, 62, 8, 205, 205, [776]
 91, 62, 0, 205, 205, 91, [654]
 121, 198, 21, 203, 127, 32, [702]
 33, 237, 75, 123, 92, 38, [598]
 0, 111, 41, 41, 41, 9, [243]
 17, 1, 0, 14, 8, 229, [269]
 6, 8, 203, 6, 23, 25, [271]
 16, 250, 205, 205, 91, 13, [780]
 225, 32, 240, 201, 198, 144, [1040]
 71, 205, 56, 11, 33, 146, [522]
 92, 24, 223, 62, 27, 205, [633]
 205, 91, 62, 51, 205, 205, [819]
 91, 62, 24, 205, 205, 91, [678]
 6, 175, 14, 0, 197, 205, [597]
 198, 91, 62, 27, 205, 205, [788]
 91, 62, 75, 205, 205, 91, [729]
 62, 0, 205, 205, 91, 62, [625]
 1, 205, 205, 91, 22, 32, [556]
 62, 175, 144, 205, 176, 34, [796]
 213, 17, 0, 1, 205, 81, [517]
 91, 35, 209, 21, 32, 244, [632]
 193, 120, 214, 8, 71, 254, [860]
 239, 32, 203, 62, 27, 205, [768]
 205, 91, 62, 51, 205, 205, [819]
 91, 62, 24, 205, 205, 91, [678]
 62, 10, 205, 205, 91, 62, [635]
 13, 50, 240, 91, 219, 95, [708]
 245, 62, 251, 219, 254, 31, [1062]
 48, 20, 241, 23, 56, 242, [630]
 62, 255, 211, 95, 58, 240, [921]
 91, 211, 31, 175, 211, 95, [814]
 47, 211, 95, 201, 241, 201, [996]
 13, 0, 0, 0, 0, 0, [44]

Na četné žádosti čtenářů uvádím znovu výpis rutiny pro tisk obrazovky na tiskárně D100M
 v podobě výpisu pro uložení programem "secamdat". Pro jistotu ještě připomínám, že čísla
 uvedená v hranatých závorkách jsou čísla kontrolní.

BETA BASIC



piše Osvald Hons



SYNTAX PŘÍKAZŮ

(To co je v závorkách není povinné)

FILL *x,y*

FILL INK barva; *x,y*

FILL PAPER barva; *x,y*

Prohlédněte si funkci **FILLED()**

1; nebo 2; Vyplní souvislou plochu bodů barvy PAPER barvou INK.

3; Vyplní souvislou plochu barvou PAPER. *X,Y* je jeden bod souvislé oblasti. Jestli má již barvu INK, nic se nestane.

Příklad: Vyplnění kružnice

CIRCLE 128,87,30: FILL 128,80

Třeba však dát pozor na omezení hardwarem Spectra a Didaktiku, kde 8 * 8 pixelů nastavených přes dvě barvy PAPER a INK. Počet pixelů nastavených přes FILL je možné zjistit funkcí **FILLED()**.

GET číselná nebo řetězcová proměnná

GET řetězcová proměnná, *x,y* (šířka,výška)(typ)

1) Čtení z klávesnice. Přečte se jeden znak z klávesnice přičemž se čeká na stlačení klávesy a nečeká se na ENTER. Jestli je daná řetězcová proměnná, vloží se znak. Do číselné se vloží pro čísla 0-9, potom A=10, B=11 atd. Vhodné pro programy s MENU s řízením.

2) Z obrazovky se odloží obdélníková oblast se souřadnicemi levého horního rohu *x,y* do řetězcové proměnné. Tato se dá zobrazit na jiném místě přes PRINT nebo PLOT. Šířka a výška se udává v PRINT pozicích, *x,y* jsou standardní souřadnice PLOT. Vzniklý řetězec se nedá vytisknout

při CSIZE 0, ale je možné měnit rozměry přes normální příkaz CSIZE.

Poznámka: řetězec pozůstává z 9 znaků: 1. je CONTROLCODE udávající, že dalších 8 byte je kódovaných jako UDG. V případě větších úseků se automaticky ukládají kódy řízení kurzoru.

Pro OVER 0 se PAPER okraje vzorem přepisují, OVER 1 a vyvolají jiné efekty. Jestliže se neudá typ, potom se předpokládá 0, t. j. bezbarvý. Proto je vždy vybarvený aktuální barvy INK a PAPER. Typ 1 odloží i atributy, proto jsou při výpise dodrženy. **POZOR** však na hardwarové omezení počítače.

JOIN (číslo řádku)

JOIN řetězec nebo pole

(sledujte i SPLIT)

1) Příkaz spojí daný řádek s následujícím řádkem. Jestliže chybí číslo řádku, vezme se řádek s programovým kurzorem. Touto operací se ušetří 4 byte, zvýší se rychlost programu.

2) JOIN přesouvá řetězec a pole, COPY je kopíruje, ale syntaxi se neliší, proto jsou popsány společně.

a) **JOIN/COPY řetězec 1 (slicar) TO řetězec 2 (pozice)**

Přidá se řetězec 1 k řetězci 2.

Příklad: 10 LET a\$="12345",b\$="ABCDEFGH"

20 JOIN a\$ TO b\$

30 PRINT b\$: REM vypíše ABCDEFG12345

40 PRINT a\$: REM a\$ už neexistuje = chyba

Jestliže se v řádku 20 změní JOIN na COPY, řetězec a zůstane zachován.

b) **JOIN/COPY pole 1 (šlicar) TO pole 2 (pozice)**

Toto je výhodné v případě, když se pole naplnilo a je třeba je potřebovat znovu, ale potřebné aby bylo dimenzováno na větší rozměr.

Příklad: Mějme pole a\$(100,30), což nestačí a potřebujeme dalších 20 řetězců. Provedeme tento postup:

DIM b\$(20,30): JOIN b\$ TO a\$

Funkce LENGHT (1,"a\$") ukáže délku 120.

Vkládaný řetězec přispůsobuje svoji délku tomu, kde je vkládaný. Jestli je např. X znaků málo, potom utvoříme pomocné pole s potřebnou délkou a přidáme k němu krátké pole: **DIM b\$(1,40), JOIN a\$ TO b\$**

Nyní opět vytvoříme pole s původním názvem, ale s potřebnou délkou a přidáme k němu pomocné pole:

DIM a\$(1,40): JOIN b\$ TO a\$. Na konec je třeba vymazat vzniklé nadbytečné řetězce:

DELETE a\$(1 TO 2). Pozor! Zpracované pole mohou mít maximálně 2 rozměry.

KEYIN řetězec

Příkazem KEIN aš se do programu vezme obsah aš jako příkazový řádek. Tak se může program sám modifikovat. Důsledky tohoto příkazu leží mimo obzor tohoto manuálu.

Příklad: 10 LET aš="100 DATA"

20 FOR n=0 TO 9

30 LET aš+STR\$ (PEEK n)+", "

40 NEXT n

50 LET aš=aš(TO LEN aš-1): REM odebrání poslední čárky

60 KEYIN aš

Po RUM uvidíte, že program si sám vytvořil v řádku 100 příkaz DATA.

Příkaz INPUT "HESLO:";aš: KEYIN aš umístěný na začátku programu by mohl představovat ochranu programů, protože nestačí jen heslo, ale je třeba znát i číslo řádku.

V režimu KEYWORDS 3 nebo 4 se klíčová slova provedou na 1 býte tokeny.

KEYWORDS číslo

Číslo 0 a 1 řídí výstup (PRINT,LIST)

2 a 4 řídí vkládání programu.

Významy: 0 - zobrazují se grafické znaky.

1 - zobrazují se klíčová slova BB3, nastavené po startě.

2 - klíčová slova je nutné vkládat jako tokény, t.j. jednou klávesou.

3 - po vstupu je řádek atestovaný a všechna klíčová slova jsou převáděna na 1 byte. Akceptuje oba druhy vstupu.

4 - neexistuje "K" kurzor. Všechny příkazy je třeba psát po jednotlivých znacích.

LET proměnná = výraz, (proměnná=výraz)

Do jednoho příkazu je možné vložit více přiřazení.

LIST/LLIST (číslo řádku 1) TO (číslo řádku 2)

Vypíše se úsek programu. Je-li při tom nutné některou z hodnot vynechat, postupuje se jako při DELETE.

LIST DATA -> všechny proměnné

LIST VAL -> číselné proměnné

LIST VAL\$ -> řetězcové proměnné

Vypíše se aktuální obsah proměnných.

1) vypíše proměnné v pořadí: 1-číselná pole

- 2-řídící proměnné FOR cyklus
- 3-proměnné s 1 písmenovým jménem.
- 4-proměnné s více písmenovým jménem.
- 5-řetězcové pole.
- 6-řetězcové proměnné.

3. vypíše dále:

Pro pole se vypíše jen rozměry, nikoliv obsah. Pro řetězec jen prvních 15 znaků.

LIST DEF KEY

Příkaz vypíše definice všech nedefinovaných funkčních kláves.

LIST FORMAT číslo

Příkaz nastaví režim, ve kterém bude proveden výpis programu. Na začátku je nastavena 0.

- Význam čísel: 0 - klasický výpis počítače, ale řádky delší jak 32 znaků pokračují na dalším řádku až do páté pozice.
- 1 - pro struktury provádí odskok 1 meze ry.
 - 2 - pro struktury provádí odskok 2 meze ry.
 - 3 - jako 0, ale bez čísel řádků.
 - 4 - jako 1, ale bez čísel řádků.
 - 5 - jako 2, ale bez čísel řádků.

Aby byl výpis pěkný, je dobré vložit za každým THEN ELSE.

LIST PROC jméno

Vypíše se text celé procedury s jménem jména.

Poznámka: Po výpisu obsahuje adresa 23625 číslo prvního a adresa 57358 číslo posledního vypisovaného řádku. Tot je možné využít při programově řízeném DELETE nebo RENUM. Hodnotu je třeba číst funkcí DPEEK.

LIST REF údaj

Vypíše čísla řádků, na kterých se nachází reference na údaj. Může to být jméno proměnné, číslo, postupnosť znaků.

Protože tento manuál Beta Basicu verze 03 je skutečně velice rozsáhlý, zveřejňuji ho na pokračování. Bohužel proto bude i v dalších číslech roku 1993.

Vážení redakce,
zasílám Vám článek s výpisem programu, který byl odla-
děn na počítači Didaktik Gama 90. Výše uvedený program
užívám ve svých programech déle než jeden rok a proto
bych se chtěl podělit se čtenáři o jeho využití.

změna barev na obrazovce

Jindřich
Petrovský



Výše uvedený program je univerzální a lze ho uložit do libovolného programu, který se při vyvolání podprogramu zastaví a my můžeme libovolně měnit barvy pozadí a tisku na obrazovce. Po navolení barev běží normálně program dále. Podprogram je zcela relokovatelný (můžeme ho umístit na libovolné místo v paměti). Pokud nebudeme používat tiskárnu, lze ho umístit, vzhledem k jeho délce (184 bytů) i do vyrovnávací paměti (23296 - 23551).

Ovládání programu je velice jednoduché. Po vyvolání čeká počítač na stisk jedné ze čtyř kláves:

Po stisku prvních tří kláves navolíme patřičnou položku a každý další stisk libovolné klávesy (mimo ENTER - ten potvrzuje volbu) má za následek změnu barvy a to tak, že číslo barvy se postupně zvyšuje o jedničku. Po nejvyšším čísle následuje při dalším stisku číslo nejnižší, které se opět zvyšuje. Každá změna barvy se okamžitě přenáší na obrazovku. Po stisku klávesy ENTER můžeme měnit další položku a to do té doby, pokud nebudeme spokojeni.

P	- změna pozadí PAPER
I	- změna tisku INK
B	- změna jasu BRIGHT
V	- výstup

Stiskem klávesy V se podprogram ukončí a do proměnné ATRIB a ATTR-T (23695) se uloží číselná hodnota zvoleného atributu. Proměnná ATRIB byla zavedena z důvodů vícebarevných textů při kterých by se informace o barvě ztratila. Při používání pouze jedné barvy lze tuto proměnnou zrušit.

Podprogram lze libovolně upravovat tak, aby vyhovoval vašim požadavkům (změna ovládacích kláves, přidat blok pro změnu FLASH aj.), proto je výpis programu v assembleru

HISOFT GEN3 ASSEMBLER

Copyright © HISOFT 1983

All rights reserved

Pass 1 errors:00

```

          1 *D+
          2 +C-
          3
          4 ;NASTAVENI BAREV verse.1.
30000 5     ORG 30000
30000 6     ENT $
          7
29999 8 ATTRIB EQU 29999
          9
30000 10    LD A,56 ;nastav PAPER 7, INK 0, BRIGHT 0
30002 11    LD (23695),A;jako výchozí barva a ulož
30005 12    LD (23693),A;je do system.proměnných
30008 13 BARVA CALL KLAV ;test klávesnice
30011 14    CP "P"
30013 15    JR Z,PAP1 ;změň hodnotu PAPERu
30015 16    CP "I"
30017 17    JR Z,INK1 ;změň hodnotu INKoustu
30019 18    CP "B"
30021 19    JR Z,BRI1 ;změň hodnotu BRIGHTu
30023 20    CP "V"
30025 21    RET Z ;konec programu
30026 22    JR BARVA ;vrať se na začátek
          23
30028 24 PAP1 CALL KLAV
30031 25    CP 13 ;je-li stiskněn ENTER vrať se na
30033 26    JR Z,BARVA ;začátek
30035 27    LD A,(23695);vzvedněí hodnotu atributu
30038 28    AND x00111000;vyhodnoť pouze PAPER
30040 29    CP x111000 ;je PAPER ??
30042 30    JR Z,PAP2 ;pokud ano, odskoč
30044 31    LD A,(23695);vzvedni hodnotu atributu
30047 32    ADD A,8 ;PAPER=PAPER+1
30049 33 PAP3 LD (23695),A ;ulož novou hodnotu atributu
30052 34    CALL OBRAZ ;vykresli obrazovku novým
          35 ;atributem
30055 36    JR PAP1 ;a vrať se

```



30057 37 PAP2 LD A, (23695);vyzvedni hodnotu atributu
 30060 38 SUB 56 ;PAPER=0
 30062 39 JR PAP3
 40
 30064 41 INK1 CALL KLAY
 30067 42 CP 13 ;je-li stišťen ENTER, vrať se na
 30069 43 JR Z, BARVA ;začátek
 30071 44 LD A, (23695);vyzvedni hodnotu atributu
 30074 45 AND %111 ;vyhodnoť pouze INK
 30076 46 CP %111 ;je INK=7?
 30078 47 JR Z, INK2 ;pokud ano, odskoč
 30080 48 LD A, (23695);vyzvedni hodnotu atributu
 30083 49 INC A ;INK=INK+1
 30084 50 INK3 LD(23695), A ;ulož novou hodnotu atributu
 30087 51 CALL OBRAZ ;vykresli obrazovku novým
 52 ;atributem
 30090 53 JR INK1
 30092 54 INK2 LD A, (23695);vyzvedni hodnotu atributu
 30095 55 SUB 7 ;INK-0
 30097 56 JR INK3
 57
 30099 58 BRI1 CALL KLAY ;je-li stišťen ENTER, vrať se na
 59 ;začátek
 30102 60 CP 13
 30104 61 JR Z, BARVA
 30106 62 LD A, (23695);vyzvedni hodnotu atributu
 30109 63 AND %01000000;vyhodnoť pouze BRIGHT
 30111 64 CP %1000000;je BRIGHT-1?
 30113 65 JR Z, BRI2 ;pokud ano, odskoč
 30115 66 LD A, (23695);vyzvedni hodnotu atributu
 30118 67 ADD A, 64 ;BRIGHT=BRIGHT+1
 30120 68 BRI3 LD(23695), A ;ulož novou hodnotu atributu
 30123 69 CALL OBRAZ ;a vykresli jím obrazovku
 30126 70 JR BRI1
 30128 71 BRI2 LD A, (23695);vyzvedni hodnotu atributu
 30131 72 SUB 64 ;BRIGHT=0
 30133 73 JR BRI3
 74
 30135 75 OBRAZ LD(ATRIB), A
 30138 76 LD HL, 22528
 30141 77 LD DE, 22529
 30144 78 LD BC, 768




```

30147 79 LD(HL), A
30148 80 LDIR
30150 81 RET
      82
30151 83 KLAY CALL ZPOZD
30154 84 CALL #028E
30157 85 LD A, E
30158 86 CP 255
30160 87 JR NZ, KLAY
30162 88 KLAY1 CALL #028E
30165 89 JR NZ, KLAY1
30167 90 CALL #031E
30170 91 JR NC, KLAY1
30172 92 RET
      93
30173 94 ZPOZD LD B, 120
30175 95 SMYC1 LD C, 255
30177 96 SMYC2 DEC C
30178 97 JR NZ, SMYC2
30180 98 DJNZ SMYC1
30182 99 RET
      100
      101
      102 ;Výstupem je hodnota
      103 ;atributu v proměnné
      104 ;ATTRIB

```

Pass 2 errors:00

Table used: 207 from 401
Executes: 30000

Protože výpis programu, který nám J. Patrovský byl vytištěn na zapisovači ALFI a byl dosti nečitelný, nebylo možné ho přepísovat a otisknout, pře-psal jsem ho celý v Text Machinu.

hardcopy pro Vladimír Vojta robotrona

V čísle 5/AP jsem narazil na výpis programu COPY obrázky na K6304. Vlastním také tuto tiskárnu, avšak s rozhraním RS232.

Po přečtení uvedeného článku jsem si s rutinou trochu „pobral“, a upravil ji tak, že lze obrázek **odsadit**. V uvedeném výpisu stačí provést pouze několik drobných změn.

Nejdříve upravíme 20. řádek. Ten dosud vypadal takto;

```

205 91 62 65 205 205 (833)
Místo prvního bytu 205 vložíme hodnotu 241, kontrolní součet se tak změní na (869)

```

Dále přidáme na konec současného výpisu tři nové řádky

```

*
1. 0 205 205 91 6 10 (517)
2. 62 32 205 205 91 16 (611)
3. 249 201 0 0 0 0 (450)

```

Velikost odsazení volíme hodnotou posledního bytu v 1. přidaném řádku (označen *)

S tiskem zvětšeného obrázku je to již složitější. Nechal jsem se vyprovokovat a tak vznikl program, na kterém nyní pracuji.

animace

počiatkové
kroky
spectristu

pri tvorbe kreslených filmou

Animácia, podobne ako pri tvorbe kreslených filmov rieši sa pomocou vyvolávania sledu viacerých obrazov po sebe. Treba ale je obrazy (t. j. fázy) dopredu vytvoriť, pretože toto nie je možné vykonať zodpovedajúcou rýchlosťou počas animácie. Tu uvedená strojová rutina pozostáva z dvoch hlavných častí. Pomocou prvej časti (40000-40068) možno vyvolať jednotlivé fázy na predom určené miesto obrazovky, zatiaľ čo druhá časť (40069-40117) nami vytvorené fázy umiestni do priestoru pamäte (zadnej).



Rozmer jedného obrazovkového okna čiže fázy nie je viazaný, hoci pomocou tu uverejnených rutín možno používať maximálne 32 charakterov široké a 8 charakterov vysoké okno. Nevykoná sa ani adresová kontrola medzi tretinami obrazoviek, preto dávajte pozor, aby okno neprekľzlo cez obrazovkovú tretinu! (Mimochodom toto je možné upraviť známym spôsobom pri takzvaných box-scroll-och).



Pri našom vzorovom príklade zhotovíme štyri 16 charakterov široké a 8 charakterov vysoké animačné fázy (počet fáz môže byť ľubovoľný a je obmedzený len priestorom voľnej pamäte). Nezávisle od pozície objavenej sa fázy, je vhodné pripraviť tieto fázy na určitom mieste, čo nech je napríklad ľavý roh obrazovky. Tomu zodpovedajúc nastavíme v registri DE hodnotu 16384. V registri HL udaná adresa 40121 je tá adresa pamäte, odkiaľ sa začnú ukladať fázy. V registri A udáme číslo aktuálnej animačnej fázy (1, 2, ...), v registri BC konečný počet animačných okien. To v danom prípade prostredníctvom

vom 16x8 okien značí 1152, teda okrem obrazových dát obsahuje aj dáta ATTE. Táto hodnota je preto dôležitá, lebo po fázach o toľkoto postupuje ďalej ukazovateľ pamäte. Rutina toto sledujúc vykoná obrazové dáta, následne prekopíruje aj atribúty na dané miesto. Spätne volanie dát sa však vykoná obrátene, avšak v tomto prípade môžeme dopredu určiť polohu ich objavenia sa a to na adresách 40118/40119. V oboch prípadoch je ale možné podľa potreby upraviť rozmery okien na určených adresách.



Riadenie strojového kódu sa deje z BASICu. Kotúľajme napríklad jeden sud v hornej tretine obrazovky. Vyhotovíme štyri fázy deja. V prvej časti BASICového programu vykoná sa fotografovanie obrazovky, vlastne jeden za druhým vykreslíme jednotlivé fázy, nastavíme čísla fázy a tieto potom uložíme (pomocou subrutiny umiestnenej na riadku 9000). Po „vyfotografovaní“, 4 fáz nasleduje riadiaca časť, ktorého vykonávacia subrutina je na riadku 9100. Tu nastavíme číslo fázy, resp. adresu pozície objavenia sa. Zadávaním postupne klesajúcej hodnoty času sa kotúľanie suda postupne zrýchľuje. Obraz nasledujúcej fázy stiera predošlý, s výnimkou jedného charakterového stĺpca, preto necháme na oboch stranách okna po jednom voľnom stĺpci charakterov.



Keď sme už obrazy raz uložili do pamäte, tie vo forme kódov môžeme vytiahnuť s rutinou spolu, tak, že najbližšie už nie je potrebné operáciu kreslenia opakovať. V tomto prípade je dĺžka štyroch fáz 4x1152=4608 bytov, dĺžka rutinu a adresy jej volania spolu 121 bytov, čo značí, že uloženie vykonáme príkazom **SAVE „názov“, CODE 40000, 4279**. Prirodzene, že potom už v BASICovom programe je potrebná len časť po riadku 340, ale aj toto je možné prepísať do strojového kódu.



Rutina nám poskytuje mnoho možností, od učebných až po naše vlastné určené hrám.


```

10 REM prvá fáza
20 GOSUB 600
30 PLOT 10,144: DRAW 108,0
40 PLOT 40,112: DRAW 0,61
50 PLOT 19,165: DRAW 44,-44:
  DRAW 47,0
60 PLOT 19,121: DRAW 44,44:
  DRAW 47,0
70 LET a=1: GOSUB 9000
80 PAUSE 100: CLS
90 REM druhá fáza
100 GOSUB 600
110 PLOT 23,167: DRAW 35,-51:
  DRAW 47,0
120 PLOT 34,113: DRAW 13,60:
  DRAW 47,0
130 PLOT 15,126: DRAW 52,34:
  DRAW 47,0
140 PLOT 11,149: DRAW 60,-12:
  DRAW 47,0
150 LET a=2: GOSUB 9000
160 PAUSE 100: CLS
170 REM tretia fáza
180 GOSUB 600
190 PLOT 28,171: DRAW 25,-56:
  DRAW 47,0
200 PLOT 12,154: DRAW 57,-25:
  DRAW 47,0
210 PLOT 12,132: DRAW 57,25:
  DRAW 47,0
220 PLOT 27,115: DRAW 25,56:
  DRAW 47,0
230 LET a=3: GOSUB 9000
240 PAUSE 100: CLS
250 REM štvrtá fáza
260 GOSUB 600
270 PLOT 33,173: DRAW 13,-60:
  DRAW 47,0
280 PLOT 15,159: DRAW 52,-34:
  DRAW 47,0
290 PLOT 11,137: DRAW 60,12:
  DRAW 47,0
300 PLOT 22,118: DRAW 36,52:
  DRAW 47,0
310 LET a=4: GOSUB 9000
320 PAUSE 100: CLS
330 PLOT 0,111: DRAW 255,0:
  PLOT 0,110: DRAW 255,0:
  PLOT 0,109: DRAW 255,0
340 REM riadiaci blok
350 LET a=1: LET b=0: LET c=0
360 LET i=7
370 GOSUB 9100
380 LET a=a+1
390 IF a>4 THEN LET a=1
400 LET c=c+1
410 PAUSE i
420 IF cv16 THEN GOTO 360
430 GOSUB 9100
440 LET a=a-1
450 IF av1 THEN LET a=4
460 LET c=c-1
470 PAUSE i
480 IF c>0 THEN GOTO 430
490 LET i=i-1
500 IF iv1 THEN STOP
510 GOTO 370
600 CIRCLE 41,143,31: PLOT
  88,174: DRAW 0,-62,-PI:
  PLOT 40,174: DRAW 48,0:
  PLOT 40,112: DRAW 48,0:
  RETURN
9000 POKE 40120, a: RANDOMIZE
  USR 40069: RETURN
9100 POKE 40120, a:
  POKE 40118, c:
  POKE 40119, 64:
  RANDOMIZE USR 40000:
  RETURN

```


40000	237, 91, 182, 156	LD	DE, (40118)
40004	33, 185, 156	LD	HL, 40121
40007	58, 184, 156	LD	A, (40120)
40010	1, 128, 4	LD	BC, 1152
40013	61	DEC	A
40014	40, 3	JR	Z, 40019
40016	9	ADD	HL, BC
40017	24, 250	JR	40013
40019	62, 64	LD	A, 64
40021	1, 16, 0	LD	BC, 16
40024	197	PUSH	BC
40025	237, 176	LDIR	
40027	193	POP	BC
40028	235	EX	DE, HL
40029	9	ADD	HL, BC
40030	235	EX	DE, HL
40031	61	DEC	A
40032	32, 246	JR	NZ, 40024
40034	22, 88	LD	D, 88
40036	58, 182, 156	LD	A, (40118)
40039	95	LD	E, A
40040	58, 183, 156	LD	A, (40119)
40043	214, 64	SUB	64
40045	203, 63	SRL	A
40047	203, 63	SRL	A
40049	203, 63	SRL	A
40051	130	ADD	A, D
40052	87	LD	D, A
40053	62, 8	LD	A, 8
40055	1, 16, 0	LD	BC, 16
40058	197	PUSH	BC
40059	237, 176	LDIR	
40061	193	POP	BC
40062	235	EX	DE, HL
40063	9	ADD	HL, BC
40064	235	EX	DE, HL
40065	61	DEC	A
40066	32, 246	JR	NZ, 40058
40068	201	BET	
40069	17, 0, 64	LD	DE, 16384

40072	33, 185, 156	LD	HL, 40121
40075	58, 184, 156	LD	A, (40120)
40078	1, 128, 4	LD	BC, 1152
40081	61	DEC	A
40082	40, 3	JR	Z, 40087
40084	9	AD	HL, BC
40085	24, 250	JR	40081
40087	235	EX	DE, HL
40088	62, 64	LD	A, 64
40090	1, 16, 0	LD	BC, 16
40093	197	PUSH	BC
40094	237, 176	LDIR	
40096	193	POP	BC
40097	9	ADD	HL, BC
40098	61	DEC	A
40099	32, 248	JR	NZ, 40093
40101	33, 0, 88	LD	HL, 22528
40104	62, 8	LD	A, 8
40106	1, 16, 0	LD	BC, 16
40109	197	PUSH	BC
40110	237, 176	LDIR	
40112	193	POP	BC
40113	9	ADD	HL, BC
40114	61	DEC	A
40115	32, 248	JR	NZ, 40109
40117	201	RET	
40118	0	DATAB	0
40119	64	DATAB	64
40120	2	DATAB	2
40121	...		Start. adresa obraz.dat

Pre Amatérsky programátor z maďarského časopisu „SPECTRUM VILÁG“, preložil Eugen H. Becz

Pozn. red.: Príspevek Eugena H. Becze byl použit v autorské podobě, a protože ho autor napsal v textovém editoru Text Machine, který i já používám pro tvorbu předloh AP, doplnil jsem text pouze titulkem a rámečkem. Príspevek tedy neprošel žádnou jazykovou úpravou. Petr Černý

poznámky

Jan Drexler

ke zvukovému generátoru

Někteří tuzemští výrobci, např. Skalica a.s., fa. BEST a další, vyrábí zvukové generátory s IO-3-8912. Tento IO obsahuje 8-bitovou bránu, kterou výrobci bohužel nechávají bez povšimnutí, tzn. že její vývody nejsou bez úpravy využitelné.

V takovém případě by ovšem pro výrobce bylo výhodnější používat IO AY-3-8913, který je s AY-3-8912 slučitelný až na to, že uvedenou bránu V/V neobsahuje a tudíž je i levnější. Škoda, že naši výrobci AY-3-8913 nepoužívají, cena zvukových generátorů by byla menší a spotřebitel zbytečně neplatil bránu V/V, kterou stejně nemůže bez úpravy využívat.

Nevšední využití AY-3-8913 je pro tzv. podporovač neklidu. Využívá se infrazvuku o kmitočtu 7 nebo 14 (2. harmonická) Hz, který je generován jakožto aproximace sinusovky vnitřními převodníky D/A ve zvukovém IO. Pokud někteří hosté při návštěvě u Vás se nemohou rozhodnout opustit pohodlné křeslo a vydat se konečně na cestu domů, nahrajete program

INFREASOUND a infrazvuk ze zvukového generátoru zesílíte na odpovídající výkon nejlépe stejnosměrným zesilovačem a zavedete do reproduktorových soustav. Infrazvuk vyvolává velice neklidné pocity a tak není divu, že pokud použijete správný kmitočet, zesílení a odpovídající elektroakustický měnič, hosté prchají pryč v bačkorách, zanechávajíce svršky, deštníky, klobouky, čepice a další věci.

Zajímavé pokusy jsou i s ultrazvukem, který AY-3-8913 také snadno zvládne. Musíme však použít ultrazvukový reproduktor (dá se koupit asi za 40 až 80,- Kč). Málokdo ví, že ultrazvukem kolem 22.5 kHz lze odpuzovat komáry a na

tomto principu pracují i zahraniční odpuzovače. Zde ovšem velice záleží

**NÁPADY
ČTENÁŘŮ
AP**

nejen na kmitočtu, ale i na druhu hmyzu a co funguje na komáry na Amazonce, nemusí odpuzovat naše. S AY-3-8913 a ZXS lze např. sestavit dálkoměr (ZXS počítá čas mezi vysláním odrazu a prvou ozvěnou) a řadu dalších přístrojů. Můžete si vyzkoušet vliv ultrazvuku i na domácí zvířata.

Nebude dlouho trvat a běžné telefony budou mít namísto impulsní volby (přerušování účastnické smyčky) volbu tónovou. Zde se volí dvojicemi kmitočtu, což pro AY-3-

8913 není žádný problém, kmitočty však musí mít upravený tvar tak, aby obsahoval minimum vyšších harmonických. Pokud si zavoláte např. do Ameriky, můžete tam na mnoha místech své ZXS a AY-3-8913 pro novou telefonní volbu využívat.

bné vyšetření dáte v cizině téměř 100. -DM.

Zajímavé je ověřování tzv. precedenčního efektu: do

ho značky a do pravého tyto značky o několik de-

sítek ms spožděné a zjišťujeme, při jakém

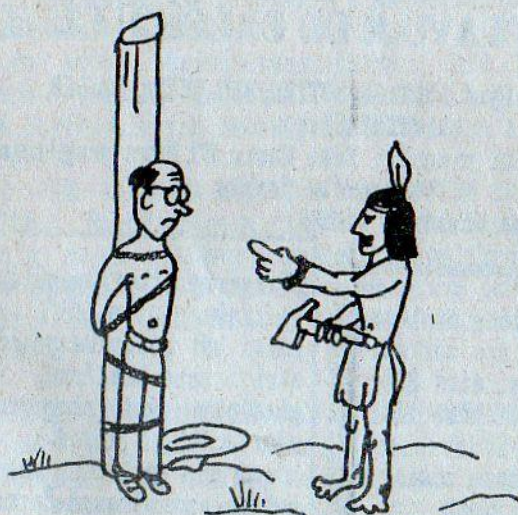
zpoždění odlišíme pí- vodní značku od zpožděné

(bude to při zpoždění kolem 40 ms)

Se zvukovým generátorem a ZXS si můžete ověřit i frekvenční rozsah Vašeho sluchu, musíte ovšem použít kvalitní zesilovač a sluchátka. Za podo-

**NÁPADY
ČTENÁŘŮ
AP**

-rex-



„Buď rád, bledá tváři, že jsem Tě osvobodil od té zářící tabule s hejblátky. Potřebuješ chytit na slunci pár barevných atributů.“

Kresba: Pavel Macek

JAK SI ZAJISTIT PRAVIDELNÝ ODBĚR

ap

V ROCE
1993

Nejjednodušší asi bude,
předplatit si ho na adrese
redakce a odeslat složenkou
typu "C" částku 132,-Kč.

Nebo si zajistit
pravidelný odběr
v některé
z těchto
prodejen:

- **KLUB 602** - Martinská 5, PRAHA 1
- **PERPETUM** - Ševčenkova 22, BRATISLAVA
- **APAS** prodej počítačů a příslušenství
Budovatelská 905, TŘEBÍČ
- **ELEKTROSERVIS**, SNP 1443-31/5
POVAŽSKÁ BYSTRICA

PŘIPRAVUJI DO DALŠÍCH ČÍSEL:

- O tom co je... ROTACE, BUTINA, BIT CHYBA, LOADER,
INTERPRET aj.
- Pokračování manuálu Beta Basic 03, Kurz programování
strojového jazyka
- Grafika na tiskárně D100M

Pokud posíláte disketu
nebo kazetu pro nahrání
programů ze sítě SPD,
nezapomeňte přiložit
20,-Kč
jinak Vám bude zásilka
vrácena zpět
nevyplaceně!

Příspěvky, které posíláte
ke zveřejnění v "AP",
musí být psány na psacím
stroji nebo tiskárně,
předlohy (výpis programu,
nákras, schém a pod.)
musí být vytištěny nebo
nakresleny výrazně-černě,
aby mohly být překopiro-
vány a použity do tisku.

PŘÍŠTÍ ČÍSLO VYJDE V BŘEZNU 1993

DIDAKTIK M S PAMĚTÍ 128 KB

SAM ZEMŘEL ZXS 48 KB DOŽÍVÁ

SINCLAIR 128 PŘEŽIJE ROK 2000

Úprava počítače Didaktik M (resp. ZX Spectrum, Didaktik Gama) na 128 kB paměti, která mj. umožňuje:

- » využívat převážnou většinu programů a her určených původně jen pro stovacetosmičky (ZXS 128K, AMSTRAD+2, +2A, resp. +3)
- » rozšířenou paměť používá ZXS128 jako disketovou jednotku (tzv. RAM disk s podstatně rychlejší komunikací než např. D40).
- » využití kopírovacích programů s kapacitou 128kB známých ze Sinclair 128kB.
- » vstup do světa stovacetosmičkových her, které se vyznačují větším počtem (zpravidla) kvalitnějších obrázků, čtyřkanálovým zvukovým doprovodem a dalšími výhodami.
- » NESTAČÍ VÁM KAPACITA KARTOTÉK A TEXTOVÉHO EDITORU PRO ZXS 48kB? Stovacetosmičkovské kartotéky a editory zpravidla mají volnou paměť několikrát větší.
- » spolu s úpravou je zdarma dodáno ukázkové software pro Sinclair 128kB demonstrující možnosti upraveného Didaktiku M

JIŽ NEMUSÍTE KAMARÁDOVI ZÁVIDĚT JE STOVACETOSMIČKU !
 Modernizujte svůj počítač v duchu technického pokroku
 a nenechte si ujít zážitek ze současných nových her
 i systémových programů pro ZXS 128kB.

OBJEDNÁVKY PRO DIDAKTIK M na uvedené adrese,
VERZE ZXS 48kB a **DIDAKTIK GAMA** na dotaz.

Jan DREXLER, Jahodová 2889, 106 00 PRAHA 10

cena inzerce

♦ řádkový inzerát: 1 řádek normální písmo = 0.50 Kč
1 řádek tučným písmem = 1, -- Kč

♦ plošná inzerce: celá strana (18x10cm) = 1.200, -Kč
půl strany (9x10cm) = 600, -Kč
čtvrt strany (9x 5cm) = 300, -Kč

♦ plošná inzerce na 2., 3., a 4. straně obálky jednou barvou;
1.800, -Kč
více barevně;
2.100, -Kč

Předplatitelé mají na veškerou inzerci 50% slevu
mimo inzerátů, které mají prokazatelně
výdělečný (podnikatelský) charakter.

Soukromý a zcela nezávislý časopis

amatérský programátor #1-93

pro amatérské programátory na počítačích
ZX SPECTRUM, DIDAKTIK a kompatibilních.

Vydává, tiskne a rozšiřuje Petr Černý, 407 61 Staré Křečany.
Předplatné na celý rok činí 132, -Kčs. Vychází každý druhý
měsíc. Cena jednoho čísla je 15, -Kčs a 42, -Kčs se hradí na
úhradu poštovného. Do tohoto čísla přispěli; J. Patrovský,
Vl. Vojta, P. Macek, E. H. Becz, J. Brossman, J. Drexler, J. Kotvald,
J. Szarka, O. Hons. Příprava stránek programem Text Machine
na počítači ZXs+ a tiskárně D100M.

Toto číslo bylo vydáno v lednu 1993.