



ATARI

klub Praha



ROČNÍK II.

Vydává 487. ZO Svazarmu —

ATARI KLUB v Praze 4.

Šéfredaktor a vedoucí redakční rady

JUDr. Jan Hlaváček.

Zástupce šéfredaktora

ing. Stanislav Borský.

Obálku navrhl RNDr. J. Tamchyna.

Adresa redakce:

487. ZO Svazarmu — ATARI KLUB Praha
REDAKCE

poštovní příhrádka 51

100 00 Praha 10

Řídí redakční rada: V. Bilek, ing. J. Bis-

kup, RNDr. J. Bok, CSc., ing. S. Borský,

ing. V. Friedrich, ing. O. Hanuš, RNDr.

L. Hejna, CSc., Z. Lazar, prom. fyz., CSc.,

ing. M. Vavřda, F. Tvrdek.

Otisk povolen se souhlasem redakce při zachování autorských práv a s uvedením pramene. Rukopisy nevyžádané redakci se nevracejí. Za původnost a věcnou správnost ručí autor.

Vychází šestkrát ročně. Neprodejné.

Členům klubu distribuováno zdarma.

Nepravidelné přílohy na objednávku jsou kompenzovány zvláštním klubovým příspěvkem.

Rozsah čísla 26 stran Neprošlo jazykovou úpravou.

Tiskne PORS, závod 001, reprografický provoz.

Do tisku předáno v XII/1988

Vydávání schváleno OV Svazarmu

Praha 4 a OŠK ONV Praha 4.

Evidenční číslo ÚVTEI 86 042.

© ATARI KLUB Praha, 1988

OBSAH

Ze života Svazarmu

Přednášky a kurzy	2
Jak plánujeme odborné kurzy ve školním roce 1988/89?	3
O ATARI přes hranice	4
Do Kladrub za počítači	4

Pro začátečníky

ATARI XL/XE a duchové	5
-----------------------	---

Tipy — triky

Studený start pomocí POKE	14
Listování při ladění programu	14
Znovu blikající kurzor	15

Otázky — odpovědi

Který BASIC je nejlepší	16
-------------------------	----

Uživatelské programy

Výukové programy pro ATARI	17
Universální výukový program	18

Technické novinky

TURBO MOS	21
-----------	----

Listárna

Křížek, hvězdička a tiskařská technika	22
Opravný lístek k příloze ZAK I-1	23

ZE ŽIVOTA SVAZARMU

PŘEDNÁŠKY A KURSY

Bez žákovských knížek a závěrečného vysvědčení proběhlo v minulém školním roce několik kursů, kterými náš klub přispěl ke zvýšení odborné vzdělanosti národa.

Na fotografiích Jana Eisnera přibližujeme atmosféru na přednáškách ing. Petra Jandíka, věnovaných ovládání disketové jednotky a užívání diskových operačních systémů. Kurs se uskutečnil v jarních měsících letošního roku.

Stejně úspěšný byl i kurs programování v jazyce LOGO, kterého lektorem byl Vojtěch Sedláček. Škoda jen, že posluchači ještě nemohli mít v rukou dlouho slibovanou učebnici.

I "Základy programování v BASICU", které přednášel František Tvrdek, přilákaly celou řadu posluchačů. Kdo přišel na přednášku později, neměl si kam sednout.

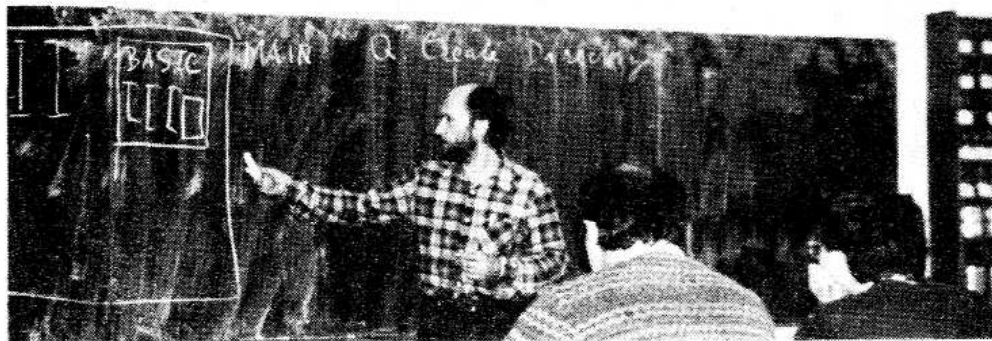
Zájem byl i o PASCAL v podání Miroslava Ondříška.

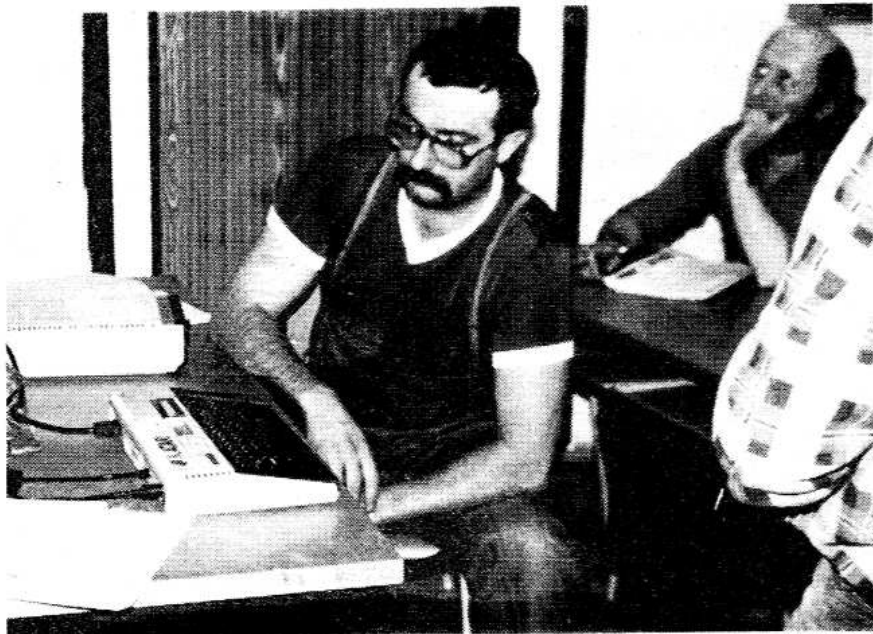
Jen Dr. Jiří Bok, CSc. si stěžoval, že zájem o assembler poklesl. Zřejmě většina těch, kteří cítili potřebu zvládnout toto "umění", využili kursy organizované v loňském roce.

Víme též o bohaté lektorské činnosti ing. Václava Fajty z Hradce Králové. S. Fajta vzkazuje všem vážným zájemcům (kroužkům), že je ochoten uskutečnit 2 - 3 denní kurs na téma "Jazyk symbolických adres mikroprocesoru 6502". Cílem přednáškového kursu je naučit posluchače základům programování v jazyku symbolických adres mikroprocesoru 6502 a seznámit posluchače s obsluhou, operováním a syntaxí makroassembleru ATMAS II. Zájemcům doporučujeme dohodnout se o podmínkách kursu přímo s lektorem na adrese:

ing. Václav Fajta
Baarova 1374
500 02 Hradec Králové

Je nepochybné, že i jinde proběhlo mnoho dalších kursů a přednášek (o jejichž existenci jsme však zpraveni nebyli). Výbor ATARI KLUBU děkuje touto cestou všem lektorům, kteří obětovali mnoho hodin svého volného času pro ostatní ataristy a věří, že s námi budou spolupracovat i v dalším období.





JAK PLÁNUJEME ODBORNÉ KURSY VE ŠKOLNÍM ROCE 1988/89?

Výbor pověřil organizováním odborných kursů v novém školním roce (a za tím účelem koptoval do svých řad) s. Zdeňka Lazara, CSc., který je rovněž členem redakční rady ZAK. Toto opatření by mělo přispět ke zkvalitnění práce na tomto úseku činnosti ATARI KLUBU. V současné době se připravuje ve spolupráci s lektory plán kursů na příští období. Bohužel ale nemůžeme - vzhledem k výrobní lhůtě ZAK - detailně informovat prostřednictvím časopisu. Plán kursů je zveřejněn na klubové nástěnce v SOU SSŽ v Ohradní ulici. Začátky kursů se plánují následovně:

BASIC	od října 1988
DISKETA	od listopadu 1988
PASCAL	od ledna 1989
LOGO	od ledna 1989

Nově se počítá se semináři:

TIPY a TRIKY	— leden 1989 (2 hodiny)
ADRESÁŘE, DATABANKY	— únor 1989 (2 hodiny)

Zápis pro jednotlivé kursy se uzavírá dnem 30. 10. 1988. Provádí se odevzdáním vlastnoručně zhotoveného lístku formátu A6 (asi 10x15 cm), na kterém bude napsáno jméno, příjmení, rodné číslo a adresa posluchače, kterémukoli členu výboru ZO nebo zasláním poštou s poznámkou KURSY k rukám s. Lazara. Z odevzdaných lístků bude získán přehled o zájmu a budou konkretizovány jednotlivé kursy. Lístky budou zaslány zpět jejich majitelům a budou sloužit jako průkazky pro volný vstup do kursu.

-HI-

O ATARI PŘES HRANICE

Několikrát jsme již informovali, že náš klub navázal kontakty s ataristy v Sovětském svazu. Někdy jsme též konzultovali, jak zajistit vzájemnou spolupráci, která by ne-
zůstala jenom na papíře. Nakonec se příklá-



níme k návrhu sovětských kolegů, který spo-
čívá v individuálním navázání počítačových
přátelství.

Kdo má zájem o výměnu zkušenosti o počí-
tačích ATARI, příp. výměnu programů, ale
také o vzájemné poznání se třeba i formou
výměnných návštěv, necht' napíše na adresu

142092 Moskevská oblast

g. Troick

Institut spektroskopii AN SSSR

s. Jegorov

Dopisy pište rusky (azbukou). Z následu-
jícího přehledu vyberte kam byste si chtěli
dopisovat: Baku, Buchara, Frunze, Jerevan,
Kijev, Leningrad, Moskva, Novorosijsk, Riga,
Samarkand, Simferopol a černomořské po-
břeží, Tallin, Taškent, Ulan-Ude, Vladimír.
Spojení zprostředkují soudruzi z Akademie
věd SSSR.

výbor

Jak nás informoval MUDr. Vladimír Kříž
z Rehabilitačního ústavu v Kladrubech (PSČ
257 62, okr. Benešov, tel. Vlašim - 0303—
42191 nebo 42461), mají pacienti léčeni
v tomto ústavu po úrazech a operacích mož-
nost v rámci celodenního rehabilitačního
programu učit se, zdokonalovat se nebo ob-
novovat ztracené dovednosti s výpočetní
technikou. Protože sám Dr. Kříž dal k dispo-
zici ústavní dílně výpočetní techniky svůj
vlastní počítač ATARI, i tento typ mají mož-
nost pacienti používat. Píše, že jim chybí jen
program pro plotter ATARI 1020.

I když samozřejmě nikomu nepřejeme, aby
se dostal do Kladruhu, přesto na přání velmi
rádi zveřejňujeme zajímavou informaci Dr.
Kříže: "Členové ATARI KLUBU, kteří utrpě-
li úraz či prodělali operaci (a mají bydliště
v Čechách, což je naše spádová oblast) a po-
třebují rehabilitaci (nezávisle na tíži úrazu
či operace), mohou být přijati na intenzivní
rehabilitaci do našeho ústavu. Zde pak mají

**DO
KLAD-
RU
POČÍTAČI**

možnost věnovat se v rámci celodenního
programu i této činnosti v praxi, včetně stu-
dia literatury.

Dále připravujeme přerušovaný kurs (3
krát 2 - 3 týdny) pro těžce tělesně postiže-
né občany, pro něž by mohla být práce s po-
čítačem nejen novou životní náplní, ale mož-
ností životního či profesionálního uplatnění.
Kurs plánovaný na příští rok (jaro - léto - pod-
zim), budeme inzerovat v časopise Svazu
invalidů ELÁN a v denním tisku. Pro první
(teoretický) běh počítáme s výběrem 30 zá-
jemců, z nichž pro další kursy, zaměřené i na
praxi, počítáme s počtem cca 20 osob."

Děkujeme za informaci. Zbývá dodat, že
ATARI KLUB Praha poskytne podle svých
možností účinnou pomoc v této vysoce hu-
mánní činnosti.

red

Pro

začátečnicků

Uživatelé mikropočítačů Commodore C64 nebo Sord M5 uvádějí jako jednu z předností svého počítače tzv. **spritovou grafiku**.

Sprity (čti sprajty = skřítki, duchové) jsou uživatelem definované plošné obrazce, kterými může pohybovat po obrazovce, aniž tím narušuje původně zobrazený text nebo kresbu.

ATARI XL/XE

A DUCHOVÉ

Obvykle je též možné nastavit prioritu zobrazení spritů, tj. mají-li původní kresbu zakrývat nebo naopak, testovat vzájemnou polohu spritů nebo spritu a původní kresby apod.

V málokteré příručce k počítačům Atari se lze dočíst, že i tyto počítače mají spritovou grafiku. Obvykle se označuje jako P/M grafika a umožňuje definovat až 8 spritů — 4 hráče (Player) a 4 střely (Missile). Všechny sprity jsou jednobarevné, přičemž vždy jeden hráč a jedna střela mají stejnou barvu. V programech obvykle využíváme zejména spritů typu hráč, neboť mají oproti střelám jednodušší způsob návrhu a širší oblast použití.

Bit a byte

Aby spritovou grafiku mohli využívat i začátečníci, kteří programují v jazyce Basic, musíme si nejprve objasnit pojmy bit a byte (čti bajt), bez jejichž znalosti se návrh spritů neobejde.

Bit je dvojková číslice, která může nabývat dvou hodnot — 0 a 1. Pomocí bitů budeme kódovat jednotlivé body při návrhu spritů, přičemž jednička bude představovat zobrazený bod, nula nezobrazený.

Osmice bitů tvoří jeden **byte** (například 01101011). V našem případě bude byte představovat osm bodů umístěných vedle sebe. Abychom nemuseli byty zapisovat pouze pomocí

nul a jedniček, budeme jejich hodnoty převádět z dvojkové soustavy do soustavy desítkové, takže osmi nulám bude odpovídat 0, osmi jedničkám 255.

Kdo není zběhlý v dvojkové aritmetice, tomu poslouží při převodu následující návod:

1. Jednotlivé bity označíme postupně (zleva doprava) jako
2. Ke každému bitu a_i ($i = 0$ až 7) vypočteme jeho váhu b_i podle vzorce:
3. Výslednou hodnotu získáme jako součet všech osmi vah, tj.:

Například byte 01101011 má desítkovou hodnotu 107, neboť platí:

$$\begin{array}{lll} a_7 = 0 & b_7 = 0 \times 2^7 = 0 \times 128 = 0 \\ a_6 = 1 & b_6 = 1 \times 2^6 = 1 \times 64 = 64 \\ a_5 = 1 & b_5 = 1 \times 2^5 = 1 \times 32 = 32 \\ a_4 = 0 & b_4 = 0 \times 2^4 = 0 \times 16 = 0 \\ a_3 = 1 & b_3 = 1 \times 2^3 = 1 \times 8 = 8 \\ a_2 = 0 & b_2 = 0 \times 2^2 = 0 \times 4 = 0 \\ a_1 = 1 & b_1 = 1 \times 2^1 = 1 \times 2 = 2 \\ a_0 = 1 & b_0 = 1 \times 2^0 = 1 \times 1 = 1 \end{array}$$

$$b_7 + b_6 + b_5 + \dots + b_1 + b_0 = 107$$

Návrh spritu

Obrazovka má v nejjemnějším rozlišení (grafika 8) rozměr 320x192 bodů. S touto rozlišitelností pracuje i spritová grafika. Jednotlivým bodům grafiky 8 budeme říkat **elementární body** neboli **pixels**. Například bod v grafice 15 (160x192 bodů) je tvořen 2 pixely, znak v grafice 0 se skládá z 8 x 8 pixelů apod.

Šířka spritu typu hráč je v základním zobrazení 8 elementárních bodů, obdobně jako u znaku. Jelikož k zakódování jednoho bodu je třeba jeden bit, slouží k vyjádření jednoho řádku spritu právě jeden byte. Na rozdíl od znaků však může být výška spritu (počet řádek) až 256 bodů. V tomto mezním případě je plocha pro sprite tvořena svislým pásem, který nahoře i dle přesahuje přes rámeček obrazovky (ta má pouze 192 řádků).

Chceme-li tedy navrhnout podobu spritu, připravíme si mřížku 8 x n bodů, kde n je předpokládaná výška. Do této mřížky zakreslíme tvar spritu a jednotlivé řádky odshora vyjádříme hodnotou příslušného bytu. Například k zakreslení spritu ve tvaru ducha ze hry Ghost Chaser potřebujeme mřížku 8 x 21 bodů. K zakódování potřebujeme tedy celkem 21 číselných hodnot (viz. obrázek).

Umístění spritů v paměti

V základním formátu zobrazení se sprite skládá z maximálně 256 řádků po 8 bodech. K uložení tvaru spritu do paměti je tedy třeba 256 bytů. Pro 4 sprity typu střela a oblast pomocné paměti musíme vyhradit celkem 2048 bytů (2 kB). Celá oblast paměti pro P/M grafiku je organizována takto (adresy vzhledem k počátku oblasti):

- 0000 — 0761 oblast pomocné paměti
- 0762 — 1023 oblast paměti pro střely
- 1024 — 1279 paměť hráče 0
- 1280 — 1535 paměť hráče 1
- 1536 — 1791 paměť hráče 2
- 1792 — 2047 paměť hráče 3

Obrázek spritu ducha ze hry Ghost Chaser:

7	6	5	4	3	2	1	0	
				X	X			00001100 = 12
				X	X			00011000 = 24
			X	X	X			00111000 = 56
	X	X	X	X	X			01111100 = 124
	X		X		X			01010100 = 84
	X	X	X	X	X			01111100 = 124
		X		X				00101000 = 40
		X	X	X				00111000 = 56
		X	X	X				00111000 = 56
		X	X	X	X			00111100 = 60
	X	X	X	X	X	X		01111110 = 126
X	X	X	X	X		X	X	11111011 = 251
X		X	X	X			X	10111001 = 185
		X	X	X				00111000 = 56
		X	X	X				00111000 = 56
		X	X	X	X			00111100 = 60
			X	X	X			00011100 = 28
			X	X	X			00011100 = 28
				X	X	X		00001110 = 14
					X	X		00000110 = 6
						X	X	00000011 = 3

Kromě tohoto základního zobrazení existuje ještě **redukované zobrazení spritů**. V tomto případě je předloha pro sprite vysoká maximálně 128 řádků, přičemž každý zobrazovaný řádek je tvořen dvěma elementárními řádky grafiky 8. V redukovaném zobrazení vyžaduje P/M grafika pouze 1024 bytů:

0000 — 0383 oblast pomocné paměti
0384 — 0511 oblast paměti pro střely
0512 — 0639 paměť hráče 0
0640 — 0767 paměť hráče 1
0768 — 0895 paměť hráče 2
0896 — 1023 paměť hráče 3

Oblast obrazové paměti pro P/M grafiku může být umístěna kdekoli v paměti RAM počítače, s jediným omezením: počáteční adresa celé oblasti musí být dělitelná číslem 2048 (u redukovaného zobrazení 1024). Jak takové vhodné místo najít? Nejlepší a nejjednodušší je umístit sprity na nejvyšší přístupné adresy paměti RAM.

Horní hranici použitelné paměti RAM vyjadřuje buňka MEMTOP na adrese 106, a to v násobcích 256 bytů (tj. ve stránkách). Pro naše potřeby stačí tuto hodnotu snížit o 8 ($8 \times 256 = 2048$) resp. o 4 ($4 \times 256 = 1024$), tj.:

POKE 106, PEEK (106)-8 pro normální zobrazení

POKE 106, PEEK (106)-4 pro redukované zobrazení

Nová hodnota buňky MEMTOP (PEEK (106) * 256) vyjadřuje počátek obrazové paměti pro spritovou grafiku.

Po provedení výše uvedeného příkazu dojde k rozporu mezi činností operačního systému a grafického procesoru. To se projeví chybami v zobrazení, nejčastěji ve formě náhodně se vyskytujících znaků ve spodní oblasti obrazovky. Grafický systém uvedeme do pořádku příkazem pro nastavení grafického režimu (například GRAPHICS 0)

Nyní již můžeme do vyhrazené paměťové oblasti uložit data definující vlastní podobu spritu. Nejprve si ale celou oblast paměti vyčistíme. Aby se sprite zobrazil přibližně uprostřed obrazovky, umístíme jej doprostřed paměti vyhrazené pro hráče 0:

```
10 POKE 106,PEEK(106)-3:GRAPHICS 0:PMB=PEEK(106)
20 FOR I=PMB*256+1024 TO PMB*256+2047:POKE I,0:NEXT I
30 FOR I=0 TO 20:READ A:POKE PMB*256+1146+I,A:NEXT I
40 DATA 11,23,55,124,34,124,40,53,56,60,126,251,185,56,56,60,28,28,14,6,3
```

Hodnota 1146 na řádce 30 určuje pozici prvního řádku obrazu spritu od počátku paměti pro P/M grafiku ($PMB \times 256$). Neboť střed obrazovky pro hráče 0 je dán jako:

$$(1024 + 1280) / 2 = 1152$$

začíná viditelná část spritu 6 řádků nad středem obrazovky.

Po spuštění předchozího programu se na obrazovce žádný sprite neobjevil. Je sice uložen v paměti, ale grafický koprocesor GTIA, který zajišťuje zobrazení P/M grafiky, dosud nedostal příslušné informace. Ty mu předáme obsazením příslušných paměťových buněk.

Systémové proměnné spritové grafiky

K inicializaci P/M grafiky a zobrazení spritů na obrazovce musíme nastavit hodnoty jednotlivých systémových buněk (proměnných), se kterými pracují grafický procesor ANTIC a koprocesor GTIA.

Jsou to následující paměťové buňky:

DMACTL (559) - řadič zobrazení

Slouží k nastavení základních charakteristik zobrazení, povoluje zobrazení základní i spritové grafiky. Režim zobrazení se nastavuje volbou jednotlivých bitů buňky (0 až 5):
bity 1 a 0 - šířka obrazovky

- 00 - 0 pixelů
- 01 - 256 pixelů (úzká)
- 10 - 320 pixelů (standardní)
- 11 - 384 pixelů (široká)

- bit 2 — povolení zobrazení pro střely
- bit 3 — povolení zobrazení pro hráče
- bit 4 — režim zobrazení spritů: 1 — normální (256 řádků)
0 — redukované (128 řádků)
- bit 5 — povolení zobrazení pro základní grafiku.

Standardní hodnota buňky 559 je rovna 34. Chceme-li povolit zobrazení hráčů, musíme hodnotu změnit příkazem:

POKE 559,58 pro normální zobrazení

POKE 559,42 pro redukované zobrazení

GPRIOR (623) — registr priority

Slouží k nastavení priority při současném zobrazení více spritů, resp. spritů a normální grafiky. Vyšší bity této buňky umožňují nastavení speciálních režimů P/M grafiky. Jejich popis však přesahuje rámec tohoto článku.

POKE 623,1 — hráči 0—3, barvy 0—3, rámeček

POKE 623,2 — hráči 0—1, barvy 0—3, hráči 2—3, rámeček

POKE 623,4 — barvy 0—3, hráči 0—3, rámeček

POKE 623,8 — barvy 0—1, hráči 0—3, barvy 2—3, rámeček

Číslování barev odpovídá prvnímu parametru příkazu SETCOLOR.

Při dosazení jiných hodnot (například 0 nebo 5) dochází k neočekávaným efektům při zobrazování spritů.

Režim, v kterém sprity překrývají základní grafiku, nastavíme příkazem:

POKE 623,1

PCOLR0 až PCOLR3 (704 až 707) — registry barev spritů

COLOR0 až COLOR4 (708 až 712) — registry barev kresby

Slouží k nastavení barev spritové i normální grafiky. Hodnota každé z uvedených buněk je rovna hodnotě výrazu:

$16 * \text{KÓD BARVY} + \text{JAS}$

kde kód barvy je číslo 0 až 15, které můžeme nalézt v uživatelské příručce k počítači. Například chceme-li nastavit barvu spritu 0 na bílou (kód barvy = 0), použijeme příkaz:

POKE 704,14

GRCTL (53277) — řadič P/M grafiky

Spolu s buňkou DMACTL slouží k povolení zobrazení spritů na obrazovce. Abychom viděli příslušný sprite, musí být povolen v obou buňkách současně:

POKE 53277,1 — povoluje střely

POKE 53277,2 — povoluje hráče

POKE 53277,3 — povoluje hráče i střely

K zobrazení hráčů na obrazovce je tedy nutné uvést příkazy:

POKE 53277 2:POKE 559,42

PMBASE (54279) — registr počátku P/M grafiky

Udává v násobcích 256 počátek paměti pro uložení předloh spritů. V ilustrovaném příkladě dosadíme do proměnné PMBASE hodnotu proměnné (basické) PMB, čili:

POKE 54279,PMB

SIZEP0 až SIZEP3 (53256 až 53259) — registry šířky hráčů

Slouží k nastavení skutečné šířky zobrazovaných hráčů. V normálním zobrazení má hráč šířku 8 pixelů (1 bod = 1 pixel). Kromě toho však existuje i možnost zobrazení hráče ve dvojnásobné a čtyřnásobné šířce (16 nebo 32 pixelů), tj. 1 bod = 2 pixely nebo 4 pixely). Šířku nastavujeme dosazením hodnoty do příslušné buňky:

- 0 nebo 2 normální zobrazení (8 pixelů)
- 1 dvojité zobrazení (16 pixelů)
- 3 čtyřnásobné zobrazení (32 pixelů)

Například příkaz

POKE 53256,0

nastaví normální šířku zobrazení pro hráče 0.

Přifazením hodnot všem výše uvedeným proměnným docílíme zobrazení spritu na obrazovce. Buňku DMCATL (na adrese 559) přitom nastavujeme jako poslední. K již uvedeně části programu tedy přidáme následující řádky a program znovu spustíme.

```
50 POKE 54279,PMB:REM PMBASE
60 POKE 53227,2:REM GRACATL
70 POKE 523,1:REM GPRIOR
90 POKE 53256,0:REM SIZEPO
90 POKE 53249,128:REM HPOSP0
100 POKE 701,14:REM PCOLRO
110 POKE 559,58:REM DMCATL
```

Význam příkazu na řádce 90 si uvedeme v následujícím odstavci. Pokud jsme program zapsali bez chyby, objeví se po spuštění uprostřed obrazovky bílý duch.

Vodorovný pohyb spritu

Abychom mohli vytvořený sprite použít v programu, musíme se naučit pohybovat jím po obrazovce. Jednodušší je pohyb vodorovný. Ke změně vodorovné souřadnice spritu stačí změnit hodnotu příslušné systémové buňky HPOSP0 až HPOSP3 (53248 až 53251). Změnou hodnoty této buňky o jednotku provede sprite pohyb o 2 elementární body. Středu obrazovky přitom odpovídá hodnota 128. Je-li hodnota buňky příliš nízká nebo příliš vysoká, není sprite vidět. Viditelný rozsah je asi 50 až 210.

Postupnou změnou hodnoty proměnné HPOSP0 můžeme tedy uvést sprite 0 do vodorovného pohybu. K ovládání spritu užijeme křížového ovladače a výše uvedený program rozšíříme o následující řádky:

```
120 X=128:Y=121
130 Z=STICK(0)
140 IF Z>4 AND Z<9 THEN X=X+1
150 IF Z>3 AND Z<12 THEN X=X-1
160 IF X<0 THEN X=0
170 IF X>255 THEN X=255
175 POKE 53249,X
500 GOTO 130
```

Po spuštění programu připojíme ovladač do předního konektoru a můžeme našim spritem pohybovat.

Svislý pohyb spritu

Svislou polohu spritu na obrazovce neurčuje žádná proměnná, ale pouze poloha kódů spritu v rámci vyhrazené oblasti paměti. Celá paměťová oblast (256, resp. 128 bytů) tedy představuje svislý pás na obrazovce, přičemž nižší adresy odpovídají výše uloženým obrazovým řádkům. Chceme-li tedy umístit sprite doprostřed obrazovky, nebudeme jeho data načítat ihned od počátku příslušné paměťové oblasti, ale přibližně do jejího středu (jako v našem případě).

Budeme-li nyní chtít spritem pohybovat nahoru nebo dolů, budeme muset změnit příslušnou část paměti. Celý obraz spritu musíme posunout dopředu (pohyb nahoru) nebo dozadu (pohyb dolů) o tolik bytů, o kolik bodů chceme sprite přemístit.

Tuto nepříliš složitou manipulaci zvládneme i v Basicu. Využijeme přitom skutečnosti, že číselný obraz spritu je uložen v řádce 40 našeho programu ve formě dat. Připíšeme další řádky programu:

```

130 IF Z=5 OR Z=9 OR Z=13 THEN Y=Y+1
190 IF Z=3 OR Z=10 OR Z=14 THEN Y=Y-1
200 IF Y<0 THEN Y=0
210 IF Y>234 THEN Y=234
220 RESTORE 40
230 POKE PMB*256+1023+Y,0
240 FOR I=0 TO 20:READ A:POKE PMB*256+1024+Y+I,A:NEXT I
250 POKE PMB*256+1045+Y,0

```

Příkazy na řádcích 230 a 250 mažou pozůstatky poslední zobrazené pozice spritu.

Spustíme-li nyní celý program, můžeme již spritem pohybovat do všech směrů. Svislý a šikmý pohyb je však pomalý a nedokonalý. To je způsobeno malou rychlostí interpretu Basicu. K vytvoření iluze plynulého pohybu budeme muset vytvořit podprogramy ve strojovém kódu.

Strojové rutiny pro svislý pohyb

Návrh strojových podprogramů pro pohyb spritu je pro znalce strojového kódu procesoru 6502 poměrně jednoduchou záležitostí. Jelikož tento článek je určen zejména pro začínající, uvedeme obě strojové rutiny již hotové:

DOLU	PLA	NAHORU	PLA
	PLA		PLA
	STA 204		STA 204
	PLA		PLA
	STA 203		STA 203
	LDY #21		LDY #21
LOOP1	LDA (203),Y	LOOP2	LDA (203),Y
	INY		DEY
	STA (203),Y		STA (203),Y
	DEY		INY
	DEY		INY
	CPY #255		CPY #23
	BNE LOOP1		BNE LOOP2
	RTS		RTS

Oba strojové podprogramy převedeme do dat a předchozí část programu (řádky 180 až 250) nahradíme následujícími:

```

130 DY=0
190 IF Z=5 OR Z=9 OR Z=13 THEN DY=1
200 IF Z=3 OR Z=10 OR Z=14 THEN DY=DY-1
210 IF Y+DY<0 OR Y+DY>234 THEN DY=0
220 IF DY>0 THEN A=USR(ADR(DOLUS$),PMB*256+1024+Y)
230 IF DY<0 THEN A=USR(ADR(NAHORUS$),PMB*256+1024+Y)
240 I=Y+DY

```

Řádek 250 zrušíme a řádek 120 nahradíme následujícím:

```
120 X=122:Y=121:GOBUS 1 000
```

Nyní ještě doplníme strojové podprogramy:

```

1000 DIM DOLUS$(21),NAHORUS$(21)
1010 RESTORE 1050
1020 FOR I=1 TO 21:READ A:DOLUS$(I,I)=CHR$(A):NEXT I
1030 FOR I=1 TO 21:READ A:NAHORUS$(I,I)=CHR$(A):NEXT I
1040 RETURN
1050 DATA 104,104,133,204,104,133,203,160,21,177,203,200,145,203,136,136,192,255,
,203,245,93
1060 DATA 104,104,133,204,104,133,203,160,1,177,203,136,145,203,200,200,192,23,2
03,245,93

```

Ted' již můžeme celý program spustit. Pohyb spritu již bude plynulý ve všech směrech. Budeme-li uvedené podprogramy využívat pro vlastní aplikace, musíme změnit podtržené hodnoty 21, resp. 23 na řádcích 1050, resp. 1060. Hodnota 21 vyjadřuje délku spritu v bytech (řádcích), a tím i velikost přesouvané paměti. Hodnota 23 je vždy o 2 vyšší. Obdobně bychom museli změnit i hodnotu 234 na řádku 210. Tato hodnota je rovna výrazu:

255 — délka spritu v bytech

a určuje nejnižší možnou pozici spritu na obrazovce i v paměti počítače

Ještě jednou svislý pohyb

Komu se nelíbí podprogramy ve strojovém kódu a nemá k nim důvěru, může docílit srovnatelného efektu i bez nich, pouze s využitím příkazu PRINT. Příkaz

PRINT "ATARI"

neprovádí nic jiného, než že ukládá jednotlivé znaky řetězce ATARI, resp. jejich kódy, do příslušné části obrazovkové paměti. Pokud bychom měli převést předlohu spritu do řetězce a ten dokázali vytisknout ne do paměti pro obrazovku, ale do paměti pro P/M grafiku, mohli bychom využít příkazu PRINT i pro pohyb spritu.

Nejprve uvedeme podprogram pro převod tvaru spritu do řetězce A\$. Tento podprogram umístíme na řádky 1000 až 1080 a přepíšeme jím původní strojové rutiny.

```
1000 DIM A$(23)
1010 RESTORE 40
1020 FOR I=2 TO 22:READ A
1030 IF A<34 OR (A>127 AND A<132) THEN A=A+32:GOTO 1050
1040 IF (A>53 AND A<95) OR (A>191 AND A<224) THEN A=A-64
1050 A$(I,I)=CHR$(A)
1060 NEXT I
1070 A$(1,1)=" ":A$(23,23)=" "
1080 RETURN
```

Příkazy na řádcích 1030 a 1040 provádějí byty z kódu ASCII do vnitřního zobrazovacího kódu počítače. První a poslední znak řetězce jsou mezery, které se zobrazí jako prázdné řádky a budou při pohybu mazat přebývající část původní předlohy.

Nyní nahradíme příkazy pro svislý pohyb (řádky 180 až 240) novými instrukcemi, které budou tisknout vytvořený řetězec A\$ na správné místo v paměti. Toho docílíme posunem ukazatele počátku videopaměti (na adresách 88 a 89) do oblasti paměti pro P/M grafiku.

```
130 IF Z=5 OR Z=9 OR Z=13 THEN Y=Y+1
190 IF Z=6 OR Z=10 OR Z=14 THEN Y=Y-1
200 IF Y<0 THEN Y=0
210 IF Y>0 THEN Y=324
220 POKE 89,Y:POKE 89,ST
230 POSITION 0,0:PRINT A$
240 POKE 93,OB1:POKE 89,OB2
```

Proměnné OB1 a OB2 slouží k uchování původní adresy počátku videopaměti. Tu uložíme následujícími příkazy:

```
120 X=128:Y=121:GOSUB 1000
132 POKE 752,1:OB1=PEEK(98):OB2=PEEK(99)
124 ST=PMS+4:REM ADRESA HRACE 0
```

P/M grafika a Turbo BASIC XL

Stejným způsobem jako dosud, můžeme se sprity pracovat i v populárním Turbo Basicu. Tento interpret nemá zvláštní příkazy pro P/M grafiku. Ke svislému pohybu spritů lze však s výhodou použít příkazy pro přesun v paměti MOVE a -MOVE. K tomu účelu nahradíme všechny příkazy pro svislý pohyb (120 a od 180 výše) následující částí programu:

```
120 X=122:Y=121
180 DY=0:PB=PBM*256+1024+Y
190 IF Z=5 OR Z=9 OR Z=13 THEN DY=1
200 IF Z=6 OR Z=10 OR Z=14 THEN DY=-1
210 IF Y+DY<0 OR Y+DY>234 THEN DY=0
220 IF DY>0 THEN MOVE PB,PB+1,21:POKE PB,0
230 IF DY<0 THEN -MOVE PB,PB-1,21:POKE PB+20,0
240 Y=Y+DY
500 GOTO 130
```

Vzhledem k vysoké rychlosti interpretu jazyka Turbo BASIC je uvedená metoda pohybu spritu po obrazovce postačující i bez nutnosti použití speciálních strojových rutin.

OSS Basic a P/M grafika

Firma OSS patří mezi nekovalitelnější výrobce systémových programů pro počítače Atari XL/XE. Kromě operačního systému DOS XL a jazyka Action! je též autorem tří verzí jazyka Basic, které se liší počtem příkazů a rychlostí zpracování. Jsou to BASIC A+, BASIC XL a nejlepší z nich BASIC XE. Interprety BASIC XL a BASIC XE jsou dostupné pouze na cartridgi, ale BASIC A+ existuje i v kazetové a diskové verzi. Všechny tři interprety obsahují soubor příkazů pro práci se spritovou grafikou. Umožňují tak jednoduchý a rychlý návrh spritů a jejich ovládání.

PMCLR n	vymaže paměť pro hráče číslo n
PMCOLOR n,b,j	nastaví barvu b a jas j pro hráče n (jako příkaz SETCOLOR pro normální grafiku)
PMGRAPHICS r	inicializace P/M grafiky, určí vhodné místo v paměti a nastaví základní registry r=0 zobrazení bez P/M grafiky r=1 normální zobrazení (256 řádků) r=2 redukované zobrazení (128 řádků)
PMWIDTH n,s	nastavuje šířku hráče n (s=1,2 nebo 4)
PMMOVE n,x,dy	pohyb hráče n po obrazovce x je nová vodorovná pozice dy je změna svislé pozice dy>0 pohyb hráče dolů dy<0 pohyb hráče nahoru
PMADR(n)	funkce PMADR vrací počátek paměti hráče n

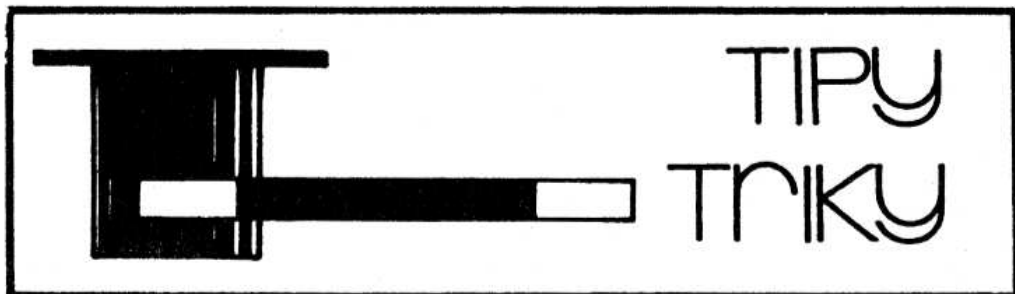
Příkaz PMMOVE n,x,0 je možno psát zkráceně jako PMMOVE n,x.

S pomocí uvedených příkladů již můžeme vytvořit program pro návrh a pohyb spritu 0 ve tvaru ducha ze hry Ghost Chaser v jazyce OSS Basic:

```
10 PMGRAPHICS 1
20 PMCLR 0
30 PB=PMADR(0):X=121:Y=122
40 FOR I=0 TO 20:READ A:POKE PB+Y+I,A:NEXT I
50 DATA 12,24,56,124,84,124,40,56,56,60,126,251,
    185,56,56,60,28,28,14,6,3
60 PMWIDTH 0,1:PMMOVE 0,X
70 PMCOLOR 0,0,14
80 DX=HSTICK:DY=VSTICK
90 IF X+DX<0 OR X+DX>255 THEN DX=0
100 IF Y+DY<0 OR Y+DY>234 THEN DY=0
110 X=X+DX:Y=Y+DY
120 PMMOVE 0,X,DY
130 GOTO 80
```

Tímto programem náš informativní úvod do spritové grafiky končí. Zdaleka jsme v něm nevyčerпали všechny možnosti, které nám spritová grafika poskytuje. Nehovořily jsme o návrhu a použití spritů typu střela, o vícebarevných spritech, rovněž jsme se nezmiňovali o možnosti ovládat sprity v rámci uživatelského přerušení ani o testování vzájemné polohy spritů a kresby pomocí kolizních systémových registrů. Přesto však již nyní umíme navrhnout a ovládat až 4 nezávislé sprity typu hráč (Player). Záleží jen na naší fantazii, kde a jakým způsobem ve svých programech spritovou grafiku použijeme. Hry jsou pouze jednou, ale zdaleka ne jedinou oblastí, kde se dá spritů využít.

(Volně podle seriálu Duszek na Atari z časopisu Bajtek, PLR, ročník 1987 připravil -fis-)



STUDENÝ START POMOCÍ POKE

Po vložení příkazu POKE 580,16 se sděluje počítači, že po stlačení RESET bude vykonán "studený start". Vložená disketa bude znovu načtena a program uložený v počítači bude vymazán.

vados



LISTOVÁNÍ PŘI LADĚNÍ PROGRAMU

Při ladění programů na obrazovce (listování) je nutné pro každé zastavení výpisu stisknout CTRL 1 nebo BREAK. Přitom se však listování zastaví a je nutné začít od počátku.

Dále popsaná krátká rutina tento postup ulehčí. Po napsání dále uvedených šesti řádků na začátek vašeho programu a po spuštění programu příkazem RUN je možno stisknutím SELECT vypsat dalších 10 řádků. Stisknutím BREAK lze program zastavit, pročíst, opravit. Dalším stisknutím SELECT pokračuje listování.

```
1 FOR N=X TO Y STEP 10
2 IF PEEK (53279)<> 5 THEN 2
3 FOR M=0 TO 9
4 LIST N + M
5 NEXT M
6 NEXT N
```

(X = první řádek programu, Y je poslední řádek programu)

vados

ZNOVU BLIKAJÍCÍ KURZOR

V ZAK č. 2/87 byl uveřejněn krátký program na vytvoření blikajícího kurzoru.

```
10 FOR I=1334 TO 1330:READ A:POKE I,A:NEXT I
20 DPOKE 548,1334
30 DATA 3,72,135,20,41,16,74,74,74,141
40 DATA 243,2,104,40,73,133,134
```

Uvedená metoda patří k nejjednodušším, má však jednu vadu. Program se inicializuje nastavením dvou buněk na adresách 548 a 549 (VVBLKDL, VVBLKDH). Tyto buňky totiž definují počáteční adresu přerušovacího programu VBI. Obrazové přerušení nastává jednou za 1/50s a pokud právě v tomto okamžiku měníme hodnotu uvedených buněk, dojde k zhroucení činnosti programu (známému "zatužení"). Přestože tato situace je málo pravděpodobná, případného uživatele může odradit.

V TURBO BASICu lze celou situaci vyřešit velmi jednoduše. Stačí řádek 20 nahradit jediným příkazem:

```
20 DPOKE 548,1664
```

Ve standardním BASICu můžeme využít rutinu operačního systému, která sama počáteční adresu přerušení nastaví. Tato rutina je na adrese 58460 (SETVBV). Celý upravený program má pak tvar:

```
10 FOR I=1334 TO 1330:READ A:POKE I,A:NEXT I
20 FOR I=1331 TO 1330:READ A:POKE I,A:NEXT I
30 I=USR(1331):POKE 54236,34
40 DATA 3,72,135,20,41,16,74,74,74,141
50 DATA 243,2,104,40,76,133,134
60 DATA 104,130,123,132,6,159,7,76,32,229
```

Podtržené hodnoty na řádku 60 určují počáteční adresu přerušovacího programu, tj. odpovídají hodnotám na adresách 548 a 549.

Následující program uvádí složitější, ale komfortnější rutinu pro vytvoření blikajícího kurzoru. Program byl původně napsán v makroassembleru MAC/65 a poté přepsán do standardního BASICu:

```
10 FOR A=1333 TO 1313
20 READ B:POKE A,B:NEXT A
30 A=USR(1334):NEW
40 DATA 133,2,133,3,135,20,41,8,240,2,139,2,141,243,2,76,36,228
50 DATA 104,133,33,133,0,139,255,133,8,139,35,133,2,139,6,133,3
60 DATA 133,2,133,32,139,37,133,80
70 DATA 133,12,141,137,2,139,132,141,139,2,139,149,141,200,2
80 DATA 133,15,141,217,2,159,2,141,213
90 DATA 2,130,0,132,6,139,7,32,32,223,108,12,0
```

Po spuštění programu zjistíte, že kromě blikajícího kurzoru se rovněž změnily některé parametry zobrazení: barvy obrazovky, levý a pravý okraj, rychlost pohybu kurzoru (funkce "autorepeat"). Tyto parametry lze rovněž průběžně měnit změnou hodnot následujících buněk:

POKE 1542,41 — blikající kurzor
POKE 1542,9 — normální kurzor
POKE 1543,x — rychlost blikání
POKE 1572,x — levý okraj obrazovky
POKE 1580,x — intenzita textu

- POKE 1585,x — barva pozadí
- POKE 1590,x — barva rámečku
- POKE 1595,x — prodleva kurzoru (funkce "autorepeat")
- POKE 1600,x — rychlost kurzoru (funkce "autorepeat")

Aby se uvedené změny provedly, je nutno nakonec provést inicializační příkaz:

A=USR(1554)

nebo stisknout klávesu RESET.

Tento program lze využít nejen ve vlastních programech, ale i jako editor pro psaní vlastních programů v jazyce BASIC.

(Podle článku Krupa, M.: Migajacy kursor, Bajtek 2/1988.

připravil —fis—)



KTERÝ BASIC JE NEJLEPŠÍ?

Nový majitel počítače Atari, který se chce seznámit se základy programování, si zákonitě musí položit otázku uvedenou v titulku. Pokud tuto poněkud naivní otázku položí svým zkušenějším kolegům, zjistí jen, jak mnoho se jejich názory na tento problém různí.

Největší část z nich pravděpodobně poradí, aby se Basic vůbec neučil, protože je to jen ztráta času a je to jazyk zastaralý, nevhodný a nemoderní. Hned vám doporučí jiný a daleko lepší jazyk, jako je například Pascal, Forth a podobně. Jediné, co vám tito rádci neporadí je, kde sehnat pro tento jazyk vhodnou českou učebnici a jeho kazetovou verzi pro počítač Atari.

Další skupinka sice vezme Basic na milost, současně vám však doporučí některou z jeho "exotických" odrůd jako např. Microsoft Basic, Basic XL a podobně. Zároveň vám též vysvětlí, že Atari Basic, který máte ve svém počítači je zastaralý, pomalý, nemoderní, zkrátka nejhorší.

Atari Basic je pro začátečníky minimálně stejně vhodný jako jiné základní verze jazyka Basic používané v počítačích Sinclair, Commodore a podobných. Takže daleko lepší, než shánět nějaký jiný zázračný jazyk, je naučit se dobře ten, který máme snadno k dispozici. Ostatně všechny solidní úpravy jazyka Basic pro počítače Atari vycházejí právě z jazyka Atari Basic a většina příkazů těchto jazyků je s ním shodná. Programy zapsané v Atari Basicu jdou většinou bez problémů natáhnout do Turbo Basicu, dojde tím pouze k jejich zrychlení.

Horší problém ovšem nastane, pokud začnete shánět nějakou učebnici jazyka Basic. V prodejnách je sice ještě k dostání publikace "Programování v jazyce BASIC", kterou vydalo SNTL. Ta však popisuje poměrně složitou verzi jazyka Basic. Výklad a terminologie jsou navíc natolik odborné, že pokud nejste absolventem vysoké školy, jsou pro vás mnohé formulace dosti nejasné.

Učebnice jazyka Atari Basic nebyla zatím vydána. Návod dodávaný spolu s počítačem je až příliš stručný a předpokládá již určité znalosti v programování. Těchto důvodů doporučuji pro začátečníky přílohu VTM z roku 1987, která se jmenuje "Domluve se s počítačem". Zde je pod názvem "Basic pro nedouky" otištěn výborný kurs Jiřího

Fraňka. Je psán nejen zajímavě, ale hlavně dostatečně srozumitelně i pro neaprosté začátečníky. Použitý dialekt Basic je sice poněkud odlišný od Atari Basicu, ale to zase dává možnost vlastní modifikace otištěných ukázkových programů při jejich použití v počítačích Atari.

Připravil Tomáš Čermák, Praha

Poznámka redakce: Všem našim čtenářům, a to nejen začátečníkům, můžeme slíbit, že se již připravuje příloha ZAK pod názvem "Turbo BASIC 1.5 XL", která je tolik žádanou učebnicí jazyka Turbo BASIC a je velmi vhodná i pro začátečníky.

UŽIVATELSKÉ PROGRAMY

VÝUKOVÉ PROGRAMY pro ATARI

ing. Miloš Adam, Krásné Údolí

Jako velmi vhodný způsob využití mikropočítače (a zejm. ATARI) se ukazuje jeho zapojení do výuky dětí. Znamená to ale mít k dispozici výukové programy. Pokud se program připraví tak, aby děti zaujal, je vyhráno. Děti výpočetní techniku přímo zbožňují a dobrý program dokáže, že se dítě doslova těší na test z matematiky, českého jazyka, přírodopisu a jiných předmětů, které mohou být za jiných okolností nepříjemným oříškem. Jde totiž o to, že se přirozeným způsobem daří naplňovat Komenského zásadu "škola hrou". Aniž by si to děti uvědomovaly, učí se při hře. Zákonitě tím roste výkonnost, výsledky ve škole a snad i vztah k počítačům. Nabyt jsem dokonce poznání, že počítač plně nahradí tradiční rákosku.

Za zmínku stojí kroužky výpočetní techniky, které se vytváří na některých základních školách. Po několikaleté praxi si dovoluji konstatovat, že tyto kroužky by si neměly klást za cíl vychovávat geniální programátory, ale vypěstovat v dětech vztah k výpočetní technice. Právě to bude nesmírně cenná deviza pro jejich další cestu životem v následujících letech. V kroužcích výpočetní techniky se kromě výuky některého základního programovacího jazyka (KAREL, LOGO, PASCAL, ale samozřejmě i BASIC) musí (!) nechat dětem prostor pro hry, přičemž ale jejich výběr závisí na lektorovi. A tady je možné namísto všelijakých střílejších panáček nabídnout něco, co souvisí s výukou, ale nepřipomíná to školu.

V této souvislosti je třeba upozornit na přiměřenost požadavků na děti. Příliš velké nároky a nevhodné programy postaví dítě proti počítači a ne k němu, jak si přejeme. Sám pracuji s dětmi ve věku 8 až 12 let a tak vidím, že KAREL je pro začátek asi nejlepší. Problém ale spočívá v tom, že k němu nelze sehnat metodiku. Totéž platí o stejně vynikajícím programovacím jazyku pro děti — LOGO.

Některé děti mají dobrý vztah k Turbo Basicu. A zde musím tolik pomlouvaný Turbo Basic hájit. Dítě odkojené KARLEM a trošku k tomu vedené programuje strukturovaně a elegantně i v této verzi Basicu.

O hráč jako takových nemá cenu mluvit. Je jich tolik, že se vždy dá nějaký program sehnat. Horší je to ale s programy výukovými. Snad každý atarista—rodič nějaký program pro své děti má, možná jej i sám stvořil. Hlavní problém spočívá v tom, že tak jak dětem pokračuje

výuka ve škole a jak zvládají novou učební látku, tak se jim včera nový program zdá zítřka starý a vyžadují další. Chudáci programátoři. Existují sice různé "matematiky", "zeměpisy", "češtiny" apod. Některé jsou opravdu krásné. Přesto jsou ale jen "dočasně zajímavé". Bohužel ale, právě kvalitní programy jsou většinou dobře utajené. To je rozhodně škoda.

V zájmu našich dětí nabízíme vytvoření jakéhosi centra pro šíření učebních programů na ATARI. To, co jsme prozatím vytvořili na Základní škole v Krásném Údolí, to jsme poskytli pražským kolegům, kteří programy natočili na referenční kazety (zejm. č. T-31).

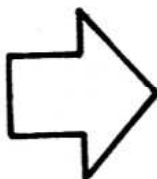
Pro lepší vysvětlení: používáme univerzální program (údajně přišel z PLR), do kterého lze vložit (jako DATA) jakékoliv otázky a odpovědi.

Naše představa spočívá v tom, že bychom soustředovali všechny učební programy a následně je podle požadavků distribuovali na kluby a kroužky. Abychom tuto službu mohli zahájit, obracím se na čtenáře pražského Zpravodaje AK s prosbou: Pokud jste vlastníky pěkného programu, ozvěte se, případně pošlete nahrávku. V nabídce uveďte co nejpřesnější údaje o programu. Doufám, že nikdo nebude za poskytnutý program pro děti očekávat finanční odměnu. V dalších číslech ZAK Praha pak budeme po dohodě s redakcí zveřejňovat seznam získaných programů, resp. těch, které jsou k dispozici.

Věřím, že nabízená služba umožní rozšířit tolik potřebné účelově zaměřené výukové programy.

☆☆☆

Nabídky programů je možno zaslat na adresu:



ATARI KLUB Praha

pošt. přihrádka 51

100 00 Praha 10

Heslo: VÝUKOVÉ PROGRAMY

nebo Ing. Miloš Adam

Nemochovice 161

683 32 p. Brankovice

UNIVERZÁLNÍ VÝUKOVÝ PROGRAM

V článku ing. Miloše Adama z Krásného Údolí je zmínka o používání univerzálního programu pro přípravu výukových programů. Požádali jsme kolegy z Krásného Údolí o jeho poskytnutí. Jsme toho názoru, že tímto způsobem můžeme napomoci k urychlenému "zpracování" učebnic našich školáků podle již uvedeného hesla "škola hrou".

Autorem programu je Tomáš Nabzdýk z PLR, pro naše podmínky jej upravil F. Kolář z Krásného Údolí.

Jak si připravíme vlastní výukový program? Nejdříve do počítače zavedeme TURBO BASIC 1.5 XL (nebo verzi TBX 2.0 DOS). Pak pečlivě opišeme řádky 10 až 750 a 2000 až 3107 dále uvedeného výpisu. Opsaný program si uložíme (pro přístě) na magnetické médium (kazetu nebo disketu). Nyní si vezmeme učebnici a začneme vymýšlet otázky a odpovědi. Podle jednoduchého algoritmu pak programujeme. Příklad je uveden v řádcích 1000 až 1032. Při našem programování budeme potřebovat vždy po třech programovacích řádcích. Na první napíšeme heslo, na druhý otázku a na třetí tři odpovědi (jednu správnou a dvě vymyšlené). Na konec tohoto programovacího řádku se uvede číslice znamenající pozici, na které se nachází správná odpověď. Přirozeně, že pozici nepravidelně měníme. Do programu lze vložit 100 otázek.

—red.—

```

0 REM *****
1 REM *****
2 REM *** Programowa! ***
3 REM *** Tomasz Habzdyk ***
4 REM *** ul.Kochanowskiego ***
5 REM *** 5/13 ***
6 REM *** 72-420 DZIWOW 1. ***
7 REM *** luty 1987 ***
8 REM *** UPRAWA ***
9 REM ***** *** Z.S. KRASNE UDOLI ***
10 POKE 566,158
20 DIM RET$(1),A$(100),A1$(100),A2$(100),A3$(100),A4$(100),A5$(5),O$(10)
25 DIM OCENA$(20)
30 GRAPHICS 1+16:POKE 712,255:POKE 708,0
35 OBRAZ=PEEK(559):POKE 559,0
40 POSITION 2,2: ? #6;"ZS KRASNE UDOLI":POSITION 4,5: ? #6;"PREDSTAVUJE"
42 FOR I=1 TO 17:POSITION 1,10: ? #6;"*":NEXT I
45 POSITION 4,14: ? #6;"univerzalni":POSITION 1,18: ? #6;"zkusebni program."
49 POKE 553,OBRAZ
50 RESTORE 3100:REM UVODNI MELODIE
55 PAUSE 3:READ A,B:IF A=-1 THEN 70:SOUND 0,0,0,0
60 SOUND 0,A,10,15:FOR I=0 TO (B+1)*5:NEXT I:GOTO 55
70 GRAPHICS 0:POKE 710,0:POKE 709,255
73 GOSUB 2000:REM INFORMACE
75 KOLEJKA=0:PUNKTY=0:OCENA$=""
80 AA=INT(RND(0)*1000):AA=AA/10:AA=INT(AA):AA=AA*10
85 IF AA>30 THEN 80
87 IF KOLEJKA=3 THEN 500
90 RESTORE 1000+AA
100 READ A$,A1$,A2$,A3$,A4$,A5$: ? "":POSITION 18-INT(LEN(A$)/2),0: ? A$
110 POSITION 2,3: ? A1$
115 ? : ? ""
120 POSITION 2,9: ? "C1] ";A2$
125 POSITION 2,11: ? "C2] ";A3$
130 POSITION 2,13: ? "C3] ";A4$
131 ? : ? ""
135 POSITION 5,19: ? "TVOJE ODPOVED(1,2 nebo 3) ";:INPUT O$
145 IF A5$=O$ THEN 300
147 IF O$="" THEN 250
150 IF A5$<>O$ AND O$<>"1" AND O$<>"2" AND O$<>"3" THEN 250
153 POSITION 1,19: ? ""
160 KOLEJKA=KOLEJKA+1
165 POKE 712,255:POSITION 9,21: ? "UDELAJ JSI CHYBU (<";A5$;")"
167 GOSUB 180
168 POKE 712,0
170 POSITION 7,23: ? "STISKNI RETURN A POKRACUJ";:INPUT RET$:POKE 712,0
175 GOTO 80
180 RESTORE 3000:REM MELODIE U CHYBY
185 PAUSE 3:READ A,B:IF A=-1 THEN SOUND 0,0,0,0:RETURN
187 SOUND 0,A,10,15:FOR I=0 TO B*3:NEXT I:GOTO 185
250 REM SPATNA KLAVESA
253 POSITION 1,19: ? "" :POKE 752,2
255 POKE 712,255:POSITION 8,19: ? "STISKL JSI SPATNOU KLAVESU";" "," "," "
260 ? : ? " STISKNI RETURN A POKRACUJ";:POKE 752,2:INPUT RET$:POKE 710,0:POK
E 712,0:REM FOKE 709,0
270 POSITION 1,19: ? "" :POSITION 1,20: ? "" :GOTO 135
300 REM SPRAVNA ODPOVED
310 POSITION 6,21: ? "DOBRE,odpoved je spravna.":RESTORE 3100
320 KOLEJKA=KOLEJKA+1:PUNKTY=PUNKTY+1
330 PAUSE 3:READ A,B:IF B=-1 THEN SOUND 0,0,0,0:GOTO 80
340 SOUND 0,A,10,15:FOR I=0 TO 2*B:NEXT I:GOTO 330
500 IF PUNKTY/KOLEJKA>=0.95 THEN OCENA$="VYBORNE !!!":GOTO 540

```


TECHNICKÉ novinky

TURBO MOS

Vladimír Jurčo, Kristian Velikov,
Rudolf Apolen, Nové Zámky

V oblasti vnějších pamětí nejnižší cenové třídy stále dominují kazetopáskové paměti pracující s dostupnými magnetofony. Disketová jednotka je ještě stále pro většinu našich majitelů domácích počítačů nedostupná. I když magnetofon nemůže nikdy plně nahradit práci disketové jednotky, může se k ní svými vlastnostmi alespoň částečně přiblížit. Na základě těchto úvah vznikl program **TURBO MOS**.

Program **TURBO MOS** je rozšířením původního operačního systému, který při ovládání datového magnetofonu nevyhniká rychlosti ani spolehlivosti záznamu. **TURBO MOS** umožňuje podstatné zrychlení a zefektivnění programátorských prací s cílem potlačit co nejvíce nedostatky kazetového magnetofonu.

Tento doplněk k operačnímu systému může programátor využít ve svých vlastních programech, ať již v **BASICu**, nebo v některém jiném programovacím jazyku.

V systému **TURBO MOS** není nutné programy upravovat, stačí nahrát příslušný program standardní rychlostí 600 Bd kopírovacím programem a poté jej zpět přehrát v rychlosti záznamu 4000 Bd a více.

Ve všech programech lze využívat ramdisk a zrychlený záznam pro práci s daty. Výhodou je vysoká spolehlivost záznamu, překonávající všechny známé způsoby záznamu na počítačích Atari. Pod **TURBO MOSem** by měly pracovat všechny uživatelské kazetové a disketové programy, které nemají softwarovou ochranu a které přímo nespolupracují s disketou.

Systém se skládá ze tří částí:

- **RAMDISK**
- **TURBO"DNM**
- **MONITOR**

Systém poskytuje uživateli různé služby, jako výpis adresáře ramdisku, výpis názvu programu na pásce, přímé nahrávání souborů do ramdisku, ukládání souborů z ramdisku na zvolené periferní zařízení, zavedení programu ve strojovém kódu atp.

Program **TURBO MOS** má dvě verze:
C—MOS a **D—MOS**.

Verze C—MOS:

Tato verze je kazetovou verzí obsahující všechny výše uvedené vlastnosti. Je určena pro kazetové programy a pro disketové programy pracující také s magnetofonem.

Verze D—MOS:

Pod tímto systémem pracují uživatelské programy, které pracují s verzí **C—MOS**, **D—MOS** navíc do určité míry simuluje disketovou jednotku a proto pod ním pracuje velká část programů, které jsou výlučně disketové. **D—MOS** obsahuje pouze část disketových operací, neboť všechny operace pro práci s disketovou jednotkou nelze na magnetofon realizovat. Práce s magnetofonem bude náročnější než práce s disketovou jednotkou, protože magnetofon je třeba manuálně obsluhovat. I přesto bude ale **D—MOS** užitečným pomocníkem pro ty majitele počítačů Atari, kteří nevládní disketovou jednotku, protože jim zpřístupní další programy, které doposud pracovaly pouze s disketou. Pod tímto systémem pracuje například disketová verze **KYAN PASCALu**, **DESIGN MASTER**, **ATARI WRITER**, **PAPER CLIP**, **STARTEXTER** a mnohé další disketové programy.



KŘÍŽEK, HVĚZDIČKA A TISKAŘSKÁ TECHNIKA

Dovoluji si ještě jednou se vrátit k nešťastným křížkům a hvězdičkám, o kterých píše v ZAK 3/88 Vladimír Čech. Mám za to, že redakce ZAK již dostatečně objasnila, jak a proč došlo (dochází) k záměnám křížku za hvězdičku. Přesto však raději ještě jednou odmítám tvrzení - cituji - že "je jasné, že chyba je způsobena spíše nedostatky v tiskařské technice, která na tento znak zásadně "zapomíná" a způsobuje problémy při vydávání veškeré technické a matematické literatury".

V tiskařské technice jako takové to není. Chyba se stala při korektuře, i když připouštím, že zařízení, na kterém pořizujeme fotosazbu předlohy pro tisk Zpravodaje AK nemá ve své znakové sadě hvězdičku.

Co se týká onoho nevhodně voleného termínu „tiskařská technika“, považují tuto možnost za vítanou příležitost seznámit alespoň ve zkratce čtenáře ZAK se současným trendem (i když pouze světovým, nikoliv československým) rozvoje fotosazby. Podotýkám, že nemalou roli zde hrají i osobní počítače, kterým je připisována v budoucnu jedna z hlavních úloh. Autor svůj příspěvek (nebo i případně celou knihu) zpracuje na počítači a do nakladatelství předává pouze příslušný počet disket, které jsou v nutném počtu přepokopovány, aby na vlastním textu mohli

současné pracovat korektor, recenzent, textový redaktor a další. Po jejich odborném zpracování se vyznačené zásahy sjednotí na jednu základní disketu, která se předá do tiskárny, kde je pomocí periferie převedena na vstupní data příslušného typu fotosázecího zařízení, které na základě sázecího programu provede vlastní sazbu. Operátor-sazeč v podstatě neprovádí sázení textu, pouze zvolí vhodný program. Tímto způsobem se zaručí autorská, respektive nakladatelská správnost. Z hlediska sazby matematiky či různých technických symbolů, které představovaly určitá technická úskalí, není dnes již problémem uložit do paměti fotosazby specifické znaky pro různá technická odvětví včetně matematiky, chemie, fyziky, případně prvků výpočetní techniky, které lze vždy v případě potřeby vyvolat nebo napevno zaprogramovat. Důkazem pro toto tvrzení může být například i fotosázecí systém SCAENTEXT 2000, který je schopen provádět zhotovení stránkových montáží i pro tak náročné tiskopisy jakými jsou například formuláře pro optické snímání dat nebo lineární sazba čárového kódu EAN. Z hlediska čtenáře je také v současné době zaváděno nové upravování základních matematických vzorců, které v minulosti na horké sazbe byly skutečně v některých případech tvrdým oříškem pro sazeče. V podstatě se jedná o změnu řazení vzorců do jedné řádky s tím, že algoritmus vzorce nesmí být pozměněn.

Zdeněk Šebesta, vedoucí výroby
reprografického provozu PORS 001

TŘI KNIHY Z PLR

Polská počítačová literatura je mezi československými počítačovými odborníky i amatéry poměrně dobře známa. Mnozí z nich jsou pravidelnými čtenáři časopisů "Bajtek" a "Komputer" a oceňují jejich obsahovou i odbornou kvalitu. Tyto dva časopisy jsou také jedinými specializovanými periodiky zabývajícími se problematikou osobních a domácích mikropočítačů, které si může prostřednictvím PNS objednat i jednotlivec (není však samozřejmě zaručeno, že jeho objednávka bude kladně vyřízena).

Návštěvníci prodejny Polského kulturního střediska v Praze byli v letním období jistě příjemně překvapeni i nabídkou něko-

lika odborných titulů z oblasti počítačové techniky. Alespoň v jedné prodejně v ČSSR si tedy mohli zájemci zakoupit za rozumné ceny publikace, které na našem běžném vnitřním trhu tak citelně chybí. Zkusme si alespoň tři z nich představit trochu blíže:

Ataristy především upoutávala kniha **Piotra Adamczewského**, kterou pod názvem **INSTRUKCJA OBSŁUGI MIKROKOMPUTERA ATARI 800 XL** vydalo varšavské SOETO v edici Informatyka. V této edici se objevují především uživatelské příručky k nejrozšířenějším mikropočítačům, manuály k jazykům a podobně. Výše uvedená publikace nahrazuje vlastně základní uživatelskou příručku k mikropočítači Atari 800 XL. Pokud bychom ji ale chtěli srovnat například s tuzemskou literaturou, vydanou našimi Atari kluby, pak vlastně nepřináší žádné nové informace. Přidáme-li skriptovou formu, poměrně nekvalitní papír i tisk a značně vysokou cenu (50,— Kčs), není divu, že mnozí zájemci knihu pouze prolistovali a vrátili zpět na pult. To, že byly nakonec prodány všechny dovezené výtisky, svědčí spíše o tom, že ne každý majitel počítače Atari má možnost získat tuzemské klubové publikace, zejména ty starší (například soubor příruček, který vydal Atari klub Olomouc u příležitosti 1. setkání ataristů ČSSR v Ti-mačích před dvěma roky). Pro úplnost si uvedme alespoň stručný obsah polské publikace: základní údaje o práci s počítačem, popis interpretu jazyka Atari Basic, stručný popis a přehled instrukcí mikroprocesoru 6502, přílohy.

Zajímavější po stránce obsahové i provedením (dokonce barevné obrázky) byla dvoudílná příručka **Stanislava Waligorského LOGO NA SINCLAIR SPECTRUM**, která vyšla v nové počítačové edici 1+1=10. Autor příručky je známým propagátorem a popularizátorem jazyka Logo v Polsku a je rovněž autorem jeho polské verze. Přestože z názvu příručky vyplývá, že jde zejména o publikaci orientovanou na počítače Sinclair ZX Spectrum, sám autor uvádí, že ji mohou s úspěchem využít i majitelé počítačů Amstrad/Schneider a Atari. Příručka je rozdělena na

dvě samostatné části - pro začátečníky (díl 1) a pro pokročilé (díl 2). Popisuje principy programování v jazyce LCSI Logo, který je implementován na všech třech uvedených typech počítačů a jehož jednotlivé verze se liší pouze nepatrně. Prakticky pouze s výjimkou kapitoly, popisující zavedení jazyka Logo do počítače, lze příručku použít i na počítače Atari XL/XE. Obě části publikace obsahují množství názorných příkladů včetně obrázků. Některé z nich byly převzaty z knihy S. Paperta "Mindstorms - children, computers, and powerful ideas", která je ve světě považována za jednu z nejlepších publikací zabývajících se aplikacemi jazyka Logo ve výuce dětí. Obě části příručky Stanisława Waligorského lze tedy našim čtenářům vřele doporučit.

Třetí poměrně zajímavou publikací je kniha **Jana Bieleckého SAM NA SAM Z JAZYKIEM C**. Jedná se o úvodní příručku programování v jazyce Hisoft C, což je verze jazyka C na počítače Sinclair ZX Spectrum. Příručka je poměrně více specializována na tento typ počítače, přesto však lze základní myšlenky a většinu příkladů použít i pro jazyk Deep Blue C, který je implementován na mikropočítačích Atari. Oba jazyky, respektive verze, se liší zejména obslužným systémem, tj. příkazy pro editaci zdrojového textu a překlad, a také knihovnou funkcí. Příručku lze proto využít zejména pro seznámení s jazykem a při vytváření prvních, jednoduchých programů. Pro vážnější práci s jazykem C lze doporučit učebnici tohoto jazyka, kterou připravují nakladatelství ALFA ve spolupráci se SNTL a která by měla vyjít v nejbližší době.

-fis-



Opravný listek pro přílohu ZAK I—1

Když jsme připravovali z vytvořených tiskových předloh dotisk přílohy ZAK I—1, zapomněli jsme do ní zařadit opravný listek, který vznikl po prvním vydání. Pro nové majitele této cenné publikace tak činíme nyní.

Dále uvedené opravy Vám možná ušetří čas při studování této publikace. Kdo chce ušetřit ještě více času, nechť se zajímá o kazetu, která byla "natukána" ještě před vydáním publikace; obsahuje všech 51 demonstračních programů a šíří se z našeho kroužku Královka. Snad se obvyklou formou přes burzy rozšíří mezi potřebné členy. Autoři publikace /kteří na kazetě spolupracovali/ ani autoři kazety ji přirozeně nemohou nakopírovat všem. Dále uvedené opravy vyplynuly právě ze snahy připravit čtenářům publikace kazetu. Publikace /resp. matrice pro ofsetový tisk/ totiž vznikla na velkém textovém zařízení, kde nebylo možné "natukané" programy zkontrolovat. To jsme dělali až v době, kdy už běžela výroba publikace.

Redakce

s.3 PRG PŘEVOD DEC - BIN

200 IF PEEK(764)=35 THEN POKE 764,255:END

s.6 PRG BIN NA HEX

110 FOR J=5 TO 8

s.16 PRG PODPROGRAM PRO ZAJIŠTĚNÍ BREAK

nutné doplnit:

20 ? "A";

30 GOTO 20

s.16 PRG VNITŘNÍ HODINY

30 T=INT((PEEK(18)*65536+PEEK(19)*256+PEEK(20))/50)

s.22 PRG OCHRANA OBRAZOVKY

40 T=INT((PEEK(19)*256+PEEK(20))/50)

220 POS.4,3:TM;" "

1 mezera

230 POS.6,3:?" "

bez mezer ":"

240 POS.7,3:7S;" "

1 mezera

260 POS.0,16:?" /40x/" 40x podtržené

-18 ř.zdola: 10- B má přebrat bity z ATTRACTu. Pole je ...

s.25 PRG VLIV LOCATE NA KURZOR

50 ? PEEK(85); " , "; PEEK(84),

s.28 PRG SKOKOVA ZMENA GRAFIKY

210 vypustit a dále doplnit:

255 IF D=7 THEN 300

320 FOR J=2 TO 7

360 FOR J=6 TO 3 STEP -1

s.32 PRG OCHRANA PŘED LIST

40 ?F,G,H

Program je pro disketovou verzi. Pro kazetovou verzi je nutné opravit řádky 100, 110 a 120, tj. místo "D:SECRETE" napsat "C:". Na připravené kazetě je již úprava provedena.

s.33 PRG TABULKA PROMĚNNÝCH

Obsah řádků 200 a 300 je nutné prohodit.

s.36 PRG VYHOZENÍ PAMĚTI

- paměť vyhaszuje pouze v disketové verzi.

s.37 PRG STARTOVACÍ ADRESY/ČÍSLO BASIC ŘÁDKY

- v řádku 130 za T /na konci/ již není žádný znak.

s.39 PRG ARD137A

- vhodnější název prg /viz 3.ř./ je ROZRUŠENÍ ČÍSEL ŘÁDKŮ.

s.41 PRG STRING MÍSTO ARRAY

50 ? : ? Z\$ (1,2); ". "; Z\$ (3,4); ". "; Z\$ (5,8); ": "; Z\$ (9,14)

s.44 PRG ERROR - OCHRANA

100 DIM M(6,6): FOR I=0 TO 6: FOR J=0 TO 6: M(I,J)=0:
NEXT J: NEXT I

s.62 PRG OTOČENÉ ZNAKY

Pozn: obsah řádků 70 a 80 je správný. Řádky jsou v publika-
ci jen prohozeny.

Pro ty, kteří mají hůře čitelný výpis tohoto programu:

50 ?#6; J; " ";

s.65 PRG XIO 18 - FILL

140 XIO, #6, 0, 0, "S:"

s.79 PRG VÝCHOD SLUNCE

40 FOR X=152 TO 159: PLOT X,0:

DRAWTO X,95: NEXT X

50 FOR Y=0 TO 70: READ R: PLOT 6,Y:

DRAWTO R,Y: NEXT Y

Pozn: na konci programu se doporučuje doplnit:

700 REM POZOR

701 REM NUTNO OVLADAT JOYSTICKEM

s.98,99 PRG ZAPÍNÁNÍ/VYPÍNÁNÍ TEXT. OKNA

100 OPEN#1,4,0,"K:": GET#1,T: CLOSE#1

340 COLOR 1: PLOT 0,0, DRAWTO 39,0

s.102 - 105 PRG KURZIVA-DATA

Pozn: Po RUN a objevení se kurzoru vlevo nahoře je nutné
trpělivě vyčkat asi 10 sec. Pak se objeví nápis,
který se přepíná z normálního do kurzivního písma.
Jestliže je stisknuto tlačítko BREAK v okamžiku, kdy
je nápis v kurzivě, lze pomocí LIST spatřit celý
program v kurzivě.

Opravte:

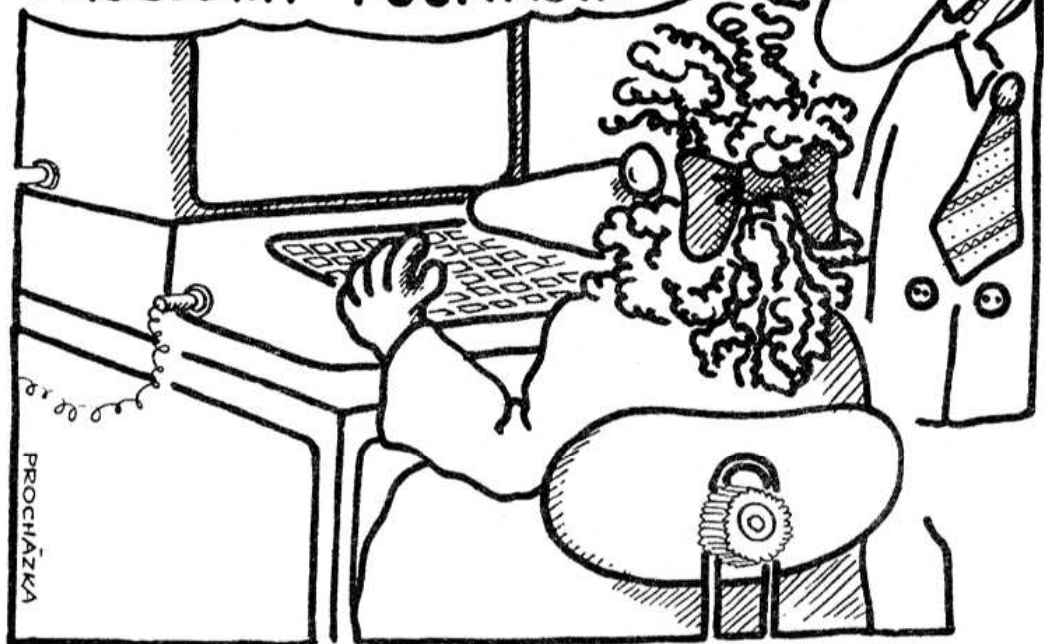
120 ?"TAK VYPADA NOVE ZOBRAZENI,"

Další překlepy aj. písařské chyby v běžném textu již nejsou
opravovány. Předpokládáme, že čtenář promine i absenci čárek,
háčků aj. znaků běžných v češtině. Jen snaha o rychlé vydání
zpracovaného rukopisu nás vedla k této úpravě.

Žádáme laskavě čtenáře, který by v publikaci našel pod-
statnou chybu, aby na to upozornil naši redakci. Děkujeme.

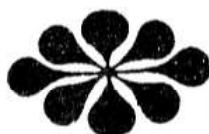
Revizi v publikaci uvedených programů a vytvoření kazety zajis-
tili ve spolupráci s autorem Milošem Bayerem ing. Václav Fried-
rich a Dr. Jan Hlaváček.

SLEČNO, SKOČTE SI LASKAVĚ
NA SEXUOLOGII A
NEZATĚŽUJTE SVÝMI
PROBLÉMY POČÍTAČ...!



žádáme všechny členy a hosty
ATARI KLUBU Praha, aby v co
nejkratší době uhradili poštovní
složenkou svazarmovský a klubový
příspěvek pro rok 1989. Maximální
termín pro zplacení je 31.1.1989.

Děkujeme.



Publikované zo súhlasom - vid' Prohlášení představitelů AK Praha.