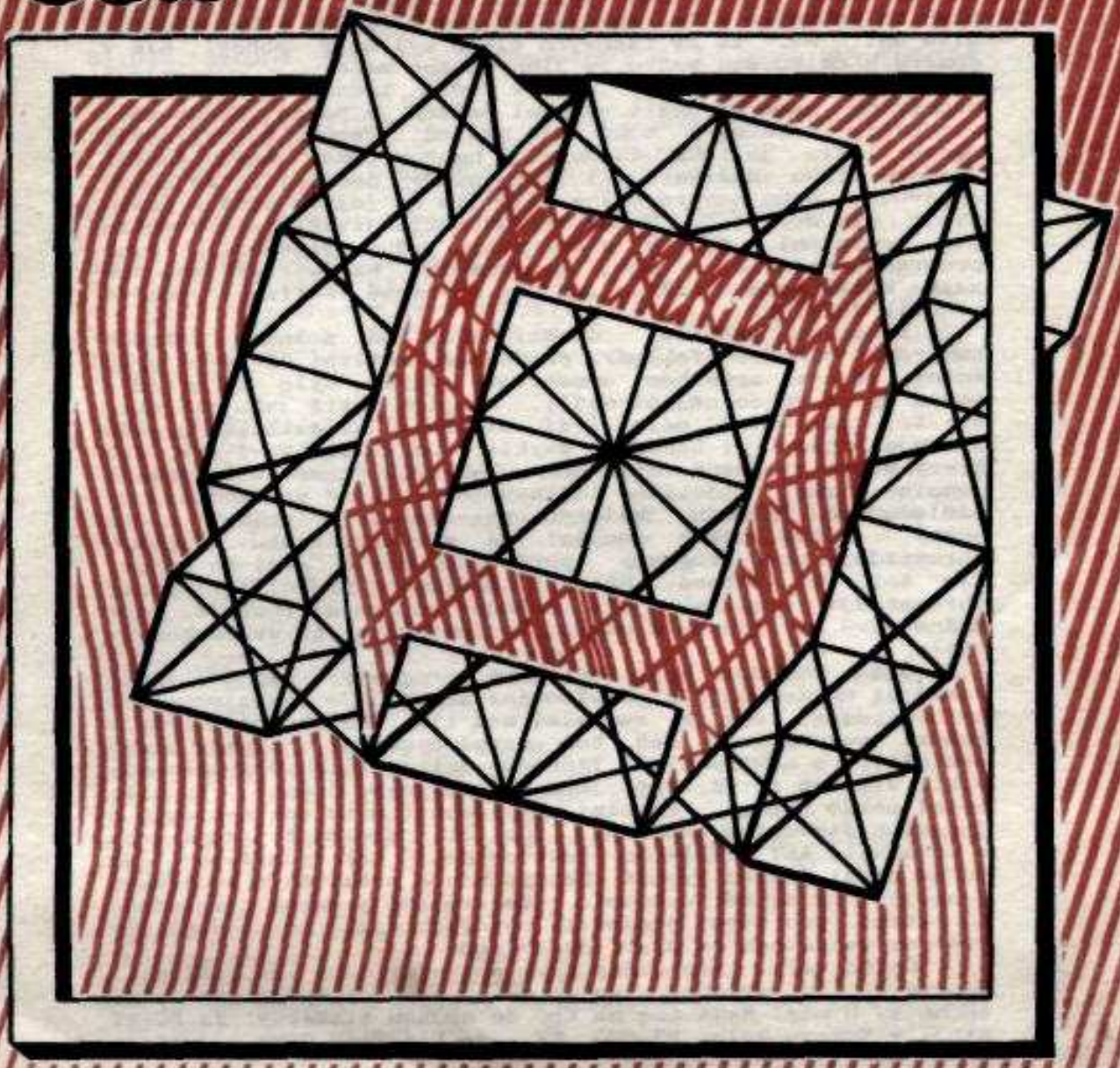
SVAZARM

602

SPECTRUM

1

89



CO MATE ZA NOVÉ PROGRAMY ?

Od té doby, co jsem se stal majitelem osobního počítače, slýchám tuhle otázku - velmi často. Sám jsem majitelem mikropočítače od roku 1981 a myslím si, že znám SINCLAIR klub 602. ZO poměrně dobře. Jeho členem jsem totiž od téhož roku. Dříve se sice jmenoval jinak, ale to nevadí. Za ta léta se toho změnilo víc. Lidé přicházeli a odcházeli a někteří z nich se pokusili zanechat těm, co přijdou po nich, něco jako odkaz. Po téměř osmi letech zbyl, pokud vím jenom jeden člověk, který dodnes aktivně pracuje, totiž Jirka Janda. My dva jsme snad poslední, kdo "přežili" z doby ZX-81. Lidí se v klubu vystřídalo mnoho, ale jen málo z nich se pokusilo o to, aby jejich práce (pokud vůbec nějaká) žila dál. Ne, že by takových členů byl malý počet, ale v poměru k celkovému počtu je jich směšně málo.
Vím, časy se mění a dřív byla trochu jiná doba. V začátcích jsme se znali všichni jménem a pokud někdo něco vymyslel, ostatní z toho měli užitek. Nebudu tu vypisovat jména lidí, jejichž duchovní i hmotný odkaz mohou dnešní členové klubu využívat. Ti lidé nikdy nečekali, že budou oslavováni nebo honorováni a také jim za jejich hodiny práce nikdo nic neplatil. Možná to bude znít divně, ale stačilo jim pomyšlení, že ve svém oboru něco zanechali. Duch programátora žije v jeho programech, hardwarový fanoušek zase kolem sebe viděl, jak jím navržené zařízení používá několik dalších.
Doba se změnila, dnešní klub má možnost dokonce honorovat články, překlady, přednášky, aktivní členy by bylo možné různým způsobem odměňovat. Říkám "bylo by možné", pokud by se vůbec nějací našli. Je jich totiž jako šafránu. Na to, že velmi málo lidí ve svém zaměstnání opravdu pracuje, jsme si bohužel zvykli. Je to zajímavý jev, pro který existují různá zdůvodnění, jak se dočteme na stránkách denního tisku. Podstatně zajímavější a méně prozkoumaný je následující fenomén. Málokdo pracuje i v oblasti svého hobby. Pokud někdo dochází na schůzky klubu a pouze shromažďuje nové programy (povětšinou hry), není to práce, ale ani fér zábava. Někteří jedinci nejen, že pouze shromažďují programy, ale dokonce se velmi rozčílí, když žádné nové hry nejsou k mání. To není vtip, to se mnohokrát stalo.
Lidé shánějí hry, užitkové programy, manuály apod. Pokud si někdo půjčí k okopírování manuál, je spokojen a ani ho nenapadne, aby se zamyslel nad tím, kolik hodin trvalo komusi neznámému, než ten manuál sehnal, napsal nebo přeložil, o naklepání do TASWORDu ani nemluvě. 0 programech platí vlastně totéž, ať jsou firemní nebo domácí. Ty naše musel někdo napsat, ty cizí zase sehnat a to ani nemluvím o úpravách firemních programů. Ono totiž přeložit taková menu v originále ART STUDIA do češtiny není tak jednoduché, jak to na první pohled vypadá. Každý, kdo s počeštěným programem pracuje, by si měl uvědomit, že ten kdo program překládal, ho patrně pro sebe přeložit nepotřeboval, neboť anglicky uměl. 0 manuálech to platí dvojnásob.
Další velmi pěstovaný zvyk poslední doby je nadávání. Nadává se na všechno, co se týká situace kolem výpočetní techniky u nás. Nadáváme na to, že nejsou tiskárny, že PC-AT stojí v Tuzexu kolem 22.000 TK, že Didaktik GAMA nefunguje

apod. Netvrdím, že situace kolem hardwaru je u nás v pořádku. Naopak, dokud nebudu moci jít do obchodu a koupit si např. IBM PC nebo tiskárnu srovnatelnou s Epson FX-85 ve stejném poměru plat/cena jako v NSR, budu tvrdit, že situace je katastrofální. Bylo by však načase začít i o programech a dokumentaci uvažovat jako o zboží, se všemi důsledky. Jen ať si každý spočítá, kolik peněz by utratil např. v Anglii za programy, které si nahrál v klubu. Zkuste si to sami: Spočítejte si programy na svých 20ti nebo 30ti kazetách a vynásobte je průměrnou cenou 8 liber za jeden program. Číslo, které vám vyjde, si přepočítejte na kurs libry v bonech a nestačíte se divit, jaký máte majetek. A všechny ty programy musel někdo koupit, přivézt, okopírovat, napsat atd. U majitelů IBM PC je to ještě markantnější, kolik z nich si uvědomí, že jenom databáze a textový procesor by stály kolem 4.000 DM.
Doba zřejmě od počátečního fandovství dospěla do stádia, kdy je nutné postavit jiná měřítka hodnot. Zkuste si spočítat, kolik programů by si mohl každý koupit v Tuzexu (pokud by je dovážel) za své klubové příspěvky. Bon můžete počítat jako 5 Kčs. Rozdíl těch dvou částek je značný. Od této sumy si odečtěte podle vlastního uvážení, kolik práce jste do rozvoje klubu věnovali. Ten poslední rozdíl je to co jste získali zadarmo, z výsledků práce někoho jiného. Nad tou částkou se zamyslete a pokud se snad trochu začervenáte, je to jen dobře.
Ale nepřemýšlejte nad tím moc dlouho, začněte raději něco dělat. Ideální by bylo pracovat tak, aby byl rozdíl za rok o něco menší, ale nestačí jenom v klubu oznámit "Já můžu překládat" a čekat až vás o to někdo požádá a potom se vymluvit na nedostatek času. Mnohem lepší přístup je sehnat zahraniční informace, přeložit je a přinést do klubu nebo poslat do zpravodaje, při které začíná pracovat redakce manuálů. V této chvíli asi někteří z vás kroutí hlavou, kde sehnat časopisy nebo knihy. Těm je určen malý návod.
Existují knihovny, ať už podnikové nebo státní. Někdo má možná známého, kterému chodí časopis ze zahraničí. Existuje mnohem víc způsobů, jak literaturu sehnat, jen trochu chtít. Dokonce se vyskytují i takoví nadšenci jako já, kteří o dovolené v zahraničí pohrdli nákupem nových tenisek PUMA a raději si předplatili časopis. Konkrétně článek o novém Spectru SAM byl také částí této atraktivní sportovní obuvi.
Nemusíte samozřejmě jen překládat. Podobná situace je i v dokumentaci k programům. V klubu si nahrajete novou hru. Za měsíc nebo dva k ní můžete napsat menší manuál a pokud jste hru nehráli a neznáte jí, nemuseli jste si jí ani nahrávat. To je úplně jednoduchá aritmetika.
Pokud píšete programy, snažte se udělat aspoň to, že program upravíte do podoby, se kterou je schopen pracovat i někdo jiný (komentáře, ošetření vstupů atd.) a přineste jej do klubu nebo pošlete informaci do INDEXU. Třeba tím jinému členovi klubu ušetříte práci se psaním stejného programu a to je také přínos, ne?
Zkrátka dělejte co je vám nejbližší, ale hlavně něco vůbec dělejte. A skončete s takovým přístupem ke klubu, který začíná převládat. Tedy s takovým, kdy člen přijde jednou za čtrnáct dní do klubu, posadí se na židli, rozbalí magnetofon a zeptá se "CO MATE NOVÉHO ZA PROGRAMY ?"
-PKCS-

P. S. Možná se vám zdá můj článek tvrdý, ale věžte, že je stejně drsný jako dnešní realita. Můžeme o tom vést spory, můžeme s tím i nesouhlasit, ale řečmi tomu moc nepomůžeme. Tento článek vznikl kromě jiného z toho důvodu, že nebylo co dát do zpravodaje. To už není důvod k červenání, to je hrůza a ostuda. Všichni si prostě zvykli, že to někdo udělá.

Uvádíme...

LUPIČ OBRAZU

Ne, to není název detektivky, ani senzační článek v novinách. Pokud znáte nějakého člena Klubu uživatelů mikropočítačů Sinclair v 602. ZO Svazarmu na Praze 6, zkuste se ho zeptat, co mu tento termín řekne. "No ovšem, program téhož jména předváděl někdo letos v červnu, ale už vám neřeknu, kdo to byl. No, ten program je něco na grafiku, ale co přesně to vám nepovím ...". Takhle by možná zareagoval - buďme skromní a položme důraz na slůvko možná. Skromnost je na místě - my, autoři programu, si rozhodně nemyslíme, že by naše dítko někoho zaujalo tak, aby si na ně pamatoval ještě po půl roce. O to více nás překvapila nabídka napsat zrovna tenhle článek.
Takže uveďme věci na pravou míru: onen "někdo" (pachatel ve věci Uloupení obrazu) byl student SPSZ Pavel Maňas, takto polovina sdružení dvou bláznů, kteří si říkají študák SoftWare (jehož druhou polovinu tvoří pisatel článku). Zmíněný Lupič obrazů se skutečně týká grafiky (také jak jinak). Tedy té formy počítačové grafiky, se kterou běžný uživatel Spectra nejčastěji setká - u zaváděcích obrázků firemních her.
Pokud je člověk šťastným vlastníkem tiskárny, určitě nejednou zatoužil vytisknout si takový zaváděcí obrázek své oblíbené hry. Ale ouha! Není-li zrovna obrázek uložen na pásku jako Screen string, určitě je nějakým jiným způsobem chráněn: uložen od jiné adresy, po čtverečkách (tzv. Mad load), spojen s celým programem, nebo prostě uložen bez návěstí.
I my jsme vlastníky tiskárny a i my jsme nejednou zatoužili vytisknout si některý pěkný zaváděcí obrázek, a protože nás přestalo bavit dřít se s každým obrázkem přes půl hodiny, napsali jsme si Lupiče. A protože samo o sobě nám to připadalo příliš jednostraně zaměřené, zařadili jsme do programu i jiné funkce: definování a přemisťování oken na obrazovce (s možností přemístěné okno znovu vymazat), výstup obrázku na ZX-printeru a změnu atributů obrazovky (obrázek se pak jeví stejně jako po vytisknutí na standartní tiskárně). Co se týče původní funkce Lupiče, dovoluje program nahrát obrazovku konvenčně s návěstím nebo po čtverečkách (Mad load). Pro speciální případy obrázků spojených s programem nebo uložených na zvláštních adresách, odkud jsou dále kopírovány dále do Screen memory, umožňuje Lupič nahrávat obrázek netradičně bez návěstí a lze tak získat téměř každý firemní obrázek. Lupič nezabírá opravdu jen u opravdu speciálních nahrávacích rutin jako třeba Cobra.

K tvorbě programu jsme využili programy Hibasic, MCTT, EDIT/ASSM, Příkres, Art Studio. Na konečné verzi se podílel grafikou Ondřej Kafka.

Marek Novotný
študák Software

Recenze. . .

****

Roland Waclawek
Mój mikrokomputer ZX Spectrum
Mlodziežowa Agencja Wydawnicza
1987
(cena 14.50 Kčs, 180,- Pzl)

****

Přestože byl vyhlášen program elektronizace národního hospodářství, nepociťujeme tuto chválihodnou akci na pultech našich prodejen knih a časopisů, kde stále chybí odborné a populární publikace a časopisy s počítačovou tématikou. Proto nám nezůstává nic jiného, než se po knížkách poohlédnout jinde - například v Polském kulturním a informačním středisku v Praze.
Jednou ze zajímavých knih, které byly k dispozici, (v létě '88) je "Mój mikrokomputer ZX Spectrum" od Rolanda Waclawka.
Jedná se o útlou publikaci s rozsahem 144 stran. Autor se v ní pokusil (a můžeme říct, že úspěšně) vytvořit jednoduchý a velice srozumitelný úvod do používání našeho počítače.
Kniha se skládá ze čtyř kapitol:

- ZX Spectrum a zbytek světa
- Opakování jazyka BASIC ZX Spectra
- Vnitřní život ZX Spectra
- V roli uživatele
-

Publikace je vhodně doplněna množstvím programů, grafů a obrázků, které pomohou k lepšímu pochopení vysvětlované problematiky.
V první kapitole nás autor lehce uvede do světa počítačů, který je zde představován ZX Spectrem. Druhá kapitola je věnována základům jazyka BASIC. Tuto kapitolu mnozí z specialistů přeskočí, ale může se stát, že i zde spolu s začátečníky naleznou poučení. Ve třetí kapitole nalezneme úvod do strojového kódu Spectra a filozofii "strojáku". Zajímavá je i kapitolka věnovaná strojákovým rutinám, například posuny obrazovky nebo její zrcadlení. Popis je proveden s pomocí jednoduchých příkládků. Nechybí ani popis jejich parametrů, což pomůže k jejich lepšímu využití i jako doplněk nebo vylepšení programů v BASICu. Také kapitolka o uložení "bejzikovského" programu v paměti nalezne mnoho pozorných čtenářů.
Poslední kapitola je věnovaná popisu některých programů, které jistě najdou širší použití. Všechny jsou v BASICu ZX Spectra. Pro zajímavost uvádíme jejich názvy:

obrazovkový editor
zamalování vnitřku obrazce
kruhové diagramy

grafy funkcí (i dvourozměrné - nakreslení trvá asi deset minut) prostorové modely chemických molekul a jednoduchou hru
pro počítač (tenis - "Match Point" je graficky mnohem zdařilejší, ale také delší).
Dvě poznámky na konec:

1) Nebojte se polštiny. Smysl vám bude zřejmý hned po prvním čtení. Mimo to například "bufor drukarky" odpovídá českému vyrovnávací paměť (lidsky buffer) tiskárny.

2) Pokud si myslíte, že se nic nového v knize nedozvíte tak dva příklady nakonec (nebo je to začátek?). Víte co udělá: {POKE 23617,236 : INPUT n}? Ne, no přece změní kurzor v INPUTU z [L] na [?]. A víte jak potlačit zastavení práce počítače a hlášku "Start tape, then press any key"? Ne, to stačí před SAVE umístit POKE 23736,187.

A co říci závěrem. Kniha je vydařená, ačkoli specialisté mohou namítat, že je příliš elementární. Je však cenově dostupná a tak si ji při troše štěstí mohou koupit i mladší uživatelé Specter. Rovněž majitelé Didaktiků Gama (a je jich mezi námi stále víc) takto mohou získat vhodný doplněk k manuálu, který dodává výrobce, a který je mírně řečeno nedostatečný. Ale to už je zase jiná písnička.

-flrc-

Učíme se programovat....

S p e c L i s p V 1.3

Jazyk LISP vznikl v 60. letech v USA, autorem je Mc Carthy. LISP je funkcionální jazyk , určený k rekurzivnímu popisu symbolických funkcí. Lze jím popsat libovolný algoritmus. Program i data mají stejnou syntaxi - tvar symbolických výrazů. Hlavním použitím LISPu je programování úloh z oblasti umělé inteligence. Nejznámější je LISP V1.5, která se považuje za referenční verzi.

Tento text není učebnicí LISPu, předpokládá znalost některé verze jazyka, (například V1.10 implementované u nás na počítacích řady PDP a SMEP). Obsahuje pouze syntaxi a popis funkcí SpecLispu V1.3 implementovaného na počítačích ZX Spectrum.

Výčet funkcí je kompletní. Pouze u několika funkcí uvedených na konci seznamu není zcela jasný jejich význam, protože tento popis vznikl metodou pokusů a omylů.

Ovládání:

SPACE.....pozastaví výpis
EDIT.....návrat do Basicu
DELETE.....maže poslední znak
CAPS SHIFT + 5.....CANCEL celé řádky
CAPS SHIFT + SPACE.....EXIT
Restart.....RAND USR 24500

Použitá symbolika: A, B. . . atomy
S.....seznam
SV....s - výraz
N.....číselný atom typu INTEGER
F.....forma

Vlastnosti: pname.....jméno atomu
apval.....jméno konstanty určené funkcí cset nebo csetq.
Subr.....vlastnost atomů, které jsou jménem funkce implementované ve strojovém kódu.
Expr.....vlastnost atomů, které označují funkce definované lambda výrazem. (Pomocí funkce de).

Funkce:

car SV	Hodnotou je první element seznamu nebo tečka dvojice. SV je neprázdný.
cdr SV	Hodnotou je druhá část SV (bez prvního elementu).
cadr SV	= car (cdr SV)
cddr SV	= cdr (cdr SV)
cons SV1 SV2	Konstruuje SV ze dvou SV.
atom SV	Test, zda SV je atomem.
eq A B	Testuje shodu dvou necíselných atomů.
quote SV	SV se bere jako konstanta. (Q-forma).
setq A SV	Využívá vedlejšího efektu - přiřazení SV atomu A, který se nevyhodnocuje. Vrací SV. Místo SV může být forma.
csetq A SV	Jako setq, ale A je konstantou.
set SV1 SV2	Využívá vedlejšího efektu – nepřímé přiřazení SV2 s-výrazu SV1. SV1 musí mít vlastnost atomu. Vrací SV2.
cset SV1 SV2	Jako set, ale SV1 je nepřímou konstantou.
rplaca SV1 SV2	Nahradí car cast v SV1 s-výrazem SV2.
rplacd SV1 SV2	Nahradí cdr cast v SV1 s-výrazem SV2.
list SV1 SV2 . .	Hodnotou je seznam argumentů.
cond S1 S2 ...	Každý seznam tvoří dvě formy, první je predikátorem. Hodnotou je hodnota té formy, pro níž platí, že predikátor nabyl hodnoty t a všechny předchozí hodnoty nil. Pro hodnoty všech predikátorů nil není hodnota funkce definována.
while F S	Příkaz cyklu, forma je predikátorem. Cyklus se provádí dokud nabývá predikátor hodnoty t. Seznam S je tělem cyklu.
eval SV	Vrací hodnotu SV. (Vyhodnocuje SV, první element SV se chápe jako funkce).
apply <funkce> S	Aplikuje funkci na seznam argumentů. (Jméno funkce jako konstantu).
function <funkce>	Speciální forma, zabraňuje vyhodnocení argumentu, který je funkcí.
reverse S	Vrací seznam prvků v obráceném pořadí na nejvyšší úrovni.
member SV S	Vrací hodnotu t, jestliže je SV obsazen v S na nejvyšší úrovni.
de A S F	Umožňuje definovat novou funkci jménem <A>. S je seznam vázaných

	lambda proměnných, forma je popisem funkce.
progn S1 S2	Speciální forma, umožňuje sekvenční zpracování příkazů aplikace funkcí, které tvoří S2. S1 je seznam programových proměnných.
Exit	Výstup z cyklu nebo formy progn.
plus N1 N2	Vrací součet argumentů.
diff N1 N2	Vrací rozdíl N1-N2
times N1 N2	Vrací součin argumentů.
div N1 N2	Vrací celočíselný podíl N1/N2.
rem N1 N2	Vrací zbytek po celočíselném podílu N1/N2.
add1 N	Vrací inkrementovanou hodnotu N.
sub1 N	Vrací dekrementovanou hodnotu N.
and SV1 SV2	Vrací hodnotu t, jestliže jsou všechny argumenty různé od nil.
or SV1 SV2	Vrací hodnotu t, jestliže alespoň jeden argument je různý od nil. Nalezne-li se, již se další argumenty nevyhodnocují.
not SV	Ekvivalentní funkce s null, estetické důvody.
null SV	Vrací hodnotu t, jestliže SV je nil nebo prázdný seznam.
equal SV1 SV2	Testuje shodu dvou SV.
zerop N	Testuje nulovost N.
number SV	Testuje, zda SV je číselným atomem.
minusp N	Testuje zápornost N.
greaterp N1 N2	Testuje, zda N1>N2.
lessp N1 N2	Testuje, zda N1< N2.
t	Implicitní konstanta TRUE.
nil	Implicitní konstanta FALSE.
print SV	Vrací hodnotu t, vedlejším efektem je tisk SV na obrazovku.
Read	Vrací SV přečtený z klávesnice, SV se nevyhodnocuje.
Rnd	Vrací pseudonáhodný číselný atom typu INTEGER v rozsahu 0 až 255 odvozený od systémové buňky FRAMES.
pch N	Vytiskne ASCII znak odpovídající N. (Jako CHR$ v Basicu).
prt SV	Pravděpodobně tisk na tiskárnu. Možno abortovat CAPS SHIFT + SPACE.
trace	Vrací hodnotu t. Vedlejším efektem je trasování programu. Vypnutí: (trace nil).
get A <vlastnost>	Vrací hodnotu vlastnosti (odkaz) daného A. Vlastnost zadat jako konstantu.
putprop A SV <vl>	Uloží SV na pozici dané vlastností atomu. Vrací hodnotu vázanou na A.
remprop A <vlast>	Zruší danou vlastnost atomu A. Pokud existuje vrací t, jinak nil.
oblist	Vypíše seznam vlastností (property-list) funkcí a atomů. Vlastnosti viz dříve.
save	SAVE
load	LOAD
verify	VERIFY
cls	CLS
paper N	PAPER N
ink N	INK N

border N	BORDER N
gc	Garbage Collector.
stack N	Pravděpodobně nastavuje hloubku výpočtového stacku.
penu	???
pend	Z argumentu vytvoří seznam, ale to pravděpodobně není hlavním efektem funkcí.
fwd N	???
left N	???
right N	???

K napsání libovolného programu v jazyce LISP stačí pět základních funkci CONS, CAR, CDR, EQ a ATOM.

Prosím o předání zkušeností při práci s jazykem SpecLisp V1.3, popřípadě o upřesnění popisu dosud nejasných funkcí (ozn. ???).

Vladislav ČERNÝ, 22. 5.. 1985, Plzeň, 61065

Podrobnější informace o jazyce LISP V1.10 tamtéž.

S C O P E

(pokračování)

SCR (scroll)

Formát - SCR; číslo: nebo proměnná

Provádí - Scroluje určený počet řádků. Definiční obor je od 3 do 24, kde 24 scroluje celou obrazovku.

Komentář - Je-li číslo mimo obor, program přestane pracovat.

SOUND

Formát - SOUND; výška tónu, délka tónu: nebo proměnná, proměnná:

Provádí - Zahraje nadefinovaný tón.

Příklad - 10 REM Sound; 10, a:

Komentář - Tento příkaz nepracuje stejně jako BEEP v BASICu, viz oddíl zvuk.

TEST

Formát - TEST; číslo testu, proměnná, číslo, návěští:

Provádí - Je to nejkomplexnější slovo SCOPE. Porovnává danou proměnnou s daným číslem a podmíněně skáče na určenou návěští... Podmínku určuje prvé číslo ve slově a říká se mu čislo testu, Další podrobnosti následuji:

číslo testu	podmínka
194	skok, není-li testované číslo rovno proměnné
202	skok, je-li číslo rovno proměnné
210	skok, je-li číslo větší než proměnná

218 skok, je-li číslo menší než proměnná
196 volá rutinu, je-li číslo různé od proměnné
204 volá rutinu, je-li číslo rovno proměnné
212 volá rutinu, je-li číslo větší než proměnná
220 volá rutinu, je-li číslo menší než proměnná

Příklad - 10 REM Test; 202, a, 10, A
20 REM Test; 220, a, 20,

Komentář - Pozor na to, že 4 testy provádějí skok na návěští a 4 volají rutinu. Jestliže se zmýlíte a místo volání rutiny budete žádat skok, může se program vrátit do BASICu nebo přestane pracovat.

VAR (variable) - proměnná

Formát - VAR; písmeno, číslo:

Provádí - Mění hodnotu proměnné.

Příklad - 10 REM VAR; c, 10:

Komentář - Var může být celé číslo od 0 do 255. Pro větší čísla je Bvar. Proměnných může být 52, a to: a až z nebo A až Z

WIPE

Formát - WIPE; číslo: nebo proměnná:

Provádí - Maže určený počet řádek odspodu obrazovky. Wipe = 24; maže celou obrazovku. Rozsah čísel může být 1 až 24.

Příklad - 10 REM Wipe; 1:
20 REM Wipe; d:

Komentář - Musí se dbát na to, aby číslo bylo v definičním oboru nebo se systém zhroutí.

6. Použití barev ve SCOPE

Jak víte z manuálu BASICu, jsou barvy označeny čísly 0 až 7 a použity samostatné příkazy INK a PAPER.

Ve SCOPE je pro barvu papíru, inkoustu, bright a flash použito jediné číslo. Vypočítává se takto: Násobte barvu papíru (0-7) osmi a připočtěte číslo barvy inkoustu. Pro bright připočtěte 64 a pro flash 128. Chcete-li obojí, připočtěte 192.

Tabulka barev

PAPER \ INK	černá	modrá	červená	fialová	zelená	sv.modrá	žlutá	bílá
černá	0	1	2	3	4	5	6	7
modrá	8	9	10	11	12	13	14	15
červená	16	17	18	19	20	21	22	23
fialova	24	25	26	27	28	29	30	31
zelena	32	33	34	35	36	37	38	39
sv.modrá.	40	41	42	43	44	45	46	47
žlutá	48	49	50	51	52	53	54	55
bílá	56	57	58	59	60	61	62	

upozorění:
V jediném příkazu Bdr: je číslo od 0 do 7.

7. Použití zvuku ve SCOPE

Zde je úmyslně použit příkaz odlišný od BASICového BEEP, protože ten používá způsob dle hudební notace. Příkaz SOUND je rychlejší a umožňuje i glissando.
Příkaz SOUND není vytvořen dle hudební notace, je skutečně vytvořen pro akci typu zvuk, nebo-li pro zvukové efekty.
Následující tabulka vám pomůže vytvořit si představu o vlivu obou hodnot - výšky i délky tónu.
Nepoužívejte délku tónu mimo definiční obor, neboť by se vám mohlo stát, že budete čtvrt hodiny čekat až zvuk skončí.

Zvuková tabulka

výška	oktáva	délka
225	C - 2	1 - 10
127	C - 1	1 - 20
63	C	1 - 40
31	C + 1	1 - 80
15	C + 2	1 - 160
7	C + 3	1 - 255

Poznámka: čím vyšší číslo tónu, tim nižší tón a čím vyšší číslo délky, tím delší zvuk.

8. Příklady použití slov SCOPE

Již jsme vás upozornili, že program musí začínat slovem ORG; a končit slovem EXIT;.
Nyní vám předkládáme několik příkladů programů - vyzkoušejte si je. Nezapomeňte, máte-li program napsaný, zadejte přímo PRINT USR 60450 - tím se program přeloží a vložením RAND USR 40000 se rozbíhá.

Použití zvuku

```
10 REM Org; 40000, 50000:
15 REM Var; a, 1:
20 REM Label; A:
25 REM Sound; a, 1:
30 REM Inc; a, 1:
35 REM Test; 194, a, 95, A:
```

Tímto dostaneme zvuk typu laser. Přeloženo do češtiny, tento program znamená: uvést počátek kompilovaného programu na adresu 40000 a rutiny od 50000. Ustav proměnnou se jménem a přiřaď jí hodnotu 1. Adresa za návěštím A je počáteční při skoku. Vydej daný zvuk. Přičti k proměnné a 1. Je-li hodnota proměnné a menší než 95, skoč za návěští A. Není-li, skončí a vrátí se do BASICu. Vyzkoušejte svoji fantazii a zkuste vytvořit nějaké zvuky. Můžete například, použijete-li předchozí program, vytvořit dva zvuky, jeden směřující vzhůru a druhý dolů. Vložte:

```
17 REM Var; b, 63:
26 REM Sound; b, 1:
28 REM Dec; b, 1:
```

9. Ukládání znaků na obrazovku

Napište:

```
10 REM Org; 40000, 50000:
15 REM Var; a, 0:
20 REM Routine:
25 REM Label; A:
30 REM Put; 59, 10, a, "SCOPE"
35 REM END:
40 REM Call; A:
80 REM Exit;
```

Vyzkoušejte různé znaky, slova a pozice na obrazovce. Chcete-li znaky posouvat, přidejte předchozímu programu:

```
36 REM Label; B:
38 REM Over; 0:
45 REM Halt; 3:
50 REM Over; 1:
55 REM Call; A:
60 REM Inc; a, 1:
70 REM Test; 194, a, 30, B:
```

Animovat je také docela jednduché, následující program. Ač jednoduchý, ukazuje možnost pohybu v barvě.

```
10 REM Org; 40000, 50000:
15 REM Var; a, 0:
20 REM Routine;
25 REM Label; A:
30 REM Put; 49, 10, a, - = - =
31 REM Halt; 3:
32 REM Put; 57, 10, a, = - = -
33 REM Halt; 3:
34 REM Put; 57, 10, a, :  (dvě mezery)
35 REM End;
36 REM Label; B:
40 REM Call; A:
45 REM Halt; 1:
60 REM Inc; a, 1:
65 REM Test; 194, a, 24, B:
70 REM Call; A:
200 REM Exit;
```

10. Body, přímky, malování

Uložte následující program. Je to jednoduchá opakující se instrukce Plot:

```
10 REM Org; 40000, 50000:
15 REM Var; a, 0:
20 REM Label; A:
25 REM Plot; 58, a, a;
30 REM Inc; A, 1:
35 REM Test; 194, a, 174, A:
40 REM Exit;
```

Nyní nahraďte následující řádek:

```
25 REM Plot; 48, a, 1:
```

Zkuste malování - např.:

```
12 REM Var; b, 174:
20 REM Draw; 58, +b, +B:
```

Zkuste následující:

```
10 REM Org; 40000, 50000:
15 REM Bdr; 0:
20 REM Chg; O:
22 REM Var; d, 100:
25 REM Var; a, 5:
30 REM Var; b, 0:
35 REM Routine;
40 REM Label; A:
45 REM Note; SQUARE .(čtverec)
50 REM Plot; 6, d, 50;
55 REM Draw; 5, +a, +b:
60 REM Draw; 4, +b, -a:
65 REM Draw; 3, -a, +b:
70 REM Draw; 2, +b, +a:
75 REM End;
80 REM Call; A:
85 REM Halt; 100:
200 REM Exit
```

(dokončení příště)

Kapitoly strojového kódu.

SUPERCODE II

Moduly budou v dalším textu označovány zkratkou SC a pořadovým číslem daným v programu SUPERCODE.

Poznámky k použití modulu:

1.) Většina rutin je relokovatelných. Lze je umístit od libovolné adresy. Volání rutiny se provede z BASICU instrukcí PRINT USR adresa, nebo RANDOMIZE USR adresa.

2.) Nerelokovatelné rutiny jsou vyznačeny v návodech.

3.) Z těchto rutin lze relokovat také rutiny SC22, SC23, SC71, SC76, ale pouze do adres odlišných o celistvý násobek 256 od adresy původní.

Postup je následující:

Nastav danou rutinu, dej Q (výstup). Původní adresa je A. Nová adresa bude Z = A + 256 * Z2, kde Z2 je celé číslo.

Konstanty:

SC22, SC23	Z1 = 228	Z2 = 229
SC71	Z1 = 235	Z2 = 235
SC76	Z1 = 231	Z2 = 231

Vlož program:

```
9990  FOR c = a T0 a + b - l : IF PEEK c = z1 0R PEEK c = z2 THEN POKE C, z2 + PEEK c
9991 NEXT c : G0 TO 9990
```

Meze cyklu jsou dány rozsahem adres, na kterých je modul uložen. Potom nahrajeme upravený modul na pásku a uložíme zpět do paměti od nové adresy.

4.) Pokud je v popisu napsáno např. POKE 63127/8, číslo, znamená to 2-bytový argument. V tomto případě ukládáme nižší byte čísla na první adresu, vyšší byte čísla na další adresu.

Při standartním použití programu SUPERCODE od adresy 53590 do konce paměti je RAMTOP nastaven na 53589.

POPIS MODULŮ:

SC1 - PIXEL UP SCROLL:
Start 64001, délka 97.
Roluje celý obraz nahoru o 1 bod (ne ATTRIBUTY).
Pokud chceme rolovat celý obraz včetně barev, voláme 8x rutinu SC1 a jednou rutinu ATTR UP SCROLL (SC37).

SC2 - PIXEL DOWN SCROLL:
Start 64098, délka 99.
Roluje celý obraz dolu o 1 bod (ne ATTRIBUTY).

SC3 - CHR$/ATTR UP SCROLL:
ROM-rutina, start 3190.
Roluje celou obrazovku včetně ATTR po řádcích.
Rutina se nedá relokovat!

SC4 - PIXEL LEFT SCROLL:
Start 65462, délka 32.
Cyklické rolování: POKE 65475,63
Vypnutí cyklu: POKE 65475, 55
Inverse: POKE 65475,0
Roluje celý obraz o 1 bod doleva (ne ATTRIBUTY)

SC5 - PIXEL RIGHT SCROLL:
Start 65494, délka 32.
Cyklické rolování: POKE 65475,63
Vypnutí cyklu: POKE 65475,55
Inverse: POKE 65475,0
Roluje celý obraz o 1 bod doprava (ne ATTRIBUTY)

SC6 - CHR$ LEFT SCROLL:
Start 64275, délka 25.
Cyklické rolování: POKE 64291,119
Vypnutí cyklu: POKE 64291,54
Roluje celý obraz o 1 znak doprava (ne ATTRIBUTY).
Pokud chceme rolovat obsah obrazovky i s barvami, voláme 1x SC6 a 1x ATTR LEFT SCROLL (SC39).

SC7 - CHR$ TOP LEFT SCROLL:
Start 64300, délka 25.
Cyklické rolování: POKE 64316,119
Vypnutí cyklu: POKE 64316,54
Roluje horní třetinu obrazovky o 1 znak doleva (ne ATTRIBUTY)

SC8 - CHR$ MID LEFT SCROLL:
Start 64325, délka 25.
Cyklické rolování: POKE 64341,119
Vypnutí cyklu: POKE 64341,54
Roluje střední třetinu obrazovky o 1 znak doleva (ne ATTRIBUTY)

SC9 - CHR$ L0W LEFT SCROLL;
Start 64350, délka 25.
Cyklické rolování: POKE 64366,119

Vypnutí cyklu:' POKE 64366,54
Roluje spodní třetinu obrazovky o 1 znak doleva (ne ATTRIBUTY)

SC10 - CHR$ T0P/M1D LEFT SCROLL:
Start 64375, délka 25.
Cyklické rolování: POKE 64391,119
Vypnutí cyklu: POKE 64391,54
Roluje horní dvě třetiny obrazovky o 1 znak doleva (ne ATTRIBUTY)

SC11 - CHR$ MID/LOW LEFT SCROLL:
Start 64400, délka 25.
Cyklické rolování: POKE 64416,119
Vypnutí cyklu: POKE 64416,54
Roluje spodní dvě třetiny obrazovky o 1 znak doleva (ne ATTRIBUTY)

SC12 - CHR$ RIGHT SCROLL:
Start 64425, délka 25.
Cyklické rolování: POKE 64441,119
Vypnutí cyklu: POKE 644'41,54
Roluje celý obraz o 1 znak doprava (ne ATTRIBUTY)

SC13 - CHR$ TOP RIGHT SCROLL:
Start 64450, délka 25.
Cyklické rolování: POKE 64466,119
Vypnutí cyklu: POKE 64466,54
Roluje horní třetinu obrazovky o 1 znak doprava (ne ATTRIBUTY)

SC14 - CHR$ MID RIGHT SCROLL:
Start 64475, délka 25.
Cyklické rolování: POKE 64491,119
Vypnutí cyklu: POKE 64491,54
Roluje střední třetinu obrazovky o 1 znak doprava (ne ATTRIBUTY)

SC15 - CHR$ LOW RIGHT SCROLL:
Start 64500, délka 25.
Cyklické rolování: POKE 64516,119
Vypnutí cyklu: POKE 64516,54
Roluje spodní třetinu obrazovky o 1 znak doprava (ne ATTRIBUTY)

SC16 - CHR$ TOP/MID RIGHT SCROLL:
Start 64525, délka 25.
Cyklické rolování: POKE 64541,119
Vypnutí cyklu: POKE 64541,54
Roluje horní dvě třetiny obrazovky o 1 znak doprava (ne ATTRIBUTY)

SC17 - CHR$ MID/LOW RIGHT SCROLL:
Start 64550, délka 25.
Cyklické rolování: POKE 64566,119
Vypnutí cyklu: POKE 64566,54
Roluje spodní dvě třetiny obrazovky o 1 znak doprava (ne ATTRIBUTY)

SC18 - RIPPLE LEFT SCROLL:
Start 64575, délka 18.
Speciální scroll-efekt jako kdyby se každý znak otáčel na válečku

SC19 - SHUTTER LEFT SCROL:
Start 64593, délka 18.
Speciální scroll-efekt, destruktivní rolovaní ve znakové pozici vlevo, atributy zůstávají.

SC20 - RIPPLE RIGHT SCROLL:
Start 64611, délka 18.
Jako SC18, ale doprava.

SC21 - SHUTTER RIGHT SCROLL:
Start 64629, délka 18.
Jako SC19, ale doprava.

(pokračování příště)

****

Klubové zprávy:

*** Letos konečně vyjde 2. díl knihy Jiřího Sritra > Rutiny ROM < . Nezapomeňte si tuto knížku objednat !

*** Pokračování článků z minulého čísla otiskneme opět příště.

*** Očekáváme Vaše písemné příspěvky, recenze, manuály a pod.. Uveřejněné články honorujeme.

*** Klub Sinclair rozšířil svoji činnost o sekci > CP/M <.
Schůzky: liché úterý od 17.00 v Městské stanici mladých techniků na Julisce (hlavní vchod). Je nutné se přesouvat ! liché soboty od 13.00 ve VSB Juliska (společně se sekcí Spectrum).

*** Uvítáme aktivní členy pracující s počítači 128+, 128+2 a 128+3.

Sinclair 602, technický zpravodaj pro mikroelektroniku a výpočetní techniku. Vydává 602.Z0 Svazarmu pro potřeby vlastního aktivu.
Odpovědný redaktor Karel Rott (tlf. 781-51-18).
Adresa redakce: 602.Z0 Svazarmu, Spectrum klub, Wintrova 8,
Praha 6, 160 41. Telefon: 32-85-63.
Povoleno UVTEI pod evidenčním číslem 37 006. Cena 6,- Kčs
dle čcú č. 1030/202/86.
Náklad 800 výtisků. Praha, leden 1989