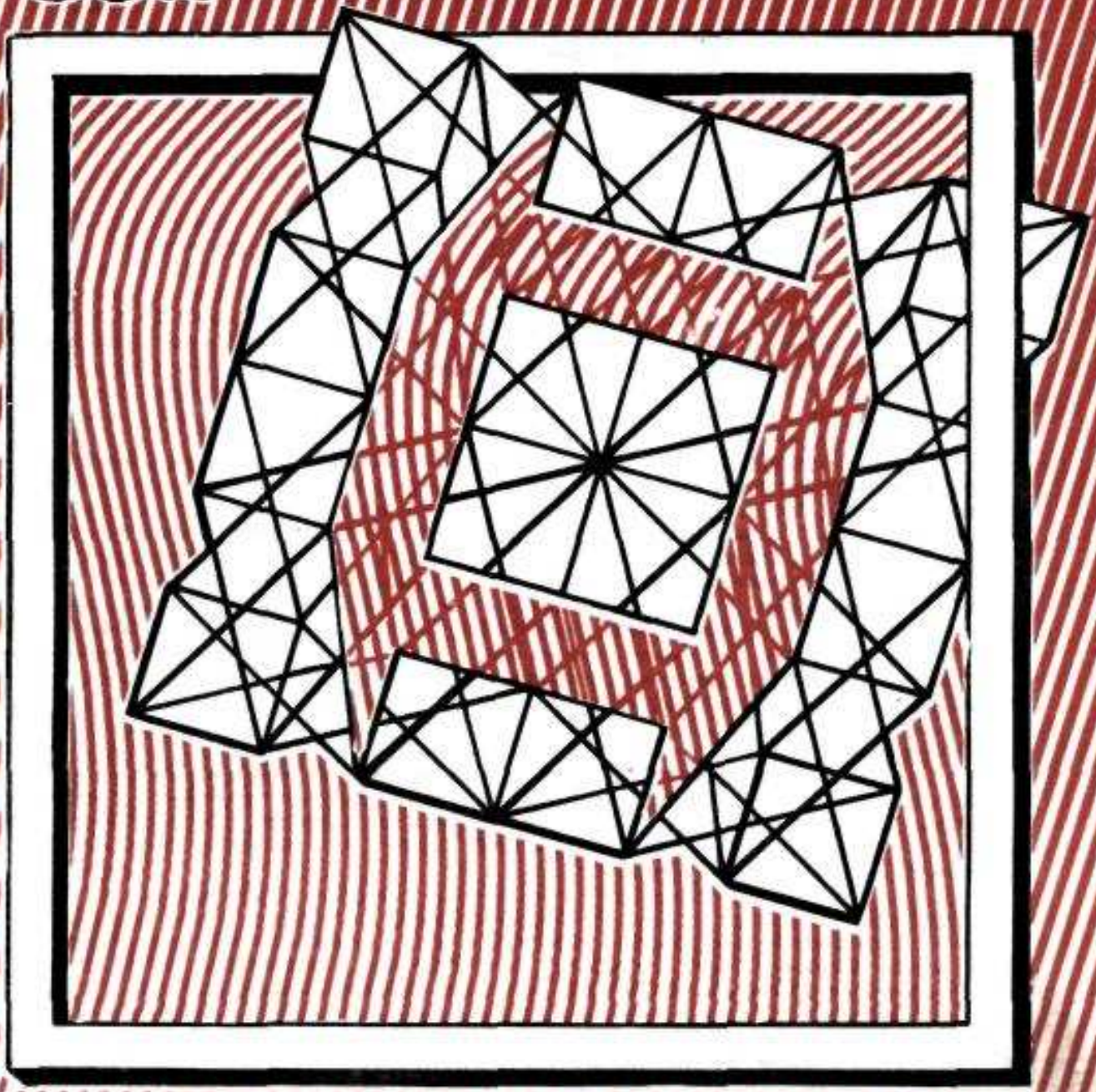
SVAZARM

SPECTRUM

3
89

602



Kopírovací programy...

***** LEC COPY + *****
(C) J. Lamač 9/1986

Tento program je určen pro pohodlné a rychlé kopírovaní programů na kazetách. Byl psán s ohledem na používaní Specter s 80 kilobajty RAM. v normálním Spectru však pracuje také (pokud se spokojíte s paměťovým prostorem okolo 37400 bajtů místo 65535). Výhodami programu je mimo jiné průběžné zobrazovaní velikosti volné paměti v průběhu nahrávání, schopnost nahrát i bloky s chybným kontrolním součtem, půlení a spojování bloků, nedestruktivní skok do Basicu s možným návratem přes teplý start a možnost úpravy programů, přímo v paměti kopírovacím programem.

Kromě obvyklých funkcí pro kazetový magnetofon umí také uchovat svůj obsah na microdrive nebo jej přes network poslat jinému Spectru, které tak může nahrát právě sejmutý program. O výhodách tohoto uspořádání s více Spectry a kopírování na více kazet současně se nejlépe přesvědčíte sami.

Obsluha programu

je jednoduchá. Po nahrání programu do paměti Spectra (LOAD "LC+" nebo LOAD *"m";drive;"LC+") se vlastní strojový kód přemístí na konec paměti, nastaví Basic ramtop na 24599, smaže basicovský zavaděč a ozve se charakteristickým zvukem. Zároveň proběhne test paměti (zjistí se, zda je přítomna paměť 80 kByte) a podle výsledku se nastaví patřičné proměnné. Blikající kurzor oznamuje připravenost programu přijímat vaše příkazy.

Příkazy programu

se vypisují jednotlivými písmeny. Každý příkazový řádek se skládá z vlastního příkazového slova a parametrů (ty jsou ovšem nepovinné). Při pokusu o příkaz, který program nezná, je vypsán report "Bad command". Totéž nastane při zadání nesmyslných parametrů. Zadáme-li příkaz vyžadující ke své činnosti ZX Interface 1, a tento není připojen, dozvíme se že "Interface 1 not present". Je lhostejné, píšeme-li příkazy malými nebo velkými písmeny. Parametry se oddělují čárkou, lze však používat i mezeru. Před příkazem nebo mezi ním a parametry lze napsat libovolný počet mezer. Průběh všech příkazů lze předčasně ukončit klávesou BREAK.

LOAD

bude zřejmě nejvíce používaný příkaz. Nahraje do paměti program z kazety, a to za bloky, které již případně v paměti jsou. Při nesouhlasu kontrolního součtu vypíše report "Parity error". Nevejde-li se nahrávaný blok do paměti, příkaz se ukončí.

SAVE from, to

zaznamená na pásek všechny bloky počínaje blokem "from" a konče blokem "to" (včetně). Při napsáni samotného SAVE ihned ukončeného klávesou ENTER se nahraje všechen obsah paměti.

VERIFY from

kontroluje shodu dat v paměti se záznamem na kazetě, a to od bloku "from". Při nezadání parametru se verifikuje celá RAM.Při chybě se vypíše report "Verify error in block xx", kde "xx" je číslo bloku, kde došlo k chybě. Pak se pokračuje verifikaci dalšího bloku.

CAT
slouží k nalezeni určitého bloku na kazetě. Zobrazí všechny dostupné údaje z každé nalezené hlavičky.

CLEAR
uvolni paměť pro nahráváni dalších bloků.

CLS
smaže obrazovku.

PARITY
Kontroluje správnost kontrolních součtů všech bloků. Při nesouhlasu některého součtu vypíše report "Parity error" následovaný správnou hodnotou kontrolního součtu.

LIST from
vypíše seznam nahraných bloků v paměti počínaje blokem "from". Neuvedeme-li parametr, vypíši se všechny bloky.

DELETE block
vypustí z paměti blok s číslem "block".

INFO from
výpiše počáteční adresu, délku v paměti a flag-bajt každého bloku, počínaje blokem "from". Neuvedeme-li parametr, INFO se provede od prvního bloku. Poslední vypsaná adresa je pointer na první bajt, který již není součástí nahraných bloků (dvě koncové nuly, viz dále).

PEEK adr
vypíše obsahy jednotlivých paměťových buněk počínaje adresou "adr".

POKE adr,val1,val2,...
změní obsahy jednotlivých paměťových buněk na hodnoty "val1", "val2" atd., a to od adresy "adr". Poslední parametr musí být bezprostředně následován klávesou ENTER.

Upozornění: adresy používané v příkazech INFO, PEEK a POKE NEJSOU skutečnými adresami, na kterých se data nacházejí. Jsou to jakési "pseudoadresy", jichž pro snadnější činnost i obsluhu používá kopírovací program a které se při vykonání příkazu transformují na skutečné hodnoty adres. Není proto možné např. změnit hodnoty systémových proměnných nebo pouknout do kopírovacího programu samého.

WRITE
vyšle obsah paměti na network. "Číslo vysílací stanice je přitom 0, takže je možné posílat data několika počítačům současné.

WRITE drive,name
uchová obsah paměti na microdrive o číslu "drive" a se jménem "name". Záznam na microdrive má standardní ROM formát jako typ FILE, lze s ním proto provést i MOVE bez dalšího kopírovacího programu apod.

READ

naplní paměť bloky přijatými z networku. Bloky se připojí k blokům, které již v paměti jsou. Při přeplnění paměti se činnost ukončí, vypíše se report "Out of memory" a v paměti jsou data v takovém stavu, v jakém byla před provedením READ. Nedoporučuji v průběhu tohoto příkazu mačkat BREAK, neboť program by pak nemohl určit konečnou adresu všech bloků.

READ drive, name

naplní paměť bloky přečtenými z microdrive. Ostatní jako READ.

BASIC

skok do Basicu přes chybové hlášení. Do programu se lze kdykoli vrátit příkazem RUN (teplý start, všechny bloky zůstanou zachovány). Hodí se ve chvílích, kdy potřebujeme provést CAT nebo ERASE.

Formát dat v paměti

budou potřebovat znát všichni, kdo budou chtít pomocí LC+ půlit a spojovat bloky anebo měnit jejich obsah. Data mají tvar spojového seznamu. Každý blok začíná dvěma bajty, které udávají délku vlastního datového bloku (tj. bez těch zmíněných dvou). Vlastní datový blok začíná flag-bajtem, pak následují jednotlivé bajty bloku a blok je ukončen kontrolním součtem, který se rovněž ukládá do paměti. Obsazení jednoho bajtu navíc je bohaté vyváženo skutečností, že lze kdykoli porovnat kontrolní součet nahraný z pásku s kontrolním součtem vypočteným a tak se vyhnout krizové situaci, kdy se např. mohla změnit nahraná data v paměti (krátkodobý výpadek sítě apod.). Každý blok tak vlastně zaujímá paměť o velikosti své délky plus čtyři. Na konci všech datových bloků jsou dva nulové bajty, podle kterých program usuzuje, kde mu vlastně data končí. Pozor proto při příkazu POKE, je třeba vždy nejdříve napouknout tyto dvě nuly na patřičné místo a pak teprve např. změnit délku bloku, jinak se program s největší pravděpodobností zhroutí.

Bohaté využití programu a nesmírnou spokojenost všem jednotlivým i hromadným uživatelům přeje

autor programu

Murphologie:
==========
První zákon programování:
Každý daný program právě vložený do počítače je zastaralý.

Sutinův druhý zákon:
Největší zábavu skýtají ty počítačové úlohy, které nejsou v praxi naprosto k ničemu.

Greerův 3. zákon:
Počítačový program dělá jen to, co mu řeknete, nikdy vsak nedělá to, co byste chtěli, aby dělal.

Poulsenova prognóza:
Jestliže se něco nechá běžet na plný výkon, určité se to poláme.

Kontrola joysticku BASICem

čili, není tak zle, jak se zdá

Jednou z mnoha nepříjemností vzniklých z toho, že píšeme program v BASICu je daná jeho rychlosti. Máme na mysli užíváni joysticku. Představte si takovou situaci — píšeme hru, ve které potřebujeme ovládat objekt, pohybující se po obrazovce. Nejlepší by bylo ovládat ho joystickem. V mnoha počítačích existuje možnost odečítat v libovolném momentě hodnotu stavu portu, na který je joystick připojen. Tato možnost se nedá prakticky využit, když je vyhodnoceni příslušných bitů v BASICu tak pomalé, že všichni po několika pokusech přecházejí na řízeni pomoci klávesnice. Dlouhý čas se používá tento způsob, až se nakonec objevilo řešení, jehož výsledkem je uvedený program.

Nejdůležitější jeho část je podprogram, umístěný od řádku 20 do řádku 70. Podprogram vykonává následující činnosti - na řádku 20 vynuluje proměnné (L-levá, S-střelba, D-dolů, G-nahoru) odpovídající daným funkcím joysticku a nastaví proměnnou P-pravá na hodnotu odečtenou z portu 31 (což je interface Kempston). V řádcích 30 až 60 hodnoty proměnných zůstanou stejné nebo se změní na 1, což odpovídá signálu "joystick je vychýlen na danou stranu". Na řádku 80 se inicializují proměnné X a Y, čili určíme polohu objektu, který budeme joystickem řídit. V řádcích 90 až 130 je část programu, která tiskne posunovaný objekt (jeho barva se mění podle toho, je-li stisknuto tlačítko střelba).

Ještě několik slov k objasnění podprogramu na řádce 20. Nebudeme vysvětlovat jeho algoritmus, jen si povíme jeho činnost pro různé hodnoty proměnné P. Výsledkem jeho činnosti je uložení hodnot 1 nebo 0 do proměnných L, S, G, P a D odpovídající hodnotám příslušných bitů bajtu, který se odečítá z portu 31. (Pro znalce asembleru Z80, - je to na základě proměnných SRL a SLA.) Změnou hodnoty 31 v řádku 20 je možné používat i jiný interface než Kempston s podmínkou změny podmínek proměnných L, P, D, G a S, odpovýdajícím podle příslušných bitů bajtu portu. Program je rovněž možné používat všude tam, kde je potřebné představení hodnot v binární formě.

Prohlédneme-li si program pozorně, snadno zjistíme, že v něm chybí základní ochrana proti překročení parametrů PRINT AT a PAPER, proto se nedivte, že například po dlouhém ohybu vpravo se ohlásí chyba překročení parametru.

Program v Basicu vypadá takto:

```
 10 GO TO 80
 20 LET L=0: LET G=0: LET S=0:LET D=0:LET P= IN 31
 30 IF P>=16 THEN LET S=1: LET P=P-16
 40 IF P>=8 THEN LET G=1: LET P=P-8
 50 IF P>=4 THEN LET D=l: LET P=P-4
 60 IF P>=2 THEN LET L=l: LET P=P-2
 80 LET X=16: LET Y=10
 90 GO SUB 20
100 PRINT AT Y,X; PAPER 5-S;" "
110 LET Y=Y-G+D
120 LET X=X+P-L
130 GO TO 90
```

(Bajtek 10/88)

SUPERCODE II (pokračování)
****

SC22 - PIXEL BOXLEFT SCROLL
Start 58571, délka 112, konec 58682.
Voláni: RAND USR 58571
Pozice pravého horního rohu: POKE 58679,hodnota
Zadaní třetiny obrazu: POKE 58680,64 (horní)
POKE 58680,72 (střední)
POKE 58680,80 (spodní)
Šířka obdélníku v bodech: POKE 58681.hodnota
Délka obdélníku ve znacích: POKE 58582,hodnota
Roluje obsah obdélníku o výšce 1/3 obrazovky o 1 bod doleva. Zadání POKE je nutné před každým voláním. Šířka obdélníku se musí zadat. ATTRIBUTY zůstávají na místě. Rutina není relokovátelná!

SC23 - PIXEL BOXRIGHT SCROLL
Start 58608, délka 75, konec 58682.
Volání: RAND USR 58608
Pozice levého horního rohu: POKE 59679,hodnota
Zadání třetiny obrazu: POKE 58680,74 (horní)
POKE 58680,72 (střední)
POKE 58680,80 (horní)
Šířka obdélníku v bodech: POKE 58681.hodnota
Délka obdélníku ve znacích: POKE 58682,hodnota
Funkce je stejná jako u SC22, ale roluje doprava.

SC24 - SCREEN$ FILL
Start 64828, délka 30, konec 64857.
Kód znaku: POKE 64829,hodnota
Výšká obdélníku: POKE 64831,hodnota
šířka obdélníku: POKE 64834,hodnota
Pozice PRINT AT levého horního rohu: POKE 64832,řádek
POKE 64835,sloupec
Vyplní zadaný obdélník stejnými znaky, jejichž kód je na adrese 64829.

SC25 - SCREEN$ STORE
Start 64744, délka 12, konec 64755.
Ukládá obsah obrazovky do paměti. Počáteční adresa uložení se zadá na adresu 64745/6 a musí být nad RAMTOPEM. Délka ukládaného bloku je 6912 bytů.
Poznámka: RAND USR 3756 dělá kopii obrazovky na tiskárně (jako příkaz COPY).

SC26 - SCREEN$ OVERPRINT
Start 64756, délka 28, konec 64783.
Vyvolá obraz uložený pomocí STORE. Počáteční adresa uložení se zadá na adresu 64757/8.

SC27 - SCREEN$ EXCHANGE Start 64784, délka 25, konec 64808.
Uloží právě existující obraz do paměti a zároveň vyvolá z paměti obraz jiný (výměna informace mezi obrazovkou a paměti). Počáteční adresa uložení se zadá na adresu 64785.

SC28 - SCREEN$ INVERT
Start 64809, délka 19, konec 64827.
Udělá inverzi celého obrazu.

SC29 - SCREEN$ MERGE
Start 63976, délka 25, konec 64000.
Míchá obraz uložený v paměti s obsahem obrazovky. ATTR se nemění. Adresa uloženi se žádá na adresu 63980/1.

SC30 - INK CHANGE
Start 64858, délka 25, konec 64882.
POKE 64859,barva. Mění barvu INK v celém obraze.

SC31 - PAPER CHANGE
Start 64883, délka 31, konec 64913.
POKE 64884,barva. Mění barvu PAPER v celém obraze.

SC32 - FLASH ON
Start 64914, délka 17, konec 64830.
Zapíná blikáni v celém obraze.

SC33 - FLASH OFF
Start 64931, délka 17, konec 64947.
Vypíná blikáni v celém obraze.

SC34 - BRIGHT ON
Start 64948, délka 17, konec 64964.
Zapíná vysoký jas v celém obraze.

SC35 - BRIGHT OFF
Start 64965, délka 17, konec 64981.
Vypíná vysoký jas v celém obraze.

SC36 - ATTR FILL
Start 64982,délka 44, konec 65025.
Voláni: RAND USR 64982
Nová hodnota ATTR: POKE 64983,hodnota
PRINT AT pozice levého horního rohu: POKE 64986/5, hodnoty
Šířka obdélníku: POKE 64988,hodnota
Výška obdélníku: POKE 64989,hodnota
Plní ATTR v zadaném obdélníku na zadanou hodnotu.

SC37 - ATTR UP SCROLL
Start 65026, délka 55, konec 65080.
Voláni: RAND USR 65026
Nová hodnota ATTR: POKE 65027,hodnota
PRINT AT pozice levého horního rohu: POKE 65030/29,hodnoty
Šířka obdélníku: POKE 65032, hodnota
výšká obdélníku: POKE 65033, hodnota
Roluje ATTR v zadaném obdélníku o 1 řádek nahoru. Dá se zadat nový obsah spodního řádku ATTR.

SC38 - ATTR DOWN SCROLL
Start 65081, délka 62, konec 65142.
Volání: RAND USR 65081
Nová hodnota ATTR: POKE 65082,hodnota
PRINT AT pozice levého horního rohu: POKE 65085/4,hodnoty
Šířka obdélníku: POKE 65087,hodnota

Výška obdélníku: POKE 65088,hodnota
Roluje ATTR v zadaném obdélníku o 1 řádek dolů. Dá se zadat nový obsah horního řádku.

SC39 - ATTR LEFT SCROLL
Start 65204, délka 52, konec 65255.
Volání: RAND USR 65204

Nová hodnota ATTR:	POKE 64205,hodnota
PRINT AT pozice levého horního rohu:	POKE 65208/7,hodnoty
Šířka obdélníku:	POKE 65210,hodnota
Výška obdélníku:	POKE 65211.hodnota
Cyklické rolování:	POKE 65240,26
Vypnutí cyklu:	POKE 65240,0

Roluje ATTR v zadaném obdélníku o 1 znak doleva. Dá se zadat nový obsah ATTR v pravém sloupci nebo cyklus rolování.

SC40 - ATTR RIGHT SCROLL
Start 65143, délka 61. konec 65203.
Volání: RAND USR 65143

Nová hodnota ATTR:	POKE 65144,hodnota
PRINT AT pozice levého horního rohu:	POKE 65147/6,hodnoty
Šířka obdélníku:	POKE 65149.hodnota
Výška obdélníku:	POKE 65150,hodnota
Cyklické rolování:	POKE 65179,26
Vypnutí cyklu:	POKE 65179,0

Roluje ATTR v zadaném obdélníku o 1 znak doprava. Dá se zadat nový obsah ATTR v levém sloupci nebo cyklus rolování.

SC41 - ATTR SWAP
Start 65256, délka 21. konec 65276.
Starý ATTR: POKE 65257.hodnota
Nový ATTR: POKE 65258.hodnota
Nahradí ATTR zadané hodnoty ATTR hodnoty nové.

SC42 - LASER ZAP
Start 63950, délka 26, konec 63975.
Délka zvuku: POKE 63951.hodnota
Produkuje zvuk střelby z laserové pistole.

SC43 - UNI NOTE SOUND GEN Start 64647, délka 28. konec 64674.

Frekvence:	POKE 64648.hodnota
Rozpětí:	POKE 64649.hodnota
Délka zvuku:	POKE 64651.hodnota
Změna frekvence nahoru:	POKE 64670,28
Změna frekvence dolů:	POKE 64670.29

Produkuje zvuk sirény s měnitelnými parametry.

SC44 - DUAL NOTE SOUND GEN
Start 64675. délka 31. konec 64705.

Délka zvuku:	POKE 64682,hodnota
Frekvence 1:	POKE 64693,hodnota
Frekvence 2:	POKE 64702,hodnota

Produkuje souzvuk dvou tónů konstantních frekvencí.

SC45 - UNI BEEP SIMULATOR

Start 63000. délka 10, konec 63009.
Výška tónu: POKE 63001/2,hodnota
Délka tónu: POKE 63004/5,hodnota
Produkuje 1 tón jako BEEP.

SC46 - MULTI BEEP SIMULATOR
Start 63010, délka 24, konec 63033.
Dekrement výsky tónu: POKE 63011,hodnota
Počet not: POKE 63012.hodnota
Výška tónu: POKE 63014/5.hodnota
Délka zvuku: POKE 63017/8,hodnota
Produkuje vícehlasé měnící se tóny (např 2 sirény).

SC47 - OBLIGUE SCROLL OFF
Start 63034, délka 17, konec 63050.
Speciální SCROLL efekt, pohybuje obrazem po bodu doleva a přitom některé linky mizl. Viz také moduly SC1B až SC21 a SC47.

SC48 - CHR$ DOWN SCROLL
Start 63051, délka 73, konec 63123.
Roluje obraz dolů po znacích, ATTR na místě.

SC49 - CHR$ ROTATE
Start 63163, délka 42, konec 63204.
Pracuje jen se znaky uloženými v RAM! (Alternativní znakový soubor nebo UDG). Otáčí znak o 90 stupňů doprava. Počáteční adresa obrazu znaku se zadá: POKE 63166/7,hodnota

SC50 - CHR$ REFLECT Y-AXIS
Start 63124, délka 19, konec 63142.
Dělá zrcadlový obraz znaku podle svislé osy. Pouze pro znaky v RAM! Adresa znaku: POKE 63127/8,hodnota

SC51 - CHR$ REFLECT X-AXIS
Start 63143, délka 20, konec 63162.
Dělá zrcadlový obraz znaku podle vodorovné osy. Pouze pro znaky v RAM! Adresa znaku: POKE 63144/5,hodnota

SC52 - 24 LINE PRINTING
Návod jak v BASICu tisknout do spodních dvou řádků obrazovky: Použije se příkazu PRINT #1;AT k,0; a dál už je to normální. Je-li k-0, tiskneme do řádku 22 (počítáno od 0), jinak tiskneme do řádku 23. Vymazáni spodní části obrazovky - RAND USR 3438.

SC53 - SCREEN$ SEARCH
Start 60039, délka 123, konec 60161.
Umožňuje určit CHR$-kód na žádané pozici obrazovky.
Př.: PRINT AT 7,13;:LET Z - USR 60039
Výsledkem je kód znaku na pozici 7,13.

SC54 - SCREEN$ PRINT
Start 63728, délka 49, konec 63776.
Parametry: POKE 63732,INK
POKE 63738,PAPER
POKE 63744,FLASH
POKE 63750,BRIGHT
POKE 63756, INVERSE

POKE 63762,OVER
POKE 63768 a 63771, AT
POKE 63774,CHR$ (kód znaku)
Tiskne 1 znak.

SC55 - RND # GENERATOR
Start 63777, délka 18, konec 63794.
Poskytuje 2 bytové náhodné číslo do systémové proměnné SEED.
Čtení hodnoty: LET X = PEEK 23670 + 256 * PEEK 23671

SC56 - BLOCK MEMORY INSERT
Start 63795, délka 11, konec 63805.
Počet bytů: POKE 63796,hodnota
Počáteční adresa: POKE 63798/9,hodnota
Vkládaná hodnota: POKE 63801.hodnota
Vkládá konstantu na všechny adresy zadaného bloku.

SC57 - BLOCK LINE ERASE
Start 63806, délka 96, konec 63901.
Počáteční řádek výmazu: POKE 23728,X
Poslední řádek výmazu: POKE 23729,Y
Vymaže blok řádků od X po Y.

SC58 - CHR$ SWAP
Start 63902, délka 43, konec 63944.
Starý kód: POKE 63903,hodnota
Nový kód: POKE 63905,hodnota
Nahradí všechny znaky starého za znaky nového kódu v obraze.

SC59 - CHR$ SCRAMBLE
ROM rutina! Návod jak způsobit aby počítač šifroval při pokusu o LIST. Je nutné přepsat systémovou proměnnou CHARS: POKE 23606/7,hodnota. V tom případě je nutné před každým tiskem obnovit původní hodnoty: 0 a 60.
Poznámka: Tímto způsobem je možné změnit význam všech znaků na vlastní znakový soubor. Proměnná CHARS v sobě obsahuje o 256 menší hodnotu, než je počáteční adresa tabulky znaků, která začíná původním znakem mezera a konči původním znakem c v kroužku (viz manual). V tom případě bude předefinován význam všech kláves a systémová hlášení budou nesrozumitelná.

SC60 - SUPER RENUMBER
Start 59294, délka 681, konec 59974.
Úprava Basic-programu: Přečísluje řádky od S s intervalem 1 včetně argumentu příkazu GOTO, GOSUB, RESTORE, LIST, LLIST, SAVE.. ..LINE, kromě argumentu jako je 2.8 nebo N+3, které jsou zvýrazněny. Pokud argument není existující řádek, bere se číslo následujícího řádku.
Interval: POKE 59580.1
Počáteční řádek: POKE 59582/3,8

SC61 - LINE RENUMBER
Start 64706, délka 38, konec 64743.
Úprava Basic-programu: Přečísluje řádky. POZOR, nelze použít v programu s funkcemi, jejichž argumenty jsou čísla řádků, jako jsou skoky apod. Viz popis SC60.

SC62 - DEC-HEX CONVERTER
Start 60595, délka 118, konec 60712. Po vyvolání modul očekává zadat z klávesnice desítkové číslo v rozsahu 0 až 65535. Na obrazovku vytiskne hexadecimální ekvivalent zadané hodnoty. Program

SC63 - HEX-DEC CONVERTER
Start 60713, délka 113, konec 60825.
Pracuje stejné jako modul SC62, pouze převod je opačný.

SC64 - REM KILL CONDENSER
Start 60494, délka 101, konec 60594.
Úprava Basic-programu: Vypouští všechny poznámky REM - šetří místo v paměti.

SC65 ON ERROR GO TO
Start 60826, délka 73, konec 60898.
Na začátku programu zadej číslo řádku, na který se má skákat při vzniku chyby: POKE 60877/8,x. Potom vyvolej tuto rutinu. Stane-li se chyba, program skočí na řádek X. Kód chyby lze nalézt na adrese 23681. Ignorují se chyby typu 0 OK, 8 END, 9 STOP.
Potlačení chyb: 1 POKE 23613,PEEK 23613-2
9999 POKE 23610,255 s POKE 23613,PEEK 23613+2

SC66 ON BREAK GO TO
Na začátku programu zadej číslo řádku, na který se má skákat při vzniku chyby: POKE 60952/3,X. Stane-li se chyba, program skočí na řádek X. Ignorují se chyby typu D BREAK, H STOP in input, L BREAK.

SC67 FREE SCROLLER
Návod jak se dá zadat po kolika řádcích chceme, aby systém roloval, než se zeptá Scroll?:
Počet řádka zadáme: POKE 23692,X (proměnná SCR CT)
Chceme-li, aby roloval bez omezení, zapíšeme tam hodnotu 255.

SC68 NON-DELETABLE LINE
Návod, jak způsobit, aby určitý řádek v programu Basic byl nezrušitelný: Pomocí rutiny SC88 si zjistíme počáteční adresu, od které je řádek uložen - například X.
Potom provedeme: POKE X.O : POKE X+1,0
Tím jsme nahradili číslo daného řádku nulou, což způsobí jeho nezničitelnost. Chceme-li takto ošetřit hned první řádek programu najdeme jeho adresu v proměnné PROG (LET X = PEEK 23635 + 256 • PEEK 23636).

SC69 BORDER EFFECTS
Start 60000, délka 38, konec 60037.
Délka trvání: POKE 60006,hodnota
Barvy: POKE 60020,hodnota
šířka pruhu: POKE 60029,hodnota
Vytváří pruhy v okolí obrazu (BORDER).

SC70 INITIALISE
Start 63382, délka 108, konec 63489.
Nuluje všechny numerické proměnné, do řetězců dosadí mezery. Na rozdíl od CLEAR neruší existencí těchto proměnných.

SC71 VARIABLES LIST
Start 60222, délka 185, konec 60406.
Vypisuje všechny proměnné, které jsou momentálně v paměti:
1.) Numerické proměnné
2.) STR$ proměnné
3.) Numerické pole
4.) STR$ pole

5.) FOR-NEXT řídící proměnné
Nedá se relokovat! Vypisuje pouze názvy proměnných!

SC72 STR$ LIST
Start 63490, délka 154, konec 63643.
Vypisuje LIST každého řádku programu Basic, který obsahuje zadanou sekvenci znaků, např. "SUPER", apod.

SC73 STR$ REPLACE
Start 63644, délka 83, konec 63726.
Nahradí každý výskyt řetězce v programu Basic jiným řetězcem
stejné délky!

SC74 FLASH SWAP
Start 60162, délka 30, konec 60191.
Přepíná funkci všech políček obrazovky: co blikalo nebliká a naopak.

SC75 BRIGHT SWAP
Start 60192, délka 30, konec 60221.
Přepíná funkci všech políček obrazovky: Zjasní to, co bylo méně jasné, a naopak.

SC76 PRINT FILL
Start 59136, délka 158, konec 59293.
Nakreslí uzavřený obrazec a nastav PLOT na pozici ležící uvnitř obrazce. Potom zadej POKE 59293,ATTR který má být dosazen. Vyvolej tuto rutinu. Rutina se nedá relokovat!

SC77 RECORD SOUND
Start 65290, délka 28, konec 65317.
Připoj EAR zdířku k magnetofonu, nastartuj přehrávání (max. výstupní napětí). Zadej CLEAR 32767 : RAND USR 65290. Zvuk z magnetofonu se zaznamená do počítače. K reprodukci používej rutinu SC78.
Rutina se nedá relokovat! Používá celou polovinu paměti od adresy 32768 po 65535.

SC78 REPLAY SOUND
Start 65318, délka 32, konec 65349.
Přehraje záznam pořízený rutinou SC77.
Nedá se relokovat!

SC79 SCIFI CHR$ SET
Start 57344, délka 768, konec 58111.
Alternativní soubor znaků. znaků.POKE 23607,223 přepne CHARS na tyto znaky. Přepnutí zpět: POKE 23607,60.

SC80 PROTECT PROGRAMM
Návod jak ochránit program před listováním:
1.) Do programu zařadíme: 1 POKE < PEEK 23635 + 256 » PEEK 23636),100
Program lze použít, ale nelze v něm listovat, dokud nezapíšeme na stejnou adresu nulu.
2.) Přepíšeme PROG: POKE 23636,150 Návrat do normálního listingu: POKE 23636,92
3. Použít rutin SC58, SC98, zrušit funkci ERROR a BREAK – SC65, SC66, zapsat do programu nezrušitelné řádky viz SC68.
4.) V nezrušitelném řádku zadat PAPER = INK.
5.) Používat autostart (nahrávka příkazem SAVE...LINE)
6.) Dát do programu: LET ERR - 256 * PEEK 23614 + PEEK 23613: POKE ERR,0 : POKE ERR + 1,0 Každá chyba nebo přerušení nyní způsobí havárii.

7.) Vynulovat proměnnou DF SZ: POKE 23659,0 - účinek je stejný jako v předchozím bodu.

SC81 APPEND STATEMENT
Start 60407, délka 86, konec 60492.
Usnadňuje editaci textu v Basicu.
Nastav řádkový kurzor na požadovaný řádek. Potom volej tuto rutinu. Dostaneš se do módu EDIT, ale kurzor bude umístěn na konec editovaného řádku. K urychlení reakce kurzoru zadej: POKE 23561,5 : POKE 23562,2 : POKE 23608,0 s POKE 23609,9 (přepisují se systémové proměnné REPDEL, REPPER, RASP a PIP). Účinek tohoto modulu je efektivní u dlouhých řádků.

SC82 CONTRACT PROGRAMM
Start 61400, délka 687, konec 62086.
Urychluje chod programu a šetří, pamět tím 2e umístí text programu do nejmenšího možného počtu řádků. Další možnosti úprav programového textu jsou v rutinách SC90, SC100, SC103, SC104.

SC83 EXPAND PROGRAM
Start 62087, délka 317, konec 62403.
Roztáhne text programu tak, Se je jeden příkaz na jednom řádku (kromě řádku s příkazem IF....THEN). Po spustění se zeptá od kterého řádku má začít fungovat. Jestliže dáš jen ENTER, roztáhne celý program. Po skončení práce je nutno použít rutinu SC60 (SUPER RENUMBER) k přečíslování řádků.

SC84 REM FILL
Start 58892, délka 244, konec 59135.
Vkládá poznámku REM do libovolného řádku programu.

SC85 DATA FILL
Start 63205, délka 177, konec 63381.
Natahuje data uložená v paměti do DATA příkazu v řádku 1.
Počáteční adresa: POKE 63209/10.H0DN0TA
Délka: POKE 63206/7,HODNOTA

SC86 ANALYSIS PROGRAMU
Start 62404, délka 129, konec 62532.
Tiskne počet řádků a příkazů v programu.

SC87 TAPE HEADER READER
Start 62533, délka 286, konec 62818.
Dekóduje a tiskne údaje z hlavičky souboru.

SC88 LINE ADDRESS
Start 59975, délka 13, konec 59987.
Tiskne adresu prvního znaku řádku, který je označen kursorem.

SC89 SCREEN$ GRID
Start 62819, délka 38, konec 62856.
K ulehčení nastavování PRINT/PLOT parametru rozsvěcuje šachovnicovítě každý druhý čtvereček.

SC90 MONOCHROME PROGRAMM
Start 62943, délka 54, konec 62996.
Nastaví barvu pozadí v obrazu na stejnou hodnotu (týká se textu programu).

SC91 ANALYSE MEMORY
Start 62857, délka 86, konec 62942.
Počáteční adresa: POKE 62866/7,hodnota
Tiskne analýzu obsahu paměti počínajíc od zadané adresy. Tiskne adresu (dekadicky), a obsah v HEX, DEC i ve znakové podobě.

(příště dokončení)

Na náš trh přichází elektronika a výpočetní technika známe anglické firmy AMSTRAD - pestrý výběr počítačů se systémy ZX Spectrum, CP/M, PC (včetně PC/XT a PC/AT), PPC640 portable, tiskárny, rozsáhlá nabídka příslušenství a antény s přijímači pro družicový příjem.

Zajímavým typem je počítač:

SINCLAIR PC 200

kompatibilní s IBM PC/XT.
Základní technické data:
16-bit procesor 8086 - 8 Mhz
512K RAM + možnost použití standardních rozšiřovacích karet Paralelní port (Centronics) Seriový port (RS 232)
Klávesnice standard AT
Vestavěný disk 3.5" 720k
Konektor pro externí jednotku disku 5.25" 360k nebo 3.5" 720k
Kompatibilní ROM BIOS
Patice pro matematický co-procesor 8087
IBM kompatibilní rozšiřující konektor
Anagolový port pro joystick, kompatibilní s herním software
CGA display adapter!
40 znaků x 25 řádek a 80 znaků x 25 řádek textu v 16-ti barvách
Rozlišovací schopnost v rastru 8 x 8 bodů
320 na 200 bodů při 4 barvách nebo
640 na 200 bodů při 2 barvách
modulátor pro standardní TV
MDA display adapter:
80 znaků x 25 řádek při rastru 9 x 14 bodů
Rozměry: 450 (š) * 85 (v) * 335 (d)
Hmotnost: 5.4 kg

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Prodejna AMSTRAD ve VTI 602. ZO Svazarmu, Martinská 5, Praha 1 (tel. 22 87 74 - pondělí až pátek 08.00 - 10.00 hod.) zajišťuje prodej, servis a poradenskou službu rozsáhlého sortimentu výpočetní techniky a elektroniky (včetně typů Sinclair ZX Spectrum 128+2a, 128+3, CPC, PCW, PC 200, ostatních PC kompatibilních, tiskáren a dalších).
Obvyklé služby VTI 602. ZO Svazarmu (prodej publikací, programového vybavení, kopií z mikrofiši atd.) prodejna poskytuje na místě a v čase obvyklém (úterý až čtvrtek 10.00 - 12.00, 14.00 - 17.00 hod.. Pátek 10.00 - 12.00, 14.00 - 16.00 hod. v Martinské 5, Praha 1 (tel. 22 87 74)).
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

Pro začátečníky...

Jistě se po nalistování na obvyklou stránku ptáte, kde je další pokračování základů programování ve strojovém kódu. Když jsme autora žádali o další kapitolu, řekl nám, že po vydání knihy Ladislava Zajíčka "BITY DO BYTU" se jeho základy stávají zbytečnými.

****

Ladislav Zajíček

BITY DO BYTU

Základy programováni ve strojovém kódu - assembleru Z80

Mladá fronta 1988 cena 17 Kčs

****

Jak již lektor ve své předmluvě píše, dá se tato kniha zařadit mezi populárně-naučnou literaturu - učí základům programováni ve strojovém kódu - assembleru Z80 jednoduchou a každému snadno přístupnou formou.

Kniha je rozdělena na tři části:

V první části nám autor nabízí příběh trvající týden, do kterého se máte vžít, vše si přečíst, odzkoušet a tím zvládnout základy potřebné k pochopeni druhé části.

Bitové pondělí s bity a bajty - co je bit a bajt, adresa a jiné základní výrazy.

Úterý zaslíbené paměti - již název napovídá, že se bude mluvit o pamětech, dále o interpretu, kompilátoru a pod..

Hexadekadická středa - poznáme rozdíly mezi číselnými soustavami, mezi assemblerem a monitorem atp..

Logický čtvrtek pana Boolea - zde jsou vysvětleny základy Booleovy algebry (logické funkce AND,OR,NOT,XOR).

Zakódovaný pátek - informuje o mezinárodních znacích ASCII,dále jak počítač ukládá program napsaný v assembleru, jak jej zobrazuje, relokuje a j..

Mikroprocesorová sobota - vysvětluje funkce jednotlivých registrů Z80.

Neděle na lince - osvětluje, jak počítač počítá, jak udělat smyčku v assembleru a další informace před vstupem do druhé části knihy.

Druhá část obsahuje sedm kapitol:

- Adresovací mody Z80 - 1. a 2. část
- Skoky, voláni a návraty
- Logické instrukce
- Bitové manipulace a posuvy
- Aritmetické instrukce a blokové prohledáváni
- O interfacingu

Experimenty v každé z kapitol názorně ukazuji funkce a možnosti strojového kódu - assembleru Z80.

Třetí část knihy má jen dvě "kapitoly":

V části nazvané "Den po" autor čtenáři odpovídá na otázky, které ho mohly při čtení knihy napadnout.

Příloha obsahuje např.:

- Výpočet doby provedení programu
- Diagram registrů
- Schematické znázornění průběhu vybraných instrukcí
- Vývody pouzdra Z80 a blokový diagram
- Instrukce Z80 zařazené do skupin
- Soubor instrukcí Z80
- Vliv instrukcí na stavové indikátory

Na knihu "Bity do bytu" čekalo nejenom mnoho spectristů,ale i uživatele ostatních počítačů obsahujících procesor Z80. Tato kniha vzhledem ke své kvalitě se jistě stane nejčtenější knihou vaší knihovničky. Mimochodem: má vůbec cenu tuto knihu ukládat do knihovničky, když ji budete stále potřebovat ? A nebojte se v knize označovat důležité věci, pro vlastní větši přehled, než jaký vám nabízí obsah.

-WILLOW-

****

Klubové zprávy:

*** Předem se omlouváme za opožděné zaslání zpravodaje. Příčin zpoždění je mnoho (vypadají jako výmluvy, ale jsou skutečné): nedostatek kvalitních podkladů, špatná redakční *práce*, opoždění v tiskárně atd.. Ale plánovaná 4 čísla ročně budou.

*** Postrádáme Vaše příspěvky. Dopisů, ve kterých nám nabízíte svoje programy, je dost. Ale většinu příspěvků pro jejich rozsáhlost nelze otisknout. Závažné příspěvky předáváme redakci MIKROBAZE. Pište nám o Vaši práci - skutečném využití ZX Spectrum nebo QL.

*** Služba INDEX postrádá Vaše nabídky pomoci - jak při řešeni software, tak hardware. Velice žádaná je pomoc při připojení nejrůznějších zařízení (tiskáren, diskových jednotek atd.) i při programovém řešení nejrůznějšího druhu. Obrací se na nás i podniky . Naše možnosti nestačí a ani nelze řešit Vaše problémy individuálně. Využijte možnosti INDEXu - adresa ne. Vaše adresa nebude sdělena - žádost o pomoc Vám bude zaslána k Vašemu kontaktu. Již vyřešené nebude nutno pracně objevovat znovu.

*** V dopisech nás žádáte o adresu bydlištěm Vám blízkých členů. Napište nám adresu Vašich nebo Vám známých klubů nebo kroužků. Pomůžete mnohým z Vás při řešení někdy velice jednoduchých a Vám známých problémů. Navážete také někdy velice užitečné a neočekávané kontakty.

*** Prostory i provoz klubu Vám neumožňovaly aktivní setkání. Od září pracujeme v lepších podmínkách - pomoc nám nabídla Městská stanice mladých techniků na Julisce- bezprostřední soused našeho Výcvikového střediska branců.

*** V našem klubu pracuje sekce CP/M. Práci vedou dnes již populární Jiří Lamač a Jakub Vaněk - autoři jednoduché a plně kompatibilní verze CP/M na ZX Spectrum 48 kB. Jednoduchá hardwarová úprava umožňuje provoz CP/M na ZX 128. Bez hardwarové úpravy je vyřešeno použiti CP/M na 128+2a. Nehledejte v klubu kopírování programů. Neobejdete se ale bez pomoci při maximálním využiti Vašeho počítače.

*** Ve výše zmíněné stanici mladých techniků pracuje i náš kroužek mládeže, který má volnou kapacitu. Věříme, že je to pro Vaši malou informovanost. Vždyť kroužky mládeže jsou jinde plně obsazeny. Informujte se o možnostech účasti.

Sinclair 602 - technický zpravodaj pro mikroelektroniku a výpočetní techniku. Vydává 602. ZO Svazarmu pro potřeby vlastního aktivu.
Zodpovědný redaktor: Václav Vrba (tel. 22 42 11).
Adresa redakce: 602. ZO Svazarmu, Wintrova 8, Praha 6, 160 41. Telefon: 32-85-63.
Povoleno ÚVTEI pod evidenčním číslem 87 0O6. Cena 5.- Kčs dle ČCÚ č. 1030/202/86.
Náklad 800 výtisků.
Praha, červen 1989.
