

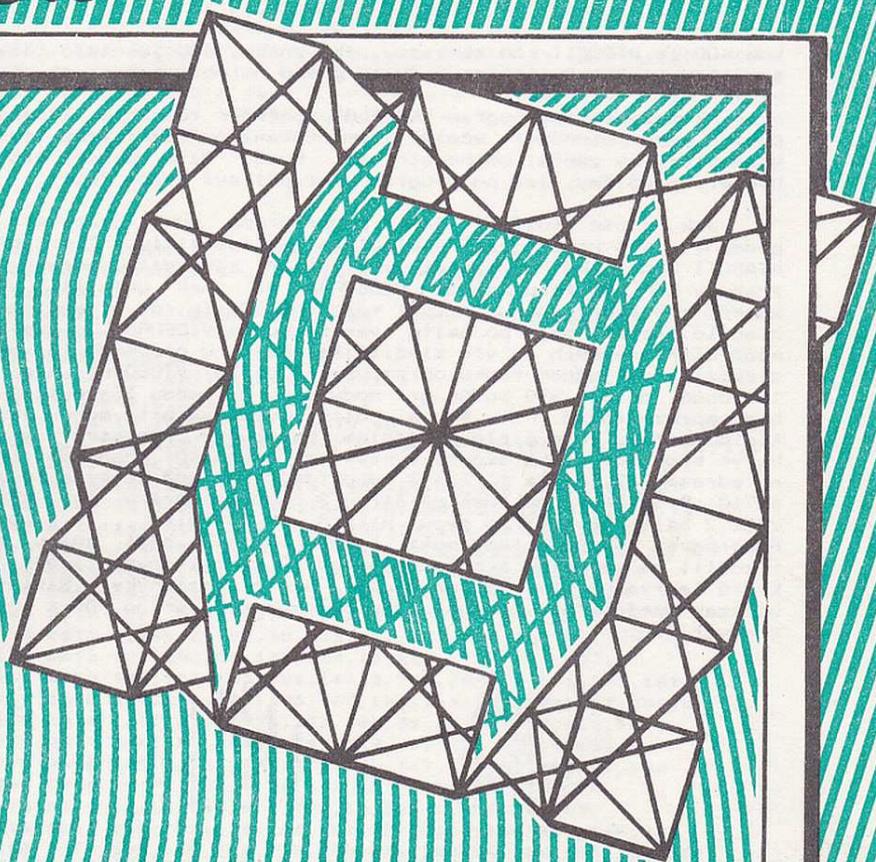


602

ATARI

3

90



ZE ZÁPISNIKU PROGRAMÁTORA

TEXT V GRAFICKÝCH MODECH (4)

V poslední části našeho seriálu si nastíníme činnost podprogramu TEXTPLOT jehož autorem je známý Charles Brannon. Podprogram byl napsán pro zobrazování textu v grafických módech 3, 5, 7 a 15. Jeho použití je velmi jednoduché. Po instalaci do paměti a vyvolání grafického modu stačí napsat příkaz:

```
X=USR(1536,KOD,COL,X,Y)
```

kde KOD je ATASCII kód zobrazovaného znaku, COL je číslo barvového registru, a X a Y jsou souřadnice znaku na obrazovce.

Protože je podprogram TEXTPLOT poměrně rozsáhlý a detailní popis jeho činnosti včetně komentovaného zdrojového textu v assembleru by zabral mnoho stránek, budou dále pouze nastíněny některé problémy, jež podprogram musí při své práci řešit.

Jak jistě tušíte, část podprogramu musí pracovat velmi podobně, jako podprogramy z minulých částí seriálu. T.j. převzít ATASCII kód znaku, vypočítat jeho interní kód, dále adresu tvaru znaku v ROM a adresu kam do VIDEORAM má znak umístit. Vlastní způsob zobrazení znaku je však rozdílný. Zatím co u gr. modu 8 stačilo přenést bajt po bajtu tvar znaku do VIDEORAM, u grafických módů čtyřbarevných je věc složitější. Např. u gr. modu 15 máme k dispozici na jednom řádku obrazovky 40 bajtů VIDEORAM ale pouze 160 bodů, oproti 320 bodům gr. modu 8. V gr. modu 15 je tedy každý bod reprezentován d v ě m a bity. Tyto dva bity mohou vyjádřit číslo 0 až 3. Toto číslo je číslem barvového registru, v jehož barvě bude bod na obrazovce vykreslen. Číslo "0" odpovídá registru na adrese 712, čísla 1, 2 a 3 odpovídají po řadě adresám 708, 709 a 710. Protože každý znak má šířku 8 bodů, zabere v gr. modu 15 znak 2 bajty (8 bodů po dvou bitech = 16 bitů ... 2 bajty). Podprogram TEXTPLOT tedy musí každý tvarový bajt znaku z ROM rozdělit na poloviny a z každé poloviny zkonstruovat jeden bajt, který teprve umístí na příslušné místo obrazovky. Následující obrázek demonstruje činnost podprogramu pro písmeno "A" a barvový registr 709:

```

.....
...**...      .....*.*.....
..****..      .....*.*.*.....
**..**..      ...*.*.....*.*..
**..**..      ...*.*.....*.*..
*****..      ...*.*.*.*.*..
**..**..      ...*.*.....*.*..
.....

```

Jak je z obrázku patrné, každý nenulový bit tvarového bajtu byl nahrazen dvojicí bitů tak, že tato dvojice vyjadřuje číslo 2, které odpovídá barvovému registru na adrese 709. Každý nulový bit tvarového bajtu byl nahrazen dvojicí bitů vyjadřující číslo "0",

kteře odpovídá barvovému registru 712, což je barvový registr pozadí.

Přesně stejně, jako u gr. modu 15, pracuje podprogram v případě grafického modu 7 a obdobně pracuje i v případech gr. modů 3 a 5. Rozdílné jsou samozřejmě velikosti zobrazených znaků.

V dále uvedeném vřpisu je na řádcích 30000 až 30410 vlastní podprogram TEXTPLOT. Na řádcích 10 až 90 je demonstrační program, ze kterého je dostatečně zřejmé, jak podprogram TEXTPLOT používat.

-js-

Pozn. red.: Logickým závěrem seriálu by byl popis podprogramu SUPERTEXT, který zobrazuje znaky v jakémkoliv grafickém modu, přičemž je možné volit barvu znaků, velikost znaků, jejich umístění na obrazovce a jejich orientace (SUPERTEXT může znaky zobrazovat i "ležatě" a "vzhůru nohama"). Tento podprogram a jeho popis byl však již několikrát zveřejněn a tak nebudeme "nosit dříví do lesa".

```

BN 10 REM *** TEXTPLOT - DEMO ***
LF 20 GOSUB 30000:REM INICIALIZACE PODPROGRAMU
ZH 30 DIM T$(5)
OF 40 GRAPHICS 0:?"ZADEJ CISLO GR. MODU":INPUT GR
QG 50 T$="Text"
BS 60 GRAPHICS GR+16
EK 70 FOR I=1 TO LEN(T$):L=ASC(T$(I))
JY 80 M=USR(1536,L,1,I-1,2):NEXT I
GA 90 FOR K=1 TO 3000:NEXT K:GOTO 40
JH 30000 RESTORE 30010:ML=1536:FOR I=0 TO 252:READ A:POKE ML+I,A:
      NEXT I:RETURN
DK 30010 DATA 184,240,10,201,4,240,11,170,104,104,202,208
MC 30030 DATA 251,169,253,76,164,246,104,133,195,104,201,128
YF 30050 DATA 144,4,41,127,198,195,170,141,250,6,224,96
PI 30070 DATA 176,15,169,64,224,32,144,2,169,224,24,109
KK 30090 DATA 250,6,141,250,6,104,104,141,251,6,104,104
PY 30110 DATA 141,252,6,14,252,6,104,104,141,253,6,133
DC 30130 DATA 186,166,87,169,10,224,3,240,8,169,20,224
BT 30150 DATA 5,240,2,169,40,133,207,133,187,165,88,133
PY 30170 DATA 203,165,89,133,204,32,228,6,24,173,252,6
JF 30190 DATA 101,203,133,203,144,2,230,204,24,165,203,101
BF 30210 DATA 212,133,203,165,204,101,213,133,204,173,250,6
OR 30230 DATA 133,187,169,8,133,186,32,228,6,165,212,133
EM 30250 DATA 205,173,244,2,101,213,133,206,160,0,162,8
ZO 30270 DATA 169,0,133,208,133,209,177,205,69,195,72,104
AE 30290 DATA 10,72,144,8,24,173,251,6,5,208,133,208
BO 30310 DATA 224,1,240,8,6,208,38,209,6,208,38,209
RO 30330 DATA 202,208,228,104,152,72,160,0,165,209,145,203
OL 30350 DATA 200,165,208,145,203,104,168,24,165,203,101,207
JF 30370 DATA 133,203,144,2,230,204,200,192,8,208,183,96
DL 30390 DATA 169,0,133,212,162,8,70,186,144,3,24,101
GH 30410 DATA 187,106,102,212,202,208,243,133,213,96,0,1,28

```

UŽIVATELSKE PROGRAMY

RAMHARD

Vážení čtenáři, dostal-li se Vám do rukou program RAMHARD, máte k dispozici pomůcku, kterou, pracujete-li s ATARI Basicem, jistě oceníte.

Velkou nevýhodou všech, kteří nemají možnost pracovat s disketovou jednotkou, je relativní pomalost a zdlocuhavost přístupu k basicovým programům, ukládání jejich částí a pod. Majitelé počítačů řady 130 XE pak také většinou nevyužívají dalších 64 KB volné paměti. Možnost částečné náhrady za disketovou jednotku umožňují různé TOSy (viz. např. minulé číslo našeho zpravodaje; pozn. red.), které zpřístupňují tzv. ramdisk ve velikosti 17 až 22 KB. Pro počítače řady 130 XE existují programy RAMDISK XE 130, které umožňují využít druhých 64 KB paměti RAM. Všechny tyto programy mají určité nevýhody:

- použití TOS vyžaduje I/O operace bas. programů v upraveném modu (zařízení "T:"), případně ve standardní rychlosti
- použití Ramdisku 130 XE pak vůbec neumožňuje I/O operace jinak, než ve standardní rychlosti
- navíc Ramdisk 130 XE publikovaný v časopise Elektronika má jisté chyby.

Všechny uvedené nevýhody odstraňuje program RAMHARD, který můžete používat jak na počítačích 800 XL/XE, tak i na ATARI 130 XE, spolu s magnetofonem s úpravou TURBO 2000 resp. SUPERTURBO.

Program RAMHARD zachovává všechny vlastnosti ATARI Basicu včetně možnosti I/O operací ve standardní rychlosti. V paměti je uložen na adresách \$0700 až \$1200 a posouvá začátek paměti pro Basic o asi 2,8 KB. Program nelze používat s disketovou jednotkou!

Verze 4.0

Po zavedení do počítače pomocí jakéhokoli běžného zavaděče TURBO 2000 program zjistí, jedná-li se o ATARI řady 800 či 130 XE. Podle typu počítače instaluje ramdisk. Jeho kapacita u 130 XE je asi 63 KB, u 800 XL/XE 4+10 KB. Ani u jednoho typu se ramdisk po studeném startu nesmaže!

S ramdiskem u 800 XL/XE lze používat jakékoli běžné I/O operace pro zařízení "R:" - LOAD, SAVE, LIST, ENTER, OPEN, CLOSE, GET, PUT, RUN, přičemž "R1:" je pro 1. ramdisk (4 KB) a "R2:" pro 2. ramdisk (asi 10 KB). Jméno nahrazeného programu (souboru) je ignorováno. Je možné uložit maximálně dva programy.

Ramdisk u 130 XE vznikl opravou ramdisku V. Zilvara, který byl publikován v časopise ELEKTRONIKA 6/89. Tam byl i detailně popsán. Do ramdisku můžeme uložit maximálně 17 souborů. Je možné používat všechny I/O operace pro zařízení "R:". Zde se soubory

odlišují jménem. Programy se běžně ukládají příkazem SAVE "R:název". Název může mít délku 12 znaků. Obsah ramdisku se vypíše po zadání příkazu DOS, smaže se při současném stisknutí kláves START a RESET. Lze mazat pouze celý ramdisk najednou.

Používání Turba je u obou typů počítačů stejné. Máme k dispozici zařízení "X:". Ve spojení s tímto zařízením můžeme používat pouze příkazy LOAD "X:" a SAVE "X:".

Příkaz LOAD "X:" umožní nahrát jakýkoli běžný basický program uložený na kazetě v SUPERTURBu (včetně TURBO 2000). Po zadání tohoto příkazu počítač 1x pípne. Dále pracuje jako běžný zavaděč. Po nahrání hlavičky testuje, jedná-li se o basický program. Jestliže ano, napíše zprávu "LOADING název". Název programu bude po celou dobu zobrazen uprostřed obrazovky. Jestliže se nejedná o basický program, je vypsáno chybové hlášení a počítač opětným zapípáním oznámí připravenost k nahrávání. Po nahrání se program ihned spustí.

Příkaz SAVE "X:" uloží z paměti počítače basický program na kazetu ve formátu TURBO 2000. Po zadání příkazu se počítač zeptá na název (max. 10 znaků) a 2x zapípá. Ukládání probíhá běžným způsobem. RAMHARD umožňuje více kopií. Jejich počet se znázorňuje znaky "<" za názvem. Takto vytvořený program lze pak kdykoli nahrát do počítače přes libovolný zavaděč (včetně RAMHARDu).

TURBO funkce se opouštějí stlačením RESET. Je třeba dbát na to, že podobně jako u CLOAD se po zadání příkazu LOAD "X:" stávající program vymaže!

Verze 4.1

Jedná se o verzi s dokonalejším ošetřením chybových situací. Jednou z nich je chyba uživatele, když před úpravou DL (Display List) nebyl zadán příkaz GRAPHICS. U předchozí verze 4.0 se tato chyba projevila buď zhroutením obrazovky, nebo celého systému. U verze 4.1 se tato chyba většinou neprojeví, a když tak pouze zhroutením DL. Zhroutený DL lze obnovit novou funkcí verze 4.1 a to stisknutím kláves HELP a RESET. Touto operací se jednak obnoví DL a dále také některé "životně důležité funkce.

Příjemnou práci s programem RAMHARD přeje

Ondřej FRYC

RAMTUR COPYER

Program RAMTUR COPYER je verzí známého kopírovacího programu LOADCOPY (od RICO). Umožňuje využít většinu jeho funkcí. Na rozdíl od LOADCOPY se v RAMTURU nedají spouštět dvoudílné a "ramdiskové" úpravy souborů a některé programy, které využívají paměť v místě jeho uložení (v RAM pod operačním systémem). Tyto nevšody jsou však vyváženy použitím ramdisku počítačů 130 XE

(a počítačů s rozšířenou pamětí RAM - slučitelných se 130 XE) pro ukládání více souborů. U ATARI 130 XE je k dispozici 65 280 bajtů. Konfigurace RAMTURu probíhá automaticky podle skutečné velikosti ramdisku. Na počítačích 800 XL/XE je vhodnější program LOADCOPY.

Následující návod k použití předpokládá znalost programu LOADCOPY, z jehož manuálu v podstatě vychází.

Po nahrání a spuštění programu se zobrazí pracovní obrazovka programu:

RAMTUR COPYER BY U.K. FOR RAMDISK.	
	Speed: LO. [Bd] : SA. [Bd] :2270
	FROM :\$ BYTES:\$ START:\$
	1 2 3 TYPE
	L LOAD S SAVE R RUN M MOTOR
	↑ UP ↓ DOWN SPACE SELECT CLEAR RAMDISK DELETE RENAME
Ramdisk bytes:\$00FF00	

Je rozdělena do čtyř polí. V horním je hlavička programu. V levém prostředním poli se zobrazují názvy souborů a kurzor. V pravém prostředním poli jsou jednak hlášení o rychlostech turbo nahrávek, dále je zde popsán význam jednotlivých kláves. V dolním poli je aktuálně zobrazována velikost volné paměti v ramdisku.

Funkce kláves

- 1,2 a 3 - přepínají pracovní režim
- L - nahrávání z kazety
- S - ukládání na kazetu
- R - spuštění programu
- M - spuštění a zastavení motoru
- šipky - ovládání kurzoru
- mezerník - označení souboru, u kterého je kurzor
- CLEAR - zrušení označených souborů nebo celého ramdisku
- DELETE - přejmenování souborů označených nebo všech
- SELECT - nastavení záznamové rychlosti

Po nahrání souboru se jednak zobrazí záznamová rychlost nahrávky, jednak adresy ukládací a startovní, dále délka souboru.

Program RAMTUR může pracovat ve třech režimech. Informace o nastaveném režimu je zobrazena inverzní číslicí před slovem "TYPE" v pravém prostředním poli.

Režim 1

Je nastaven automaticky po spuštění programu. K práci používá pouze operační paměť bez ramdisku. Je možné nahrát pouze jeden soubor. Je jediným režimem, který pracuje na počítačích 800 XL/XE. Je také vhodný pro vyhledávání konce posledního souboru (místa pro pokračování v nahrávání).

Režim 2

Tento režim po stisku klávesy "L" nahraje jeden soubor do operační paměti a dále jej překopíruje do ramdisku. Potom zobrazí kurzor. Soubory můžeme nahrávat tak dlouho, dokud máme volný ramdisk. Názvy souborů jsou zobrazeny v levém prostředním poli. Nyní můžeme označené soubory uložit, přejmenovat, zrušit nebo spustit. Soubor označíme tak, že k jeho názvu posuneme kurzor a stiskneme mezerník. Souborů na přejmenování, ukládání nebo mazání může být označeno několik najednou.

Při přejmenování se k označeným souborům přesune kurzor a v dolním poli je požadováno nové jméno. Pokud stiskneme pouze RETURN, jméno se vymaže avšak soubor v paměti zůstane.

Při mazání jsou zrušeny všechny označené soubory, přičemž je také od těchto souborů uvolněn ramdisk.

Při ukládání jsou označené soubory nahrávány na kazetu najednou.

Spustit je možné pouze jeden označený soubor. Ze spuštěného programu (souboru) je možné vrátit se zpět do RAMTURU. Je třeba přejít do basicu (např. pomocí studeného startu - bez vypnutí počítače) a zadat příkaz: X=USR(64647). Pokud bude RAMTUR zbořen, nahrajte jej znovu do počítače a přepněte na režim 2. Soubory v ramdisku budou opět přístupné.

Všechny výše popsané funkce je možné přerušit stiskem RESET. Obsah ramdisku zůstane zachován.

Režim 3

Na rozdíl od režimu 2, který pracuje se soubory jednotlivě, pracuje tento režim vždy se všemi soubory v ramdisku. Po stisku klávesy "L" probíhá nahrávání tak dlouho, dokud se nenaplní ramdisk. Vadný soubor bude označen hlášením "Load error", zvukovým znamením, a bude ignorován. Po naplnění ramdisku se kurzor nezobrazuje. Po volbě SAVE budou uloženy všechny soubory v ramdisku, zvolíme-li přejmenování, musíme přejmenovat všechny soubory. Stiskem klávesy "CLEAR" z ramdisku všechny soubory vymažeme. Protože v tomto režimu nemůžeme žádný soubor označit, nelze ani žádný soubor spustit.

Přeji vám příjemnou práci s RAMTUREM.

Milan Hašek

JAK NA TO?

ČESKÝ "BEJZIK"

Tato rubrika bývá věnována počítačovým hrám. Bohužel, redakce v současné době nemá k dispozici žádný originální příspěvek, takže tentokrát budete muset popis hry oželet. Avšak hraní zůstaneme věrni i tentokrát. Budeme si hrát s BASICem.

Jsou mezi počítačovými nadšenci jedinci, kterým vadí, že k nim počítač "nehovoří" jejich mateřským jazykem. A vadí jim to nejen v případech uživatelských programů, ale i u programovacích jazyků. Dejme jim, a nejen jim, tedy možnost upravit ATARI BASIC tak, jak jim bude vyhovovat. A možná si rádi vyzkoušíte takovou úpravu BASICu i vy.

Pro vše zmíněné hraní s BASICem jsme připravili dva programy. První - "MODIFIKACE ATARI BASICU" slouží k provedení vlastních úprav, druhý - "Ukladač modifikovaného BASICU" vám, jak již jeho název napovídá, umožní uložit modifikovaný BASIC na kazetu. Nahrajte do počítače program TOS 4.1 a přepněte na BASIC. Oba dva programy pečlivě přepište do počítače a uložte. Druhý - "Ukladač ..." - je třeba mít k dispozici uložený příkazem: SAVE "D:název". Pro svou práci potřebuje prostředí TOSu 4.1.

Máte-li v počítači TOS 4.1 a program "Modifikace ...", můžeme začít upravovat. Úprava proběhne ve třech fázích: úprava tabulky názvů příkazů - výroků (GOTO, LIST, PRINT, ...), úprava tabulky názvů funkcí a operátorů (SIN, CHR\$, TO, ...), a nakonec úprava hlášení (READY a pod.). Pokud některou z těchto oblastí nechceme upravovat, můžeme ji přeskočit.

Po spuštění programu budeme upravovat nejprve tabulku výroků. Na otázku počítače stiskneme "A". Vypíše se nám první příkaz BASICU - "REM" a vedle něj "ROZDIL=0". Rozdíl reprezentuje součet rozdílů délek původních a nových názvů příkazů. Nyní zadáme nový název příkazu "REM" a stiskneme RETURN. Vypíše se nám další příkaz. Takto můžeme pokračovat až máme upraveny názvy všech příkazů. Posledním příkazem je "CLOAD". Pokud místo zadání nového názvu pouze stiskneme RETURN, další příkazy již modifikovány nebudou.

Nutným požadavkem z hlediska funkce upraveného BASICu je nulová hodnota "ROZDIL". Jednotlivé nové názvy příkazů mohou mít délky jiné než názvy původní, avšak všechny nové názvy musí mít dohromady stejnou délku, jako názvy původní. Jestliže tedy některým příkazům na délce přidáte, druhým musíte ubrat.

Dalším omezením vaší fantazie je požadavek, aby název jednoho příkazu nebyl zleva podmožinou názvu jiného příkazu. Příklad: přiřadíte-li příkazu RESTORE název NASTAV, nesmíte již žádnému dalšímu příkazu přiřadit takový název, který by začínal slovem NASTAV.

Při zadávání nových názvů je vhodné zaznamenat si je k názvům původním, abyste s upraveným BASICem dokázali pracovat.

Po modifikaci tabulky výroků, následuje modifikace tabulky funkcí a operátorů (pokud ji nepřeskočíte zápornou odpovědí na otázku počítače). Zde postupujeme naprosto stejně, jako v předchozím případě. Platí i stejná omezení. Začínáme funkcí GOTO (jedná se o funkci za příkazem "ON") a končíme funkcí STRIG. Doporučujeme neměnit názvy operátorů jako jsou "+-/*., apod."

Uprava BASICu končí modifikací hlášení počítače. Zde můžeme zadat i kratší názvy než původní. Delší však ne.

Jakmile jsme provedli všechny úpravy, program se s námi rozloučí svým zpomaleným vylistováním. Při tom již budou použity vámi zadané nové názvy.

Pokud si budete chtít modifikovaný BASIC uložit na kazetu, připravte si program "Ukladač ..." (uložený příkazem SAVE) a nahrajte jej do počítače příkazem RUN "D:název", přesněji vašim! ekvivalentem tohoto příkazu. Krátkou chvíli po nahrání a spuštění Vás počítač požádá o název a po jeho zadání dvakrát zahouká. Založte do magnetofonu kazetu, na kterou chcete upravený BASIC uložit, připravte mgf. k ukládání a stiskněte kteroukoli klávesu. Upravený BASIC bude uložen ve dvou blocích. Ukládání můžete pro jistotu ještě zopakovat.

Takto uložený BASIC pro svou práci opět potřebuje TOS 4.1. Do počítače se nahrává funkcí "L" TOSu. Po nahrání hlavičky a prvního bloku stiskněte "Y". Do počítače se nahrává i druhý blok. BASIC se spustí a ohlásí nápisem "MUJ BEJZIK" a vašim ekvivalentem hlášení "READY".

Vámi upravený BASIC neubírá místo v normálně využitelné paměti, protože je umístěn v paměti RAM na stejných adresách, jako standardní ATARI BASIC. Máte k dispozici i ramdisk TOSu v jeho původní velikosti. Z vašeho BASICu je možné bez ztráty programu přejít do ATARI BASICu příkazem POKE 54017,253. Zpět do vašeho se dostanete příkazem POKE 54017,255. Dále je třeba mít na paměti, že programy uložené příkazem LIST "D:název" a načítané příkazem ENTER "D:název" nebudou mezi jednotlivými upravenými verzemi "bejziků" přenosné. Přenosné budou pouze programy ukládané příkazy SAVE a CSAVE.

Přeji hodně zábavy!

Jiří Skála

```

KI 1 REM *****
KI 2 REM * MODIFIKACE ATARI BASICU, (C) AK-602, 1990 *
KI 3 REM *****
KI 4 REM
RO 100 REM NACTENI STROJOVYCH RUTIN
IC 110 RESTORE 10010:I=1536
DR 120 READ A:IF A<>-1 THEN POKE I,A:I=I+1:GOTO 120
CI 200 REM BASIC Z ROM DO RAM
IE 210 POKE 559,0:X=USR(1536):POKE 54017,255:POKE 559,34
QT 220 CLR
GS 300 REM TABULKA UYROKU
TU 310 DIM A$(3000),B$(100),C$(100)

```

```

KK 320 ? "MODIFIKOVAT TABULKU VYROKU? ... A/N":INPUT C$:IF C$(">)"
  A" THEN 500
OU 330 REM NOVA SLOVA
LI 340 POC=42143:ROZDIL=0:?:?:?
MY 350 A$(LEN(A$)+1)=CHR$(PEEK(POC)):A$(LEN(A$)+1)=CHR$(PEEK(POC+
  1)):POC=POC+2
PE 360 GOSUB 4000:IF C$="" THEN 400
OI 380 IF B$="CLOAD" THEN 400
OK 390 GOTO 350
RZ 400 REM ZAPSAT TABULKU DO RAM
YY 410 IF ROZDIL(<)0 THEN GOSUB 5000:GOTO 320
AA 420 X=USR(1581,ADR(A$),42143,LEN(A$))
MI 500 REM TABULKA FUNKCI A OPERATORU
UY 510 ? :? "MODIFIKOVAT FUNKCE A OPERATORY? ... A/N":INPUT C$
GE 520 IF C$(">)"A" THEN 700
PJ 525 REM NOVA SLOVA
AU 530 A$="":POC=42981:ROZDIL=0
SB 540 GOSUB 4000
WZ 550 IF C$="" THEN 580
SI 560 IF B$="STRIG" THEN 580
PE 570 GOTO 540
NU 580 REM ZAPIS DO RAM
BB 590 IF ROZDIL(<)0 THEN GOSUB 5000:GOTO 510
JY 600 X=USR(1581,ADR(A$),42981,LEN(A$))
TZ 700 REM HLASENI
WT 710 ? :? "MODIFIKOVAT HLASENI? ... A/N":INPUT C$:IF C$(">)"A" T
  HEN 1010
XT 730 ? :? "READY ... ":INPUT C$:IF C$="" THEN 780
AD 750 IF LEN(C$)>5 THEN ? "MOC DLOUHE!":GOTO 730
UH 760 IF LEN(C$)<5 THEN X=5:GOSUB 6000:GOTO 770
YF 770 FOR I=4 TO 0 STEP -1:POKE I+48499,ASC(C$(5-I,5-I)):NEXT I
DX 780 ? :? "STOPPED ... ":INPUT C$:IF C$="" THEN 830
JR 800 IF LEN(C$)>7 THEN ? "MOC DLOUHE!":GOTO 780
RH 810 IF LEN(C$)<7 THEN X=7:GOSUB 6000:GOTO 820
YY 820 X=USR(1581,ADR(C$),42493,7)
KY 830 ? :? "ERROR ... ":INPUT C$:IF C$="" THEN 880
BK 850 IF LEN(C$)>5 THEN ? "MOC DLOUHE!":GOTO 830
WZ 860 IF LEN(C$)<5 THEN X=5:GOSUB 6000:GOTO 870
YD 870 X=USR(1581,ADR(C$),42485,5)
ZK 880 ? :? "AT LINE ... ":INPUT C$:IF C$="" THEN 1010
LK 900 IF LEN(C$)>7 THEN ? "MOC DLOUHE!":GOTO 880
SZ 910 IF LEN(C$)<7 THEN X=7:GOSUB 6000:GOTO 920
ZR 920 X=USR(1581,ADR(C$),47525,7)
FY 1000 REM KONEC
OG 1010 CLR :POKE 622,255:?"K":LIST :END
CY 4000 REM SUB ZADAVANI SLOV
GH 4010 B$="":C$=""
PU 4020 A=PEEK(POC):IF A<128 THEN B$(LEN(B$)+1)=CHR$(A):POC=POC+1
  :GOTO 4020
MO 4030 B$(LEN(B$)+1)=CHR$(A-128):POC=POC+1
EU 4040 ? B$,"ROZDIL=":ROZDIL:?"MOVE SLOVO: ":INPUT C$
IS 4050 IF C$="" THEN RETURN
BI 4060 POM=LEN(C$):ZNAK=ASC(C$(POM,POM)):C$(POM,POM)=CHR$(ZNAK+1
  28):ROZDIL=ROZDIL+POM-LEN(B$):A$(LEN(A$)+1)=C$
AK 4070 RETURN
AJ 5000 REM SUB SLOVA SPATNE ZADANA

```

```

KX 5010 ? :? "CHYBA!!!":? "ROZDIL MENI NULOY!"?:? "ZNOVU!"
SI 5020 FOR DEL=1 TO 2000:NEXT DEL
AI 5030 RETURN
RA 6000 REM SUB DOPLNENI MEZER
UG 6010 FOR I=LEN(C$)+1 TO X:C$(I,I)=" ":NEXT I:RETURN
BO 10000 REM DATA STROJOUYCH RUTIN
NR 10010 DATA 104,169,0,133,212,169,160,133,213,169,253,141,1,211
,160,0,177,212,72,169,255,141,1,211,104,145,212,230
KU 10020 DATA 212,208,234,230,213,165,213,201,192,208,226,169,253
,141,1,211,96
OU 10030 DATA 104,104,133,205,104,133,204,104,133,207,104,133,206
,104,133,213,104,133,212,169
AY 10040 DATA 0,141,0,1,141,1,1,160,0,177,204,145,206,230,204,208
,2,230,205,230,206,208,2,230
VF 10050 DATA 207,238,0,1,208,3,238,1,1,173,1,1,197,213,208,225,1
73,0,1,197,212,208,218,96,-1

```

```

EX 1 REM *****
PK 2 REM * UKLADAC MODIFIKOVANEHO BEJZTKU, (C) AK-602, 1990 *
EY 3 REM *****
NJ 4 REM
LP 10 ? "UKLADAC MODIFIKOVANEHO BASICU!":DIM JMENO$(10)
GC 20 PIP=ADR("hhh [K+"):ULOZ=ADR("hhh <[ *"):FOR I=18192 TO 18496
:POKE I,0:NEXT I
BY 30 RESTORE 1010:FOR I=18192 TO 18496:READ A
PT 40 IF A=-1 THEN I=18319:GOTO 70
PK 50 IF A=-5 THEN I=18434:GOTO 70
NK 60 POKE I,A
IU 70 NEXT I
LV 80 RESTORE 1060:FOR I=1024 TO 1040:READ A:POKE I,A:NEXT I
GF 90 ? :? "NAZEV: ";:INPUT JMENO$:FOR I=1 TO LEN(JMENO$):POKE I+
1024,ASC(JMENO$(I,I)):NEXT I
AI 100 X=USR(PIP,2):POKE 50,0:POKE 51,4:POKE 53,4:X=US
R(ULOZ,0)
OS 110 POKE 50,16:POKE 51,71:POKE 52,65:POKE 53,72:X=USR(ULOZ,255
)
SA 120 POKE 50,0:POKE 51,160:POKE 52,0:POKE 53,192:X=USR(ULOZ,255
)
GS 130 ? :? "JESTE JEDNOU? ... [Y/N]":INPUT JMENO$:IF JMENO$="A" T
HEN 100
KO 140 GRAPHICS 0:END
OD 1000 REM DATA STROJOUYCH RUTIN
ZP 1010 DATA 162,13,189,242,32,149,128,202,16,248,162,255,154,169
,169,72,169,121,72,169,10,133,201,169,0,133,9,141
AX 1020 DATA 248,3,169,253,141,1,211,169,183,72,169,84,72,165,106
,16,2,169,160,133,106,76,148,239,0,-1
YQ 1030 DATA 169,0,133,50,133,52,169,160,133,51,169,192,133,53,16
9,255,141,1,211,32,239,20,76,0,32,0,-5
ZB 1040 DATA 32,0,33,0,33,1,33,1,33,43,33,49,33,0,1,0,42,18,40,15
,10,77,85,74,32,66,69,74,90,73,75,20,21,40,20,39,31
SN 1050 DATA 14,66,5,64,23,0,0,18,14,65,2,85,0,0,0,20,42,22,22,0,
128,6,6,52,22
OB 1060 DATA 3,32,32,32,32,32,32,32,32,32,32,0,32,49,1,128,32

```

SLABIKAR ATARISTY

TIPY PRO TOS 4.1

V prvním letošním čísle našeho zpravodaje byl uveden poměrně podrobný popis použití kazetového operačního systému TOS 4.1. Dnes vám o tomto programu nabízíme další užitečné informace a poznámky.

Copy data file

Funkce "Copy data file" - kopírování souboru - nemusí sloužit pouze ke kopírování souborů z kazety na kazetu, protože jako zdrojové a cílové zařízení můžeme použít téměř "cokoli"

Například pro vstup textu z klávesnice a jeho tisk rovnou na tiskárnu zadáme jako zdrojové zařízení "E:". Napíšeme text. Jako poslední znak textu musí být kód EOF, který obdržíme stisknutím kláves CTRL + 3. Jinou klávesou, kromě RESET se z obrazovkového editoru nedostaneme. Nyní uvedeme tiskárnu do provozu a zadáme cílové zařízení - "P:". Text se vytiskne. Při psaní textu nesmíme udělat chybu!

Majitelé tiskáren bez grafického modu, např. znakových, mají potíže s tiskem textů psaných v editorech ČAPEK a ČÍZEK. Tyto editory tisknou česká písmena grafickým režimem takže není možné na znakové tiskárny tisknout přímo. Nejdříve je třeba české znaky převést na standardní. Jsou tři možnosti.

Text vytiskneme editorem do ramdisku (zařízení "D8:"). Při tomto tisku proběhne potřebný převod českých znaků na standardní. Potom přejdeme z editoru do TOSu. Připravíme tiskárnu, z menu TOSu zvolíme "Copy data file". Zdrojové zařízení bude "D8:*", cílové pak "P:". Text bude vytisknut tak, jak jsme zadali, tj. podle formátovacích příkazů. Určitým omezením tohoto způsobu je nutnost použít pro tisk nekonečný (traktorový) papír. Soubor z ramdisku bude překopírován na tiskárnu bez čekání na zakládání jednotlivých listů papíru.

Pokud by byl text příliš dlouhý a nevešel by se do ramdisku, použijeme pro tisk kazetu. Text vytiskneme editorem na kazetu ("D:"), přejdeme do TOSu, zvolíme "Copy data file", zdrojové zařízení "D:název". Nahrajeme vytisknutý text zpátky do počítače, zadáme cílové zařízení "P:" a tak vytiskneme text. I v tomto případě je nutné použít nekonečný papír.

V případě, že nekonečný papír nemáme, je nutné provést převod českých znaků na standardní znaky speciálním programem (je k dispozici v našem klubu) a takto upravený text nahrát zpátky do editoru. Text lze potom z editoru normálně vytisknout (rovnou na tiskárnu) i na jednotlivé listy papíru.

Modify ramdisk

Do popisu této funkce TOSu ve zmíněném článku se nedopatřením "vloudily" určité nepřesnosti. Umístění ramdisku je možné měnit pouze po stránkách paměti, tj. zadává se pouze horní

bajt počáteční a koncové adresy ramdisku. Chceme-li mít ramdisk v oblasti např. \$CA00 až \$EFFF zadáme M a na otázku "Range:\$" odpovíme CAEF.

Změníme-li umístění ramdisku v paměti, stanou se všechna data, která v něm až do změny byla, pro TOS nedostupnými!

Modifikace LOMEM

LOMEM je proměnná, která ukazuje na první volnou adresu za TOSem. Podle ní se provádí inicializace Basicu, dále slouží při používání funkce "Load T 2000 file" - aby se nahrávaným souborem nepřepsal kus TOSu, a pod. Její modifikací je možné například vytvořit určitý chráněný prostor, do něhož je možné umístit buď pomocné strojové podprogramy nebo data (např. znakovou sadu). Obsahem LOMEM je dvoubajtová hodnota umístěná v TOSu dvakrát, a to na adresách \$0B0C a \$0B0E (2828 a 2830).

Z basicu provedeme změnu LOMEM posloupností příkazů POKE. Např. budeme chtít posunout LOMEM na adresu \$2000 (8192). Zadáme tedy:

```
POKE 2828,0:POKE 2830,0:POKE 2829,32:POKE 2831,32
```

Nyní příkazem DOS přejdeme do TOSu a přepneme na RAM. V menu se objeví hlášení: LOMEM : \$2000

Návrat do basicu

V normálních případech se z TOSu do basicu dostáváme funkcemi "Run cartridge" a "Basic/Ram". Může se však stát, že přejdeme do TOSu, tam spustíme nějaký podprogram a on se nám "zakousne". V tom případě nezbyvá nic jiného, než použití klávesy RESET. Tím se však zase může "zaseknout" basic. Proti takovým nemilým překvapením se můžeme zabezpečit tak, že ještě před opuštěním basicu přepíšeme hodnoty v registrech \$0C a \$0D příkazy:

```
POKE 12,PEEK(10):POKE 13,PEEK(11)
```

Názvy "D:"-souborů

Nevýhodou zařízení "D:" je, že není při nahrávání zobrazován název souboru. To je nepříjemné zvláště tehdy, musíme-li soubor na kazetě hledat. Tuto nevýhodu můžeme odstranit, zejména ve svých programech, následujícím postupem. Protože při nahrávání se název souboru ukládá od adresy \$0701 (1793) v délce 16 bajtů, stačí ihned po otevření kanálu přečíst těchto 16 bajtů a vytisknout je. V basicu to můžeme udělat např. takto:

```
10 DIM H$(20):OPEN #1,4,0,"D:":POKE 54018,60
20 FOR I=1793 TO 1808:H$(LEN(H$)+1)=CHR$(PEEK(I)):NEXT I
30 ? "Soubor :";H$:POKE 764,255
40 IF PEEK(764)=255 THEN 40
50 POKE 54018,52
60 zde pokračuje nahrávání
```

Okamžitě po otevření kanálu #1 pro zařízení "D:" je zastaven

motor, aby se pásek přestal pohybovat. Potom je do H# uložen název souboru a následně vytisknut. Nyní program čeká na stisk klávesy, abychom stačili název souboru přečíst. Jakmile stiskneme klávesu, je spuštěn motor a může proběhnout vlastní načtení souboru.

Load a Save T 2000 file

Tyto funkce jsou normálně použitelné pouze z TOSu. Mnohdy by se nám však hodily i v basicu. V tom případě však je postup složitější. Musíme v našem programu nahradit část TOSu, přesněji ty části, ve kterých se zadává co se má uložit na kazetu, nebo kam se má nahrát soubor z kazety a vyvolat vlastní nahrávací nebo ukládací rutiny, které jsou součástí TOSu.

Pro nahrávání z kazety slouží rutina TOSu na adrese \$14EF. Rutina vyžaduje, aby na adrese \$32 a \$33 byla uložena adresa počátku oblasti, kam se budou data ukládat. Dále na adrese \$34 a \$35 adresa konce této oblasti, zvětšená o 1. Před voláním rutiny musí být v akumulátoru procesoru buď "0" má-li být nahrána hlavička, nebo "255" má-li být nahrán vlastní blok dat. Protože z basicu nelze přímo ovlivňovat obsah registrů procesoru, musíme pro vyvolání rutiny použít krátký podprogram ve strojovém kódu:

```
NAHRAJ PLA ; počet parametrů ze zásobníku
        PLA ; vyšší bajt parametru - nepoužijeme
        PLA ; 0 nebo 255 (0 nebo $FF)
        JSR $14EF ;vyvolání nahrávací rutiny
        RTS ; návrat do basicu.
```

Tento podprogram pak voláme příkazem:

```
X=USR(NAHRAJ,0) ... pro nahrání hlavičky,
X=USR(NAHRAJ,255) ... pro nahrání bloku dat.
```

Pro ukládání na kazetu slouží rutina TOSu na adrese \$161E. Její použití je obdobné, jako u rutiny nahrávací. Na adresu \$32 a \$33 uložíme adresu počátku oblasti paměti, na adresu \$34 a \$35 adresu konce paměti zvětšenou o 1. Tak jako nahrávací, vyžaduje i tato rutina před vlastním vyvoláním v akumulátoru "0" pro ukládání hlavičky a "255" pro ukládání bloku dat. Pro vyvolání rutiny použijeme opět krátký podprogram ve strojovém kódu:

```
ULOZ PLA ; počet parametrů ze zásobníku
        PLA ; vyšší bajt parametru - 0
        PLA ; nižší bajt - 0 nebo 255
        JSR $161E ; vyvolání ukládací rutiny
        RTS ; návrat do basicu.
```

Podprogram vyvoláme basicovými příkazy:

```
X=USR(ULOZ,0) ... ukládání hlavičky,
X=USR(ULOZ,255) ... ukládání bloku dat.
```

Při ukládání hlavičky i bloku dat bude na jejich konec přidán jeden kontrolní bajt.

Chceme-li, aby to, co uložíme na kazetu bylo slučitelné se všemi nahrávkami TURBO 2000, musí mít hlavička následující strukturu:

1. bajt - druh souboru, většinou "3",
2. až 11. bajt - název souboru,
12. a 13. bajt - adresa, od které se má soubor uložit
14. a 15. bajt - délka souboru v bajtech
16. a 17. bajt - startovací adresa programu.

Při používání magnetofonu je zvykem, že se před spuštěním magnetofonu ozve bzučák a počítač čeká na stisk klávesy. To je možné realizovat basicem za použití příkazu SOUND a testování kláves, nebo dalším krátkým podprogramem ve strojovém kódu:

```
PIP PLA ; počet parametrů ze zásobníku
PLA ; vyšší bajt parametru "0"
PLA ; počet pípnutí do akumulátoru
JSR $FD9C ; rutina OS
RTS ; návrat do basicu.
```

Podprogram vyvoláme příkazy:

```
X=USR(PIP,1) ... při nahrávání z kazety,
X=USR(PIP,2) ... při ukládání na kazetu.
```

Všechny tři strojové podprogramy je možné umístit do textového řetězce.

Použití rutin TOSu 4.1 pro práci s TURBem 2000 demonstruje program, jehož vps je na konci tohoto článku. Jedná se o převodník souborů TURBO 2000 na binární soubor, který, mimo DOSu 2.5, vyžaduje např. i kazetový operační systém CTOS.

Různé

- TOS 4.1 je v paměti uložen v oblasti \$0B00 až \$1770 (2816 až 6000). Startovací adresa je \$0B00.

- Adresa zařízení "D:", přesněji adresa tabulky adres jeho obslužných programů je umístěna za písmenem "D" v tabulce handlerů a je \$1257. Některé programy přepisují hodnoty na adresách 10 až 13, což může zapříčinit, že po stisku RESET se v tabulce handlerů neobnoví zařízení "D:". Proto POZOR NA RESET u nevyzkoušených programů. Rutina, která po stisku RESET znovu inicializuje LOMEM a zařízení "D:" má adresu \$121A.

- Oblast paměti \$0700 až \$0AFF využívá zařízení "D:" jako buffer. Kdykoli toto zařízení použijeme, bude tato oblast přepsána. Nehodí se proto pro ukládání dat či podprogramů. Šestá stránka paměti je volná!

```

EJ 2 REM *****
OY 3 REM * PREUODNIK TURBO 2000 - FILE *
AB 4 REM *-----*
NO 5 REM * !Pracu je pouze pod TOS 4.1! *
EN 6 REM *****
NK 7 REM
CR 100 ? "K":SETCOLOR 2,0,0:? "PREUODNIK TURBO 2000 -> FILE"
UG 101 ? "-----+--+?"
HV 105 MEM=31000
HE 110 PIP=ADR("hhh Ⓚ"):NAHRAJ=ADR("hhh Ⓚ):ULOZ=ADR("hhh Ⓚ
")
NB 120 DIM FILE$(20),ZAC$(6),KON$(6),NAME$(20),BUF$(MEM)
VT 125 KON$="Ⓚ Ⓚ":ZAC$(1,1)=CHR$(255):ZAC$(2,2)=CHR$(255)
NO 126 BUF$(1,1)=CHR$(0):BUF$(MEM,MEM)=CHR$(0):BUF$(2)=BUF$
VA 127 NAME$(1,1)=CHR$(0):NAME$(20,20)=CHR$(0):NAME$(2)=NAME$
OB 130 NAM=ADR(NAME$):NAMK=NAM+17:BUF=ADR(BUF$):BUFK=BUF+MEM
EK 200 REM NAHRANI HLAVICKY
KV 210 POKE 51,INT(NAM/256):POKE 50,NAM-256*PEEK(51):POKE 53,INT(
NAMK/256):POKE 52,NAMK-256*PEEK(53)
ND 219 K=USR(PIP,1)
UY 220 X=USR(NAHRAJ,0):? "PROGRAM: ";NAME$(2,11):?
NT 225 X=USR(PIP,1)
SA 229 REM NAHRANI BLOKU
ZD 230 POM1=ASC(NAME$(14))+256*ASC(NAME$(15)):POM=BUF+POM1
SE 235 IF POM1>MEM THEN ? "PRILIS DLOUHY SOUBOR!!!":STOP
ER 240 POKE 51,INT(BUF/256):POKE 50,BUF-256*PEEK(51):POKE 53,INT(
POM/256):POKE 52,POM-256*PEEK(53)
UR 250 X=USR(NAHRAJ,255)
VD 300 REM TVORBA FILE
HB 310 ZAC$(3,4)=NAME$(12,13):KON$(5,6)=NAME$(16,17):KAM=ASC(NAME
$(12))+256*ASC(NAME$(13)):KAM=KAM+POM1-1
GA 320 KAMH=INT(KAM/256):KAML=KAM-256*KAMH:ZAC$(5,5)=CHR$(KAML):Z
AC$(6,6)=CHR$(KAMH)
UR 400 REM ULOZENI FILU
EK 410 ? :? "Zarizeni:nazev: ";INPUT FILE$
NO 420 OPEN #1,8,0,FILE$:? #1:ZAC$:BUF$(1,POM1):KON$:CLOSE #1
GA 500 ? :? "DALSI SOUBOR...Ⓚ/Ⓚ";INPUT NAME$
ST 510 IF NAME$="A" OR NAME$="a" THEN CLR :GOTO 105
KO 520 GRAPHICS 0:END

```

=====

ATARI, technický zpravodaj pro mikroelektroniku a výpočetní tech-
niku. Redaktor: Jiří Skála. Adresa, redakce: Čs. hifiklub, Wintrova 8,
160 41 Praha 6. Telefon: 341 409. Povoleno ÚVTEI pod ev. číslem 87006.
Náklad: 2000 výtisků, cena 5,50 Kčs. Vytiskla tiskárna Československé-
ho hifiklubu Praha 6, Wintrova 11.

Praha, červen 1990

Igi/2024