

T A P E L O A D I N G E R R O R

Situace vypadá beznadějně. Podarilo se vám znicit začátek programu nahrávaného na magnetofonovém pásku. Vypis není a spousta hodin stravených psaním programu je v pytli (jen zkušenosti ne). Zničena je i hlavička i zaváděcí signál i začátek programu.

Byla to chvilka klete nepozornosti nebo vypnutí proudu v době nahrávání na stejné místo, kde byla předchozí verze tohoto programu? Zbytek programu na pásku přece fyzicky existuje. Není nějaká možnost toho využít? Nebo snad, thousand devils, budete muset všechno datlovat znova?

Samozřejmě, že ne! Pokud se se svým počítačem dobře znáte a oprava je natolik důležitá, že se vyplatí obětovat víc času.

Na začátek několik obecných informací o tom, jak Spectrum spolupracuje s magnetofonem. Rutina, která čte a verifikuje libovolný soubor na pásku je v ROM na adrese #0556. Před jejím voláním je třeba do registru IX vložit počáteční adresu nahrávaného bloku, do DE jeho délku a do akumulátoru typ bloku: 0=hlavička, 255=vlastní blok. Není-li nastavena vlajka přenosu jde o verifikaci. Na páse je vždy nahráno o dva bajty víc, než je délka deklarovaná v registru DE. Na začátku je navíc zapsána hodnota registru A, což umožňuje overení typu bloku a na konci je paritní bajt vytvořený průběžně funkcí XOR ze všech nahrávaných bajtů. Tím je zajištěno konečné overení správnosti načítaného bloku.

V průběhu čtení bloku do počítače se kontroluje správnost čtení počítáním hodinových taktů mezi následnými změnami napětí přicházejícího na port #PE. Rozdíl času, který překročí 2400 taktů má za následek oblíbenou zprávu "Tape loading error" již v průběhu LOAD. N e n i zadná další kontrola, zda aktuální dvojice hran reprezentující 0 nebo 1 patří k předcházejícímu nebo už dalšímu bajtu. Po odečtení 8 bitů počítač očekává dalších osm, atd. až do dosažení požadované délky bloku. Problém je v tom, že popsána procedura LOAD požaduje na začátku zaváděcí signál s větší vzdáleností mezi hranami a speciální synchronizační impuls před vlastním čtením bajtu. A to je bohužel zničeno.

Můžeme vynechat část této rutiny a přejít hned ke čtení následujících bitů? Ano, ale nemáme žádnou jistotu zda čtení začne od začátku bajtu! I s tímto problémem se alespoň částečně vyrovnáme - ale později.

Nyní se podíváme na krátký program v assembleru, který umožní využít rutinu LOAD k našemu účelu.

```
ADRESA EQU   adresa       ; zde se začne ukládat čtený blok
        ORG   55000        ; začátek naší rutiny
        LD   IX,ADRESA    ; uložení adresy začátku bloku do IX
        LD   DE,#FFFF     ; maximální délka - stejně jí neznáme
        LD   C,#06        ; nastavení barvy okraje
```

```

XOR  A           ; nastaveni vlajky Z, nebude se overovat
                        ; parita prvnio bajtu
SCF                    ; nastaveni vlajky prnosu, bude LOAD
EX   AF,AF'          ; uschovani vlajek
DI                    ; zakaz maskovatelného preruseni
LD   HL,NAVRAT       ; ulozeni navratove adresy z LOAD rutiny
PUSH HL              ; na zasobnik
JP   #05A3           ; skok na cteni prvnio bitu
NAVRAT LD HL,#FFFF    ; vypocet mnozatvi prectenych bajtu
SBC  HL,DE           ;
LD   B,H             ; prevedeni vypoctene hodnoty
LD   C,L             ; do registru BC
EI                    ; opet povolit preruseni
RET                   ; a zaverecny navrat

```

Rutinu lze vytvorit nasledujicim programem:

```

10 CLEAR 29999: LET ADRESA=30000
20 FOR N=0 TO 28: READ A: POKE 65000+N,A: NEXT N: STOP
100 DATA 221,33,ADRESA-256*INT(ADRESA/256),INT(ADRESA/256)
110 DATA 17,255,255,14,6,175,55,8,243,33,240,255,229,195,
        162,5
120 DATA 33,255,255,237,82,68,77,251,201

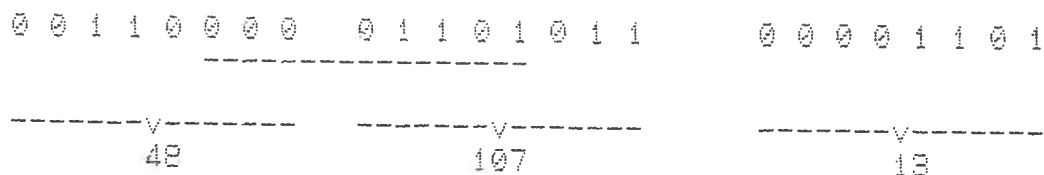
```

Po instalaci rutiny nastavime magnetofon na místo o kterém jsme presvedceni, ze na nem zacina neposkozena cast souboru. Pak magnetofon zapneme a odesleme PRINT USR 65000. Dulezite je toto poradí - n e j p r v e zapnout magnetofon a p a k spustit rutinu. Je to proto, ze program okamzite zacina cist bity a v pripade nezapnutého magnetofonu se vrati zpet do BASICu. Je take mozne, ze se netrefi do casove pauzy nahrane na pasku, coz take znamena okamzity navrat do interpretéru. Neklnte nepriznivému osudu a bez skripeni chrupu vrate pasek zpet opakujte cely postup znovu ...

Po uspesnem ukonceni nahravani se na obrazovce objevi cislo udavajici pocet prectenych bajtu.

Vratme se nyne k problemu spravneho umisteni bitu v prectenych bajtech. Nejjednodussi situace je v pripade BASICoveho programu nahraneho bez promennych (jeste jednoduchssi v pripade textoveho souboru tsswordu). Zde vime, ze posledni bajt ma hodnotu 13 (ENTER). Tato hodnota je na konci kazdeho programoveho radku.

Maly prikklad - posledni dva nactene bajty jsou 48 a 107. Pozice bitu v techto bajtech a v bajtu o hodnotě 13 jsou nasledujici:



Podtrzení označuje místo bajtu s hodnotou 13 ve dvou posledních bajtech. Z toho je zřejmé, že by bylo potřeba přesunout poslední bajt o tři bity vpravo a tři nejvýznamnější bity přesunout z bajtu předcházejícího.

Funkci přesunu jednoduše realizujeme dělením bajtu osmi. K přesunutí všech bajtů zachranovaného programu staci napsat krátký program:

```
1000 LET ocasek=@: FOR n=adresa TO adresakonce
1010 LET prenasenebity=INT(PEEK n/8)
1015 LET novyocasek=PEEK n-8*prenasenebity
1020 POKE n,82*ocasek+prenasenebity
1025 LET ocasek=novyocasek
1030 NEXT n
```

Jeste je ovšem nutné stanovit regulérní začátek našeho nestastného souboru. Musíme hledat od začátku a opět hledat hodnotu 13. Za ni musí následovat dvouбайtová hodnota - číslo programového řádku. Hodnota 13 se ovšem ještě může vyskytnout uvnitř petibajtové reprezentace čísla v BASICu po bajtu 14.

Lokalizujeme-li úspěšně správný začátek řádku nahrajeme takto získaný modul na pásku jako binární soubor - SAVE "název" CODE adresa_zacatku,delka. Pak už jen zbyva tento blok natahnout do COPY COPY a změnit informace uložené v hlavičce:

1. do prvního bajtu vložíte @ (BASIC)
2. do 14. bajtu číslo 32768 (žádný startovací řádek)
3. do 15. a 16. bajtu délku modulu

Po opětovném nahrání modulu máme k dispozici použitelný fragment našeho programu.

Bohužel v případě ostatních druhů zaznamu - datová pole, strojové programy a BASICové programy nahrané spolu s promennými - nemáme k dispozici žádné informace o hodnotě posledního bajtu. Co teď?

Je sedm možností přesunu bitu v získaných bajtech. Který z nich je ten pravý můžeme zjistit jen hloubáním nad získanými výsledky po jednotlivých přesunech a řídit se přitom intuicí, zkušeností a znalostí toho, co by asi v daném modulu mělo být.

Je tedy zřejmé, že je třeba uvážit, má-li cenu tímto způsobem program zachránovat a nejsou jiné schůdnější možnosti (koléga, napsat znovu a samozřejmě lépe ap.),

WOJCIECH INDYK
BAJTEK 2/88

TRANSLATE & TABWORD DATLING NDT (C) TOPSOFT KRALUPY/VLT