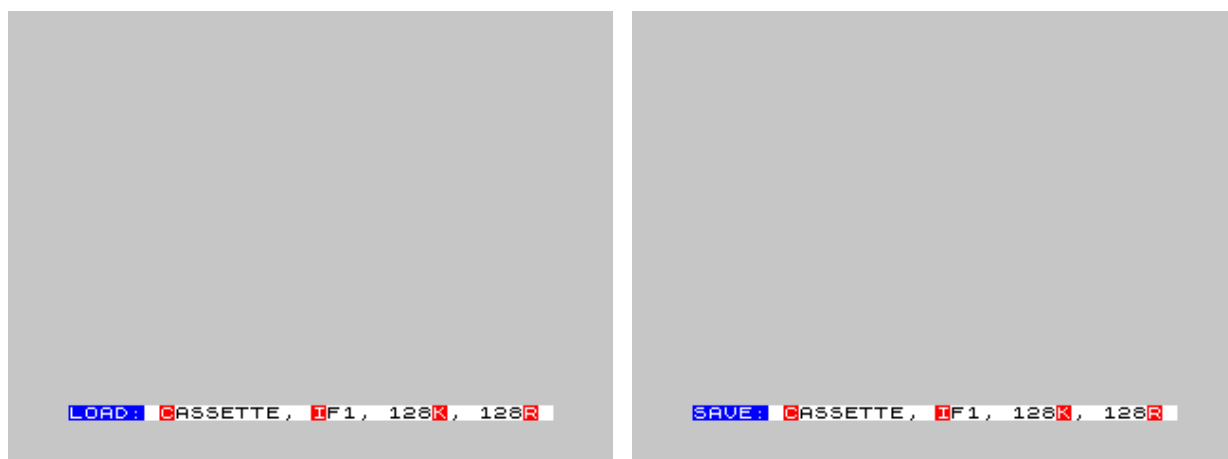


Linuxové TAPkování a TRDování přes sériové porty

Nevím přesně, kdy se objevila modifikovaná 48 ROM-RS232 od Paula Farrowa na jeho velmi zajímavém webu <http://www.fruitcake.plus.com>. Vím ale přesně, že mě nové možnosti, které se s touto ROMkou naskytly, velmi potěšily. Modifikace ROMky spočívá v doplnění LOAD / SAVE rutiny o možnost přenášet data sériovým portem. Na výběr je sériový port ZX Spectra 128K, sériové rozhraní Keypadu nebo RS-232 port v ZX Interface1. Zvlášť lze specifikovat port pro LOAD i SAVE. Nabízí se tak možnost využít sériový port pro přenos dat mezi spectry, ze spectra do PC a naopak. Lze také dostat data z PC na fyzickou kazetu a nebo častěji ze starých ošoupaných spektráckých kazet do PC.

Při prvním pokusu o LOAD nebo SAVE se v nové verzi (zkoušena v3.01) objeví menu s výběrem portu. Volby jsou: C pro kazeták, I pro Interface1, K pro Keypad u ZX 128K, R pro RS232 u ZX 128K. Při dalším použití LOAD či SAVE se použije předtím zvolená možnost.

Nastavením nevyužitých systémových proměnných (NMIADD – adresa skoku po NMI) na adresách 23728 (\$5CB0) pro LOAD, VERIFY nebo MERGE a 23729 (\$5CB1) pro SAVE lze nastavení použitého portu změnit. Hodnoty systémových proměnných mohou nabývat hodnot 'C', 'I', 'K' nebo 'R', stejně jako při výběru z menu. Hodnota 0 způsobí opětovné zobrazení menu. Dřívější verze ROM neměly toto menu a muselo poukovat. Pro více informací o této ROM navštivte stránky autora: http://www.fruitcake.plus.com/Sinclair/Interface2/Cartridges/Interface2_RC_New_RS232.htm



ROM-RS232 menu s výběrem portu pro přenos dat při LOAD a SAVE.

Na stránkách Paula Farrowa je ke stažení i program ZX Tape Player pro Windows, který spolupracuje s touto modifikací ROM. Umí navíc pracovat se soubory pro ZX80, ZX81 a Jupiter Ace. Program funguje dobře na Windows s .NET Frameworkem 3.5 což se na Linuxu (při zachování zdravé mysli) provozovat nedá. Bylo rozhodnuto, že je třeba napsat obdobný program i do Linuxu.

V Linuxu lze se sériovým portem pracovat obstojně i v shellu a opravdu funguje přenos tapek z i do spectra přes pouhý cat na správný sériový port (i USB-serial převodník), za předpokladu správného nastavení portu.

```
stty -F /dev/ttyUSB0 57600 cs8 clocal cread -cstopb parenb -parodd crtscts raw
```

Nastavení portu ze stránek http://cygnus.speccy.cz/popis_siftrd.php

Přestože ZX Spectrum používá hardwarové řízení toku dat, mohou nastat situace, že z PC do spectra odejdou ještě další data potom, co spectrum pomocí CTS vodiče zakáže další vysílání. Může

tak dojít ke ztrátě dat. Pravděpodobně to má na vině fifo uartu, fifo usb-serial převodníku nebo jiný buffer v operačním systému. A to je další důvod pro vznik specializovaného programu pro tapkování přes sériové porty. Program musí před každým vyslaným bajtem do spectra zkontrolovat stav signálu CTS o tom, že spectrum je připraveno pro příjem dalšího bajtu.

Rychlost sériového portu není přece jen blesková i při 57600(!) baudech, které zvládne tato ROM softwarově obsloužit, vezmeme-li navíc v úvahu, že sériové porty jsou v ZX Spectru 128K standardně připojeny přes IO piny zvukového obvodu AY-3-8912. Z to vyplývá požadavek na další vlastnost, dlouho v todo listu napsanou, a tou je zobrazení průběhu odesílání či přijímání dat. To by se catem dělalo hůře, snad např. pomocí pv (pipe vieweru).

Jako výchozí bod pro tvorbu programu pro přenos tapek po sériovém portu do spectra byla použita utilita „**tapheader**“ od Mika ze Zeroteamu. Dostupný je na adrese <http://zeroteam.sk/tapheader.html>. Předhozením tap souboru této utilitce vypíše vše, co v tapce je. Díky dostupnosti zdrojového kódu stačilo přidat nastavení sériového portu, několik zápisů na sériový port a funkci kontrolující stav CTS. Tak vznikla první verze programu **taptoser**.

```
./taptoser -d /dev/ttyUSB0 -b 57600 LSD.TAP
Serial device: /dev/ttyUSB0, communication speed is: 57600 Bd

Filename:  LSD
Flag:      0 Type: 0 => program
Program length: 820 bytes Runs from line 1 Length without variables: 820 bytes
Proceed bytes: 822/ 822 [=====] 100 %

Filename:  LSD.0
Flag:      0 Type: 3 => bytes Start address: 50000 Length: 2973 bytes 3rd param: 8224
Proceed bytes: 2975/ 2975 [=====] 100 %

Filename:  LSD.1
Flag:      0 Type: 3 => bytes Start address: 32768 Length: 15080 bytes 3rd param: 8224
Proceed bytes: 15082/ 15082 [=====] 100 %

Filename:  LSD.2
Flag:      0 Type: 0 => program
Program length: 413 bytes Runs from line 10 Length without variables: 413 bytes
Proceed bytes: 415/ 415 [=====] 100 %

Filename:  LSD.A
Flag:      0 Type: 3 => bytes Start address: 24990 Length: 14970 bytes 3rd param: 8224
Proceed bytes: 14972/ 14972 [=====] 100 %
```

Taptoser při nahrávání dema LSD do ZX

Současná verze umí nahrávat tap soubory do spectra přes libovolný sériový port (ttySx, ttyUSBx, ttyAMAx) s tím, že se pochopitelně další bloky dohrávají, až když je spectrum chce. Např. různá megadema či dohrávací hry. Program zvládá ukládat do tap souboru to, co pomocí SAVE ze spectra pošlete přes sérový port ven. Hlídá kontrolní součet ukládaných bloků. Průběhy nahrávání jsou indikovány progress barem inspirovaným yumem z Fedory. Při nahrávání tapek ze spectra do PC se program sám neukončí, tap soubor tak může být libovolně dlouhý. Ukončení se provede bezpečně pomocí Ctrl+C, což vyšle programu signál SIGINT, který je odchycen programem a ten uzavře nahrávaný soubor a mravně skončí.

```
./taptoser -d /dev/ttyUSB0 -b 57600 -o ROM-232.tap
Serial device: /dev/ttyUSB0, communication speed is: 57600 Bd
Output file is: ROM-232.tap

Filename:  ROM232
Flag:      0 Type: 3 => bytes Start address: 0 Length: 16384 bytes 3rd param: 32768
Data block len: 16384
Proceed bytes: 16388/ 16388 [=====] OK
```

Taptoser při nahrávání tap ze ZX do PC

V průběhu doby se objevila myšlenka na zálohování mých starých (stále funkčních) betadiskových disket do podoby TRD souborů na PC. Způsobů, jak to udělat, je určitě mnoho. Mne zaujal Cygnusův program **TRDOS image copier for SIF** (siftrd) dostupný na adrese: http://cygnus.speccy.cz/popis_siftrd.php

SIF je implementace hardwarového sériového portu pro ZX Spectrum, kde není zapotřebí řešit softwarově tak nízkoúrovňovou věc, jakou je řazení bitů ve správném pořadí, ve správný čas a na správný drát. Protože ale SIF nemám k dispozici, napadla mne možnost využít zdrojového kódu siftrdu a upravit ho pro přenos přes sériák ZX Spectra 128K, jak jinak než s využitím Paulovy ROMky. Rutiny pro sériový přenos v ROM0 ZX Spectra 128K zvládají maximální rychlost pouze 19200Bd což je třikrát pomalejší než Paulova ROM a bůh ví, jak by se daly použít, když jsou v druhé ROM. Paulových 57600Bd je zároveň jen dvakrát pomalejší než SIFových 115200Bd :-)

Ve zdrojovém kódu SIFTRDu stačilo odstranit obsluhu SIFu do zvláštního souboru, který se nepřeložil, a nahradit to pouhým naplněním registrů a voláním ROMky SA-BYTES. V registrech DE je délka dat, v IX adresa bufferu a v A je hodnota flag bajtu 255 na znamení datového bloku a ne hlavičky, která má flag byte nulový. Takto upravený **TRDOS image copier for SERIAL** (sertrd), ale ani původní program siftrd mi na mém ZX Spectru s připojeným betadiskem nefugoval. Řešení se ukázalo jednoduché. Stačí skočit pomocí RANDOMIZE USR 15616 do TR-DOSu udělat CAT diskety a RETURNem se vrát do BASICu. Pak stačilo sertrd (na zkoušku i siftrd) nahrát do ZX (jak jinak než sériově :-)) a stisknout „R“ jako čtení image disku a odeslání přes sériový port do PC. Image disku se provádí po stopách, do PC postupně dorazí 160 (dle počtu stop na disketě) datových bloků po 4096+4 bajtech.

Aby bylo možné takto vzniklý soubor použít jako normální TRD, asi tušíte, že je potřeba v PC odstranit tapkový balast, který produkuje ROMka i u datového bloku. Detailní popis tap souboru (ale i jiných) najdete například na <http://rk.nvg.ntnu.no/sinclair/faq/fileform.html#TAPZ>. Ve stručnosti: Každý datový blok, ale i hlavička, má první dva bajty s délkou dat, pak je flag byte (255) následovaný daty a nakonec kontrolní součet. Ten se počítá jako XOR flag bajtu a všech bajtů dat a musí souhlasit s posledním bajtem v datovém bloku. K tomuto odstraňování přibyla do programu taptoser volba -B nebo delším zápisem --binary, která data přijatá ze spectra okleští o zmíněný balast. Výsledkem je 640KiB TRD soubor v PC.



Sertrd (upravený siftrd) při přenášení obrazu diskety do PC přes sériový port

```
./taptoser -d /dev/ttyUSB0 -b 57600 -o images/GAME001.trd --binary
Running in binary output mode
```

```
Serial device: /dev/ttyUSB0, communication speed is: 57600 Bd
Output file is: GAME001.trd
Data block len: 4096
Data block len: 4096 4096 [#####] OK
Data block len: 4096 4096 [#####] OK
Data block len: 4096 4096 [#####] OK
Data block len: 4096 4096 [#####] OK
Data block len: 4096 4096 [#####] OK
Data block len: 4096 4096 [#####] OK
```

taptoser při nahrávání raw trd image ze ZX do PC

Schopnosti ukládat jen binární data lze využít i pro přenos dat ze spectra do PC. Např. screen\$ o délce 6912 bajtů se ze spectra dostane jednoduše SAVE "obraz" CODE 16384, 6912. Pomocí programu taptoser s volbou --binary se uloží pouze námi požadovaných 6912 bajtů bez hlavičky a dalšího balastu. Volba výstupního souboru s příponou scr (-o soubor.scr) už nám pomůže zobrazit obrázek na PC.

V blízké budoucnosti bude zapotřebí dopsat podporu i pro zápis TRD souborů na fyzickou disketu. Program siftrd umí zapsat obraz diskety. Do patche, který z něj udělá program sertrd, bude potřeba přidat odstranění tapkových bajtů. Program taptoser musí posílat image disku po čtyřkilobajtových částech s doplněnými bajty na bezhlavičkové tap bloky.

Vynechávání a dopňování tapkových bajtů se jeví jako starost navíc. Jsem přesvědčen, že kdyby specy 128 umělo standardně možnost dělat SAVE a LOAD i přes sériový port, byly by sériové přenosy ve specy světě mnohem využívanější (pokud nepočítám síť sestavenou pomocí ZX Interface1, která toto umí).

Na tuto, na první pohled, snadnou zálohu disket bylo použito: ZX Spectrum 128K s upravenou ROM1, řadič Betadisku s připojenými disketovými jednotkami, patřičný zdroj, kabeláž, USB-serial převodník, zx program siftrd upravený na sertrd a konečně linuxový program taptoser s volbou --binary. Rád bych za inspiraci (a zdrojové kódy) poděkoval autorům ROM-RS232, Tapheaderu a Siftrdu. Díky moc, skvěle jsem se bavil.

Pavel Vymetálek