Jak na to

Rady majitelům počítačů
ZX SPECTRUM, DIDAKTIK GAMA
a DELTA,
jak vkládat POKE do her.

autor:
Amatersoft

spolupracoval:
V. Švácha

Jestliže chcete vložit nesmrtelnost do oblíbené hry, doporučuji vám přečíst si následující řádky.

Abychom mohli vložit nekonečné životy do hry, musíme znát vše o nahrávání a popřípadě i spuštění hry (počet bloků, jejich značkové bajty, délku bloků, obsah zaváděcího programu a obsahuje-li strojový kód, tak i ten). Počet bloků, jejich značkové bajty (FLAG) a délku zjistíme z kopírovacích programů, které vypisují i FLAG. Pokud známe tyto údaje musíme zjistit, co obsahuje zaváděcí program (většinou je to BASIC s kombinací strojového kódu). To provedeme tak, že nahrajeme zaváděcí program příkazem MERGE "" (né LOAD ""). Jestli se nám po nahrání objevila zpráva Ø:1,OK, podařilo se nám jej nahrát. Objeví-li se na obrazovce něco jiného, (nebo nic) musíme program nahrát jiným způsobem. Jedním z nich je, že si na jinou kazetu vytvoříme hlavičku pomocí příkazu SAVE "" CODE x,y (x-odkud,y-kolik), která má stejnou délku jako zaváděcí program. Hodnotu x (adresa, od které se budou nahrávat bajty) volte jakoukoliv, doporučuji vyšší adresy (např.5ØØØØ). Potom nahrajte do počítače tuto hlavičku a k ní zaváděcí program bez hlavičky.

Příklad: Víte, že zaváděcí program je dlouhý 64Ø bajtů. Proto si na jinou kazetu nahrajte hlavičku programu: SAVE "nova" CODE 5ØØØØ,64Ø. Pravedte CLEAR 49999 (snížení RAMTOP). Nyní nahrajte hlavičku LOAD "nova" CODE (jakmile se vám nahraje hlavička - objeví se na obrazovce Bytes:nova, vypněte magnetofon) a k ní nahrajte blok dat ze zaváděcího programu (již bez hlavičky).

Nyní máte program nahrán v paměti od adresy x (5ØØØØ). Jednoduchým programem zjistíte obsah.

```
 5 CLEAR 49999
1Ø FOR A=5ØØØØ TO 5ØØØØ+y
2Ø PRINT A;TAB 1Ø;PEEK A;TAB 2Ø;CHR$ PEEK A AND PEEK A>31
3Ø NEXT A
```

Na obrazovce se vám objeví v každém řádku: první číslo - je adresa, na které je uložena druhé číslo (vytištěné na obrazovce) a má-li druhé číslo v tabulce ASCII (najdete v manuálu) přiřazen zobrazitelný znak či výraz, vytiskne se.

Příklad:

```
1Ø BORDER 7:PAPER Ø:INK Ø
```

Tento řádek byl nahrán do počítače způsobem popsaným v předchozím příkladu. Po napsání dekódovacího programu (uveden dříve) a po spuštění příkazem RUN se nám objeví na obrazovce:

```
5ØØØØ    Ø                    -dva bajty čísla řádku (čř)
```

```
50001   10                      čř=(50000)*256+(50001)
50002   27                     -délka řádku (dř)
50003   0                       dř=(50002)+256*(50003)
50004   231     BORDER
50005   55      7
50006   14                     -zapsání čísla pro BASIC
50007   0                      Pokud nesouhlasí čísla  v
50008   0                      příkazu s reprezentací čísel
50009   7                      za kódem 14 (je to možné),
50010   0                      pak překladeč BASICU pracuje
50011   0                      s číslem za kódem 14.
50012   58      :
50013   218     PAPER
50014   48      0
50015   14
50016   0
50017   0                      -hodnotu lze vypočítat:
50018   0                       číslo=(50018)+256*(50019)
50019   0
50020   0
50021   58      :
50022   217     INK 0
50023   48
50024   14
50025   0
50026   0
50027   0
50028   0
50029   0
50030   13                     -konec řádky (ENTER)
```

čísla uvedená v závorkách znamenají obsahy těchto adres (pro 50000 je to číslo 0). Dvoubajtové čísla je vždy uloženo tak, že jeho nižší část leží na nižších adresách než jeho vyšší část (výjimku tvoří číslo řádku). Hodnotu dvoubajtového čísla zjistíme : nižší+256*vyšší (hodnoty né adresy).

V některých hrách je provedena ochrana pomocí barev (INK,PAPER, popřípadě i vložením kódu změny barvy (najdete je v ASCII)). Této ochrany se zbavíme tím, že vyEDITujeme první řádek, na kterém se objevuje změna barev - jestliže nelze vyeditovat (je to nulý řádek) změňte ho na první (POKE 23756,1) a jdeme kurzorem po vyeditovaném řádku (kurzorem vpravo). Jestliže dojde ke změně barvy kurzoru provedeme DELETE, a potom zmáčkneme mezerník (SPACE) pro zachování délky řádku. Tak pokračujeme až do úplného odstranění ochrany (řádek se stane čitelným). Rádek vrátíme zpět pomocí ENTER.

Je-li v zaváděcím programu, který jsme nahrali nebo i odstranili ochranu pomocí barev, napsáno nahrávání všech dalších

částí hry (LOAD "" CODE a podobně) a na konci je příkaz RANDOMIZE USR .. nebo PRINT USR .., který spouští nahranou hru, je vložení nesmrtelnosti jednoduchou záležitostí - vložte POKE před příkaz RANDOMIZE USR nebo PRINT USR a spusťte zaváděcí program (RUN nebo RUN číslo).

Některé programy využívají strojového kódu, který provádí nahrání zbytku hry a zajišťuje také i start hry. Tento strojový kód bývá obvykle na řádku 0 za příkazem REM. Spouštění tohoto kódu je buď přes příkaz RANDOMIZE USR nebo PRINT USR. Zjistíme hodnotu (startovací adresu strojového kódu), která je za těmito příkazy. Může nastat situace, že tuto hodnotu nemůžete najít (za USR je obvykle několik příkazů typu PEEK). Nahraďte proto příkazem PRINT příkaz RANDOMIZE, vymažte příkaz USR a nahraďte ho mezerou (SPACE). Nezapomeňte případně změnit hodnoty v příkazech INK a PAPER (pokud mají stejné hodnoty) a v neposlední řadě i změnit kanál tisku na obrazovku (nesmí se vyskytovat u příkazu PRINT znak #, jinak by jste ho musel vymazat) tak, aby směřoval do horní části obrazovky. Takto změněný program nahrajte na kazetu se stejným automatickým skokem po nahrání (SAVE "x" LINE číslo). Proveďte RESET počítače a zpětně nahrajte upravený zaváděcí program. Startovací adresa by se vám měla objevit na obrazovce. Dávejte pozor i na příkazy POKE do systémových proměnných, které jsou na adresách 23552 až 23732. Tyto POKE mohou způsobit havárii systému. Nahraďte je raději příkazem PRINT, ale jsou-li důležité pro výpočet startovací adresy, musíte s nimi počítat (přepočítat startovací adresu).

Pokud jste zaváděcí program nahráli podobně jako u prvního příkladu, je zjištění startovací adresy takové: změníte pomocí POKE obsah adres, na kterých jsou příkazy RANDOMIZE USR nebo PRINT USR na PRINT a mezeru (kód 32). Všechny závislosti a podmínky jsou stejné jako u programu, který jsme nahráli příkazem MERGE "" (uvedeno v předešlém odstavci). Takto změněný program nahrajte na pásek od adresy uložení (50000) příkazem SAVE "sort" CODE 50000, délka zaváděcího programu. Proveďte RESET počítače, nahrajte hlavičku originálního zaváděcího programu a k němu nahrajte program "sort" bez hlavičky. Startovací adresa strojového kódu umístěného v zaváděcím programu by se vám měla objevit na obrazovce.

Příklad:Máme program, který po výpisu (od adresy 50000) obsahuje

```
0 BORDER 0:PAPER 0:INK 0:RANDOMIZE USR (PEEK 23613+
  +256*PEEK 23614)
```

a za tímto programem následují bajty, které zdánlivě nedávají smysl. Víme také, že tento zaváděcí program je dlouhý 640 bajtů. Také jsme zjistili, že příkaz RANDOMIZE USR .... leží od adresy 50032 (ve skutečnosti leží na adrese 23786, pokud by jsme ho nahráli příkazem LOAD ""). Změníme tedy RANDOMIZE na PRINT a USR na

pomlku (SPACE): POKE 50031,245:POKE 50032,32. Musíme změnit i hodnotu v příkazu PAPER nebo INK, protože když je stejný podklad (PAPER) a stejný inkoust (INK) není vytištěná informace čitelná. Proto změníme například PAPER 0 na PAPER 7: POKE 50014,7. To však nestačí, musíme změnit 5 bajtů za kódem 14 (aby překladač BASICU věděl o této změně: POKE 50018,7. Takto upravený kód nahrajeme na kazetu: SAVE "LOOK" CODE 50000,640. Provedeme RESET počítače, nahrajeme originální hlavičku (zaváděcího programu LOAD "") po hlavičce nahrajeme do počítače upravený kód "LOOK", ale bez hlavičky. Startovací adresa by se měla vytisknout na obrazovku (nezapomeňte na POKE do systémových proměnných a na správný tiskový kanál).

Nyní víme, kde nám začíná strojový kód, ale abychom pochopili jak pracuje nahrávací rutina v ROM (#0556 – 1366), musíme si vysvětlit některé pojmy. (# číslo s tímto znakem je v šestnáctkové soustavě)

Prvním z nich je značkový bajt (někdy značen FLAG). Značkový bajt určuje typ bloku při nahrávání (0 – hlavička, 255 – blok dat). Při nahrávání má nahrávací soubor (hlavička nebo blok dat) tuto formu zápisu na kazetě: značkový bajt – samotná data – paritní bajt. Paritní bajt je kontrolou správně nahraného bloku (provádí XOR samotných dat). Musí se shodovat s paritním bajtem, který je vytvářen při nahrávání programu. Pokud souhlasí blok, nahrál se správně, pokud však nesouhlasí objeví se zpráva R – Tape loading error (dojde k volání programu, na který jsou nastaveny parametry v ERR SP a v paměti, na kterou ukazuje ERR SP). Značkový bajt slouží k tomu, abychom nahráli pouze bloky, které mají stejnou hodnotu značkového bajtu námi požadovaného (bývá určen registrem A – LD A,255 (#FF), může obsahovat jakoukoliv hodnotu od 0 do 255). Například při příkazu LOAD "" program reaguje pouze na hlavičky (registr A má při vstupu do nahrávací rutiny hodnotu 0 a hlavičky mají značkový bajt též 0).

Dále si vysvětlíme zásobník, který je často využíván, ale i zneužíván. Pod pojmem zásobník si představte část paměti (určena registrem SP– STACK POINTER – ukazatel zásobníku), kam se ukládají nebo vybírají dvoubajtová čísla a to buď návratové adresy nebo jenom hodnoty párových registrů (AF,BC,DE,HL,IX,IY). Pro práci s hodnotami registrů se používají příkazy PUSH a POP a pro návratové adresy to jsou příkazy CALL a RET.

Postup jak pracuje mikroprocesor při příkazu CALL :

1) sníží SP o jedničku a uloží na adresu určenou SP (do zásobníku) vyšší část návratové adresy (návratová adresa má hodnotu jako adresa, na které leží další instrukce mikroprocesoru za příkazem CALL)

2)sníží SP zase o jedničku a na adresu určenou SP (do zásobníku) uloží nižší část návratové adresy
3)do registru mikroprocesoru PC se přenese hodnota, která je za příkazem CALL (např. při CALL 30000 se do PC přenese hodnota 30000),čímž se provede skok na tuto adresu (ekvivalentem tohoto kroku je instrukce JP v našem příkladě to je JP 30000)

Postup jak pracuje mikroprocesor při příkazu PUSH:

1)stejně jako u CALL (v bodě 1.), ale místo návratové adresy ukládá hodnotu párového registru.
2)stejně jako u CALL (v bodě 2.), ale místo návratové adresy ukládá hodnotu párového registru.

Postup jak pracuje mikroprocesor při příkazu RET:

1)z adresy, na kterou ukazuje SP se vezme nižší bajt návratové adresy a ukazatel zásobníku SP se zvýší o jedna
2)z adresy, na kterou ukazuje SP se vezme vyšší bajt návratové adresy a SP se zvýší o jedna
3)takto vytvořená návratová adresa se uloží do registru PC ,čímž se provede skok na tuto návratovou adresu (JP).

Postup jak pracuje mikroprocesor při příkazu POP:

1)stejně jako u RET (v bodě 1.), ale místo návratové adresy vytváří dvoubajtové číslo pro párový registr
2)stejně jako u RET (v bodě 2.), ale místo návratové adresy vytváří dvoubajtové číslo pro párový registr

Příklad: Máme jednoduchý program ve strojovém kódu

```
30000          LD   SP,65535
30003          LD   HL,4096
30006          PUSH HL
30007          CALL 30014
30010          POP  HL
30011          JP   32768
30014          RET
```

Pro názornost si nastříhejte z papíru kartičky ve tvaru pexesa.
Instrukce LD SP,65535 nám nastaví zásobník na adresu 65535. Instrukce LD HL,4096 dá do registru hodnotu 4096. Instrukce PUSH HL uloží do zásobníku hodnotu 4096 (určena registrem HL). Vezměte první kartičku, napište na ni hodnotu HL (4096), do rohu připište HL a položte kartičku na stůl (dali jste do zásobníku hodnotu registru HL – jedna kartička představuje dvoubajtové číslo). Instrukce CALL 30014 uloží do zásobníku

návratovou adresu. Je to adresa, na které leží další instrukce mikroprocesoru (POP HL), což je v našem případě hodnota 30010. Vezměte další kartičku, napište na ni návratovou adresu (30010), do rohu připište NA (Návratová Adresa), dejte tuto kartičku na poslední položenou kartu na stole (první kartu). Potom instrukce CALL ještě provede skok na adresu 30014. Na této adrese je instrukce RET. Ta nám vezme ze zásobníku posledně vložené číslo (návratovou adresu, POZOR nemusí vždy brát ze zásobníku návratovou adresu, ale i třeba hodnotu párového registru, kterou považuje za návratovou adresu). Vezměte ze stolu nejvrchnější kartičku. Na ní je napsáno číslo, na které instrukce předá řízení. Vyzvednutou kartičku nevracejte na stůl. Nyní se mikroprocesor vrátil na adresu 30010 (návratovou adresu). Zde se nalézá instrukce POP, která nám vezme ze zásobníku hodnotu pro registr HL. Vezměte ze stolu další lístek a hodnota, která je tam napsána, je dána do registru HL. Na stole by vám neměla zbýt žádná kartička (SP registr ukazuje zase na adresu 65535). Jako další instrukce je zde JP 32768, která například spouští samotnou hru.

Jestli jste pochopili vyzkoušejte následující programy:

Program 1:

```
30000   LD   SP,60000
30003   LD   HL,10
30005   LD   DE,20
30008   PUSH DE
30009   PUSH HL
30010   CALL 30018
30013   POP  DE
30014   POP  HL
30015   JP   32000
```

Program 2:

```
30000   LD   SP,40000
30003   LD   HL,32000
30005   LD   DE,20
30008   PUSH DE
30009   PUSH HL
30010   CALL 30017
30013   POP  HL
30014   POP  DE
30015   PUSH HL
30016   RET
30017   LD   HL,30
30020   RET
```

Program 3:

```
30000   LD   SP,50000
30003   LD   HL,32000
30005   LD   DE,20
```

```
30008    PUSH DE
30009    PUSH HL
30010    JP   30015
30013    XOR  A
30014    ADD  A,B
30015    LD   HL,45
30018    RET
```

Pokud jste správně pracovali, vyšlo vám, že každý z programů skáče na adresu 32000 a to rozdílnými způsoby.

Jestli jste pochopili tyto pojmy, můžete začít dekódovat strojový program v zaváděcím programu a to pomocí disassembleru nebo monitoru. Z takto dekódovaného programu zjistíte údaje o nahrávání další části hry.

Příklad: Máme tento program ve strojovém kódu

```
LD   A,255
LD   IX,25000
LD   DE,35000
SCF
CALL 1366
JP   32000
```

Instrukce LD A,255 určuje hodnotu značkového bajtu (255). Obsah registru IX určuje kam se budou ukládat nahrávané bajty (LD IX,25000). Obsah registru DE určuje kolik bajtů se má nahrát (LD DE,35000). Instrukce SCF nastavuje CY na 1 a to proto, že voláním podprogramu 1366 odlišujeme LOAD (CY=1) od VERIFY (CY=0). Podprogram na adrese 1366 nahraje do počítače program námi zvolenými parametry (kam, kolik, značkový bajt). Pokud dojde při nahrávání k chybě, volá chybové hlášení přes RST 8 - systémová proměnná ERR SP ukazuje (její dvoubajtová hodnota) na místo v paměti, kde je uložena adresa (dvoubajtová) začátku programu chybového hlášení, proto tuto hodnotu (na kterou ukazuje ERR SP) lze změnit. Jinak se vrací instrukcí RET. Tento podprogram v ROM využívá zásobník do hloubky tří dvoubajtových hodnot (PUSH - uloží adresu výstupní rutin, CALL volání k nahrání bajtů, CALL volání k nahrávání bitu). Zde se využívá toho, že program uloží adresu výstupního programu pomocí instrukce PUSH HL (HL=#053F), do kterého program skáče přes instrukci RET.

Někdy bývá strojový kód doplněn jinými instrukcemi (např. LD SP,...) nebo bývají tyto kódy doplněny ochranou, proto tyto strojové programy nedoporučuji dekódovat jedincům, kteří nemají dobrou znalost strojového kódu a systému počítače.

Jestliže víme o strojovém kódu v zaváděcím programu a vše o nahrávání hry (délka, kam se ukládá, značkový bajt, startovací adresu hry) můžeme vložit nesmrtelnost a to buď na stálo nebo ve

formě tréninku (po stisknutí T - po nahrání hry - máme nesmrtelnost, když stiskneme jedno z písmen Q-W-E-R budeme mít normální verzi hry). Pro verzi programu se stálou nesmrtelností vytváříme takovýto program v assembleru:

```
LD   A,255
LD   IX,25000
LD   DE,35000
SCF
CALL 1366
LD   A,0
LD   (35360),A
JP   32000
```

Tento program je podobný jako dříve uváděný program, ale liší se tím, že před startem hry (JP 32000) je vložen POKE ((LD A,Ø ; LD (35360),A). V BASICU tyto instrukce mají ekvivalent a to : POKE 35360,Ø. Musíte dávat tento POKE vždy až před instrukci startu hry (až po nahrání celé hry). Dávejte také pozor, kam tento program umístíte, protože by mohlo dojít k přehrání tohoto programu části hry, a taky musíte dávat pozor na zásobník. Při přehrání zásobníku dojde k zhroucení počítače (hodnota registru SP nesmí zasahovat do oblasti začínající obsahem registru IX a končící součtem IX+DE ani do její blízkosti).

Pro verzi programu s tréninkem upravíte kód takto:

```
      LD   A,255
      LD   IX,25000
      LD   DE,35000
      SCF
      CALL 1366
      DI
KROK  LD   BC,#FBFE
      IN   A,(C)
      AND  %11111
      CP   %11111
      JR   Z,KROK
      EI
      CP   %01111
      JP   NZ,32000
      LD   A,0
      LD   (35360),A
      JP   32000
```

Je tu také možnost, že nahrajeme pouze blok, do kterého chceme vložit nesmrtelnost, na přesné místo v paměti (hodnota registru IX). Do takto nahraného programu vložíme POKE a zpětně přehrajeme původní blok blokem novým s nesmrtelností. Tohoto se využívá, pokud nemůžeme najít startovací adresu hry. Hra má potom stálou nesmrtelnost. Jestli použijete tento způsob,

nemusíte upravovat vůbec žádný strojový kód ani BASIC.

Příklad: Víme, že blok, do kterého chceme dát nesmrtelnost je dlouhý 35000 bajtů a nahrává se od adresy 25000 se značkovým bajtem 255. Nesmrtelnost je POKE 35360,0. Vytvoříme proto strojový kód, který nám zabezpečí nahrání, vložení nesmrtelnosti a zpětný záznam hry na pásku:

```
ORG  16384          -kam se ukládá tento kód
LD   SP,20000
LD   A,255
LD   IX,25000
LD   DE,35000
SCF
CALL 1366
```

Tento program nám umožní správné nahrání bloku.

```
LD   A,0
LD   (35360),A
```

Toto nám vloží nesmrtelnost.

```
     DI
KROK IN   A,(#FE)
     OR   (#E0)
     INC  A
     JR   Z,KROK
     LD   A,255
     LD   IX,25000
     LD   DE,35000
     CALL #04C2          -SAVE
     EI
     RET
```

Tento program nám zabezpečí nahrání upraveného kódu zpět na kazetu. Vložte tento program do assembleru od adresy 16384 a spusťte ho. Nahrajte blok dat, v kterém chcete udělat nesmrtelnost, do počítače. Připravte si jinou kazetu. Program čeká na stisk klávesy. Stiskněte klávesu (nemačkejte SPACE) a upravený blok se vám začne nahrávat. Potom nahrajte do počítače kopírovací program. Do něho nahrajte celou hru, do které jste dělali nesmrtelnost. Místo originálního bloku (25000,35000) nahrajte upravený blok. Potom tuto hru (z kopírovacího programu) nahrajte na kazetu. Nyní máte nesmrtelnou verzi hry.

Někdy se vám stane, že i když jste správně vložil nesmrtelnost tak vám nefunguje. To je způsobeno tím, že některé hry při

nahrávání modifikují bajty (provádí XOR) nebo při startu hry přenesou nahraný blok na jiné adresy, než na jaké se nahrají. Nejčastější je, že program přehraje zásobník a tím změní návratovou adresu (přehrají adresu #ØF53F a návratovou adresu ze zaváděcího programu). Zjistit tyto způsoby může pouze programátor znalý systému počítače a strojového kódu.

Jsou programy, které se nahrávají příkazem LOAD "" CODE a sami se po nahrání spustí. To je způsobeno buď přehráním zásobníku nebo systémových proměnných. Nejdříve si vysvětlíme způsob, při kterém se přehrává zásobník. Poloha zásobníku je určena hodnotou systémové proměnné RAMTOP (je o 1 menší). RAMTOP je největší hodnota adresy, se kterou pracuje BASIC. Dojde-li tedy k přehrání zásobníku, program se již nevrací, ale skáče přímo do nahrané hry. Startovací adresu (kam skáče program po nahrání) zjistíte tak, že CLEAR (pomocí CLEAR nastevujete RAMTOP) snížíte tak, aby se program nahrával za RAMTOP. Program nahrajte zcela normálně: LOAD "" CODE. Jestliže víte hodnotu CLEAR v originálním zaváděcím programu (standartně je nastaven na 65367) snadno zjistíte startovací adresu: PRINT PEEK (CLR-8)*256+PEEK (CLR-9), kde CLR je hodnota RAMTOP (stejná jako za příkazem CLEAR). Na obrazovce se vám objeví startovací adresa vaší hry.

Příklad: U hry EXPRESS je v zaváděcím programu snížen RAMTOP na hodnotu 32767 pomocí CLEAR 32767. Vlastní program se nahrává pomocí LOAD "" CODE a to od adresy 32722 až na adresu 48114 (zjištěno v kopírovacím programu). Jelikož program přehrává svými 45 bajty zásobník, program se sám spouští. Abychom našli startovací adresu změníme CLEAR na hodnotu 32721 (zmenšíme tím přehrání zásobníku) a nahrajeme druhou část hry: LOAD "" CODE. Po nahrání si vypíšeme startovací adresu: PRINT PEEK (32767-8)*256+ +PEEK (32767-9). Na obrazovce se nám objeví startovací adresa hry EXPRES. Vložte POKE a spusťte hru: RANDOMIZE USR startovací adresa.

Hry, které přehrávají systémové proměnné, je většinou změní tak, že po nahrání programu ukazuje systémová proměnná CH ADD adresu příkazu, který se po nahrání programu provede (je to podobné jako když napíšete LOAD "" CODE: RANDOMIZE USR startovací adresa). Tímto příkazem je např. RANDOMIZE. Proto tuto adresu musíme zjistit. Nahrajte program tak, aby nezasahoval do systémových proměnných (nezapomeňte na CLEAR na adresu o 1 menší než se bude nahrávat program). Musíme vědět, kde ve skutečnosti leží první bajt. Od něho spočítáme odchylku k systémové proměnné (23645). Tuto odchylku připočítáme k adrese, na kterou jsme program nahráli. Výsledkem je adresa X (CH ADD), která nyní leží v paměti. Zjistíme hodnotu této dvoubajtové proměnné: PRINT PEEK X+256*PEEK (X+1)+1. Na obrazovce se vám objeví adresa, na které leží příkaz, který se vykoná po nahrání.

Jelikož jste program nahráli na jiné adresy, musíte tuto adresu přepočítat: víte, kde leží první bajt (ve skutečnosti), tuto hodnotu odečtete od adresy na obrazovce, tento rozdíl přičtete k adrese, na kterou jste nahráli program, výsledkem je číslo Y, které je adresou příkazu, který se má vykonat. Jednoduchým programem zjistíte obsah:

```
10 FOR A=Y TO Y+10
20 PRINT CHR$ PEEK A;
30 NEXT A
```

Na obrazovce se vám objeví příkaz ke spuštění hry, z kterého vyčtete startovací adresu programu (RANDOMIZE USR ..., PRINT USR ...).

Příklad: Hra PINBALL se nahrává od adresy 16384 a je dlouhé 16384 bajtů (zjištěno v kopírovacím programu). Proto provedeme např. CLEAR 29999 a potom nahrajeme hru pomocí LOAD "" CODE 30000. Víme, že bajt na adrese 30000 je vlastně na adrese 16384. Zjistíme rozdíl systémové proměnné CH ADD od adresy 16384: 23645-16384=7261. Tento rozdíl přičteme k adrese, na kterou jsme program nahráli: 30000+7261= 37261. Na adrese 32761 tedy začíná proměnná CH ADD. Vypíšeme obsah: PRINT PEEK 37261+256*PEEK 37262. Tato adresa je však při originálním nahrávání, proto získáme i adresu v našem nahraném programu: 23862-16384=7478+30000=37478. Na adrese 37478 leží příkaz pro start programu. Programem zjistíme o jaký příkaz se jedná:

```
10 FOR A=37478 TO 37478+10
20 PRINT CHR$ PEEK A;
30 NEXT A
```

Na obrazovce se objeví: RANDOMIZE USR 27392 což je startovací adresa hry PINBALL.

Vložení nesmrtelnosti do takovýchto her je stejné jako u her, kde známe nahrávání a spuštění.

Některé hry jsou již upravené pro lehčí vkládání pouků. Myslím tím hry, kde se při nahrávání rozbliká nápis "M1 LOADING". Způsob vložení je tento:
- nahrajeme příkazem MERGE "" basic; vypneme magnetofon
- zjistíme startovací adresu: PRINT PEEK 23639+256*PEEK 23640
- napíšeme POKE 23638,201: RUN
- zapneme magnetofon
- po nahrání celého programu se vypíše hlášení O.K.
- vložíme POKE pro nesmrtelnost
- poté rozeběhneme hru příkazem RANDOMIZE USR startovací adresa

Může se stát, že potřebujete nahrát jenom kousek kódu z bloku dat a přitom ostatní kód nechcete nahrát (třeba i z prostředku bloku dat). Předkládám vám následující řešení, které této podmínce zcela vyhovuje.

```
10 RUN 110
20 CLEAR 44999
30 FOR A=45000 TO 45201
40 READ S: POKE A,S
50 NEXT A
60 STOP
70 DATA 17,0,0,221,33,0,0,33,0,0,34,106,176,62,0,50,93,
        176,33,0,0,34,67,176,205,229,175,251,201,20,8,
        21,243,62,15,211,254,219,254,31,230,32,246,2,79,
        191,192,205,115,176,48,250,33,21,4,16,254,43
80 DATA 124,181,32,249,205,111,176,48,235,6,156,205,111,
        176,48,228,62,198,184,48,224,36,32,241,6,201,
        205,115,176,48,213,120,254,212,48,244,205,115,
        176,208,121,238,3,79,38,0,6,176,24,21,8,221,117
90 DATA 0,24,9,203,17,173,121,31,79,19,24,2,0,0,27,8,6,
        178,46,1,205,111,176,208,62,203,184,203,21,6,
        176,210,75,176,122,179,32,215,0,62,201,50,93,176,
        33,221,35,34,67,176,17,0,0,8,24,214,205,115,176
100 DATA 208,62,22,61,32,253,167,4,200,62,127,219,254,31,
        208,169,230,32,40,243,121,47,79,230,7,246,8,211,
        254,55,201
110 INPUT "KAM: ";IX''"KOLIK: ";HL''"OD KTEREHO BAJTU ";DE
120 POKE 45001,DE-256*INT (DE/256): POKE 45002,INT (DE/256)
130 POKE 45005,IX-256*INT (IX/256): POKE 45006,INT (IX/256)
140 POKE 45008,HL-256*INT (HL/256): POKE 45009,INT (HL/256)
150 RANDOMIZE USR 45000
160 IF INKEY$=CHR$ 32 THEN STOP
170 GO TO 150
```

Nejdříve spusťte program příkazem RUN 20. Do paměti se vám uloží od adresy 45000 strojový kód. Potom stačí pouze volat program příkazem RUN. Proměnná IX určuje kam se nahraje vybraná část programu, kolik bajtů se má nahrát určuje proměnná HL, a od kterého bajtu se začnou zaznamenávat do paměti určuje proměnná DE. Názorně to uvidíte při nahrávání obrázku. Chceme-li např. nahrát prostřední část obrazovky, odpovíme: KAM – 18432, KOLIK – 2048, OD KTEREHO BAJTU – 2048. Chceme-li celou obrazovku bez barev, zadejte v pořadí – 16384, 6114, 0. Strojový kód reaguje na všechny typy FLAG a bloky různé délky (podobně jako kopírovací program).

Pokud NEchcete nahrát jenom část bloku, zkuste tento program:

```
10 RUN 120
20 CLEAR 49999
```

```
30 FOR A=50000 TO 50217
40 READ S: POKE A,S
50 NEXT A
60 STOP
70 DATA 17,0,0,221,33,0,0,33,0,0,34,3,196,33,24,16,34,
        231,195,33,221,117,34,192,195,205,110,195,251,
        201,221,43,20,8,21,243,62,15,211,254,219,254,31
80 DATA 230,32,246,2,79,191,192,205,12,196,48,250,33,21,
        4,16,254,43,124,181,32,249,205,8,196,48,235,6,
        156,205,8,196,48,228,62,198,184,48,224,36,32,241
90 DATA 6,201,205,12,196,48,213,120,254,212,48,244,205,
        12,196,208,121,238,3,79,38,0,6,176,24,21,8,
        221,117,0,24,9,203,17,173,121,31,79,19,24,2,221
100 DATA 35,27,8,6,178,46,1,205,8,196,208,62,203,184,203,
        21,8,176,48,243,122,179,32,216,24,16,33,221,117,
        34,192,195,17,255,255,221,35,221,35,8,24,216,33
110 DATA 0,0,34,231,195,34,192,195,17,0,0,8,24,201,205,12,
        196,208,62,22,61,32,253,167,4,200,62,127,219,254,
        31,208,169,230,32,40,243,121,47,79,230,7,246,8,
        211,254,55,201
120 INPUT "KAM: ";IX''"KOLIK: ";HL''"VYNECHAT BAJTU ";DE
130 POKE 50001,DE-256*INT (DE/256): POKE 50002,INT (DE/256)
140 POKE 50005,IX-256*INT (IX/256): POKE 50006,INT (IX/256)
150 POKE 50008,HL-256*INT (HL/256): POKE 50009,INT (HL/256)
160 RANDOMIZE USR 50000
170 IF INKEY$=CHR$ 32 THEN STOP
180 GO TO 160
```

IX - kam se bude ukládat nahrávaný blok
HL - kolik bajtů se nahraje než dojde k vynechání
DE - kolik bajtů se má vynechat

Program spustíte příkazem RUN 20. Kód se vám uloží od adresy 50000. Potom program spouštějte jenom příkazem RUN. Pro názornost si předvedeme nahrávání obrazovky tak, že prostřední část obrázku se nenahraje (na tomto místě zůstává obsah, který tam byl před nahráním - normálně by se přehrál). Odpovíme postupně: 16384, 2048, 2048.

Tyto dva strojové kódy nejsou relokovatelné (přenositelné a spustitelné na jakémkoliv místě paměti). Můžete však provést přenos pomocí monitoru, který změní hodnoty v příkazech CALL a JP tak, že je můžete spustit i v jiné části paměti.