

ZX magazín

Časopis pro uživatele počítačů
Sinclair ZX-Spectrum, Didaktik,
Delta, Sam Coupé

3/93

Z obsahu čísla:

Tomahawk, Grand Prix, Cyclone, Skool Daze,
Battle Command, Ano Gaia, Amaurote — návody ke hrám
File Manager — Nový Public Domain
Universum vzpomíná . . .
Robotika
Animace
Systémové proměnné
Nekonečné životy
Komiks
Intro — počítače a humor

20 Kč



ZX MAGAZÍN - PŘEDPLATNÉ 1993

ZX magazín je největší a nejstarší časopis pro majitele počítačů ZX Spectrum, Didaktik Gama, Didaktik M, Didaktik Kompakt, Sam Coupé a kompatibilní v Čechách a na Slovensku - vychází již od roku 1988. Je jediný svého druhu, který je psán v češtině. Vychází 6 krát za rok. Každé 2 měsíce dostanete jeden výtisk formátu A4 s minimálně 32 stranami za pouhých Kč 18,-- (cena pro předplatitele je zvýhodněná proti ceně v PNS nebo v ostatních prodejnách, kde činí Kč 20,--).

Zaměření časopisu ZX magazín:

- zajímavé hry (především zahraniční) - popisy, manuály, žebříčky nejúspěšnějších her
- převod programů a her z kazety na disketu (seriál článků s podrobným popisem)
- rady a informace, BASIC programky pro začátečníky (které stačí pouze přepsat do počítače a začít s nimi experimentovat)
- uživatelské programy, hardware, schémata zapojení, informace o novinkách, recenze
- grafika a její animace na Spectru, programátorské finty, robotika
- tiskárny, jejich zapojení a software k nim
- disketové jednotky D40 / 80 / Kompakt, jejich ovládání a programování obslužných programů k nim, a to v Basicu nebo v Assembleru
- PUBLIC DOMAIN - programy (téměř zadarmo)
- kupóny opravňující ke slevě při nákupu programů od PROXIMY na dobírku (v každém čísle najdete objednávací kupón, s nárokem na slevu 20 Kč)
- inzerce firem i občanů
- počítače a humor (veselé příhody z oblasti výpočetní techniky, obrázky, anekdoty)

Do ZX magazínu přispívají nejlepší odborníci na počítače Spectrum / Didaktik!

Časopis je možno předplatit si na celý rok. Mimoto se bude prodávat v některých prodejnách, kde prodávají příslušenství k počítačům Didaktik a Spectrum, a také v PNS. Předplatné na celý rok činí 108,-- Kč. Objednání předplatného je velmi jednoduché: částku nám poukažte **pětidílnou** složenkou typu A (pokud ji neseženete, můžete si ji vyžádat u naší firmy). Jeden ze dvou dílů složenky, které Vám na poště po zaplacení dají, nalepte na objednávací kupón na této straně dole (můžete si ji zkopírovat, abyste si ZXM nazničili) a tento kupón pošlete na adresu: **PROXIMA, box 24, 400 21 Ústí nad Labem**. Druhý díl složenky si uschovejte pro kontrolu nebo pro případ ztráty objednávky.

Pokud by číslo, od kterého si předplatné objednááte, bylo již rozebráno, zašleme Vám ZX magazín až od dalšího čísla a zaplacená částka Vám zůstane u nás na předplatitelském kontě (přebývající část peněz převedeme do dalšího ročníku).

Jako předplatitelé dostanete časopis ZX magazín o několik dnů dříve, než se objeví v prodeji a také ušetříte 2 Kč na každém čísle!

Místo pro nalepení
kontrolního
ústrižku složenky

Objednávám předplatné ZX magazínu,
ročník 1993, od čísla _____

Adresa odesílatele:

PSČ:

Adresa příjemce:

PROXIMA, box 24, 400 21 Ústí n/Lab.

Vážení čtenáři,

úvodník je obvykle místo, kde se dozvíte všechno možné, jenom ne nic podstatného, světlou výjimku ovšem tvoří (jak také jinak, že) ZX Magazin (zvláště, když to vezmu do ruky já):

První, o čem bych se rád zmínil, je obsahové zaměření časopisu - i dále chceme zachovávat současný poměr mezi popisy a recenzemi her a ostatními články - tedy asi tak 2 : 3. Nemáme v úmyslu rozšiřovat zaměření ZX Magazínu i na další počítače, nemusíte se bát, že se tu setkáte s počítači jiných značek (žádné Atari, žádný Commodore,) - nemyslíme si totiž, že tato cesta vede k získání dalších čtenářů, naopak, mohli bychom o Vás spíše přijít. V tomto pravidle ale brzy uděláme malou výjimku - počítače řady PC. Většina z Vás se s těmito počítači už jistě setkala a setkává, další to pravděpodobně brzy čeká. Někud se tedy na nás snad zlobit, když věnujeme několik (3-5) stran těmto počítačům - budeme se ale vždy snažit o nějakou souvislost se Spectrem/Didaktikem.

Pokud jste odebírali ZX Magazin i v minulém roce, víte, že jsme začínali s formátem A5, 24-ti stranami, poměrně osklivou obálkou (návrh vypadal poměrně dobře) a texty tištěnými částečně pomocí Desktopu. Postupně jsme zvětšili počet stránek (z 28 na 32 44 až na 48), od čísla čtyři máme barevnou obálku, texty se postupně stále více připravovaly pomocí programu Ami Pro a časem jsme začali používat pro tisk laserovou tiskárnu - zvýšila se tak jak čitelnost, tak grafická úprava časopisu. S novým ročníkem jsme změnil i formát (tisk vyjde levněji, takže nemusíme příliš zvyšovat cenu). Velké změny už tedy ZX Magazin nečekají, ale drobná vylepšení bychom chtěli provádět stále - v tomto čísle najdete novou hlavičku u Listárny a komiks převzatý z časopisu Your Sinclair.

Tímto ročníkem jsme začali pro vedení databáze předplatitelů ZXM používat nový program, který nám pro každého předplatitele vede informace o adrese, zaplacených částkách, odeslaných číslech a jejich ceně a zůstatku na kontě. Program nám tiskne na samolepící štítky adresu, číslo předplatitele a zůstatek na kontě (ten Vám říká, kolik čísel ještě dostanete, stačí když si ho vydělíte cenou jednoho čísla). Pokud máte na kontě méně, než je cena jednoho čísla ZXM, tak další číslo nedostanete, peníze Vám ovšem na kontě zůstávají. Několik z Vás nám poslalo jako

předplatné na celý rok částku nižší (loňskou), pokud budete chtít i další čísla - zjistíte si kolik Vám chybí a zbývající částku pošlete složenkou (stejně jako u předplatného). Občas si také objednáváte ZXM od čísel, která ještě delší dobu nevyjdou (vzhledem k velkému počátečnímu skluzu ve vydávání je to pochopitelné). My u těchto objednávek předpokládáme, že chcete první číslo, které můžete dostat a tak objednávku upravujeme na to číslo, které má právě vyjít (zatím si nikdo nestěžoval, tak doufáme, že Vám to tak vyhovuje), i zde pak chybí peníze na poslední čísla ročníku.

Rádi bychom už v příštím čísle uveřejnili podmínky předplatného na příští rok, to abyste mohli případné doplátky na tento ročník spojit hned s předplatným na ročník příští (čímž se ušetří práce nám i Vám). Pokud budete posílat doplatek nebo nové předplatné, pište prosím vždy Vaše předplatitelské číslo!

K předplatnému ještě drobnou poznámku - občas se stane, že se některá složenka na poště zatoulá a my ji nedostaneme, totéž se může stát se zasláním číslem a úplně vyloučit nemůžeme ani naši chybu, pak nám samozřejmě napište a raději pošlete kopii ústřížku a zopakujte objednávku (kdyby třeba

nedošla složenka, nemusíme Vám psát, abyste znovu psali částku a od jakého čísla).

Pak bych Vás chtěl požádat o příspěvky - čím více příspěvků, tím vyšší úroveň časopisu. Pište také, co byste chtěli, abychom ZXM dělali pro Vás a podle Vašich představ. Nemohl by někdo z Vás napsat popis (recenzi) programu (třeba Artist II nebo napsat nějaký článek, který by srovnal na ZX a kompatibilních existující textové a grafické editory, kompilátory basicu, tabulkové procesory, databáze a další programy, popsal by jejich výhody i nevýhody, který by se prostě zabýval v souhrnu některým typem programu a usnadnil by tak hlavně začátečnickům orientaci?

A nakonec bych se zmínil o jedné záležitosti, která Vám zřejmě (dle mnoha dopisů) nedá spát - „Proč se INTRO vyskytuje na konci časopisu a ne na začátku, jak by se dalo očekávat“. Podle mnoha seriózních průzkumů se většina novin i časopisů čte a prohlíží od konce a tak je INTRO sice vzadu, ale vlastně přijde první na řadu.

Váš Univerbzum

OBSAH ČÍSLA 3/93

Vážení čtenáři	1
Tomahawk - popis hry	2
Cyclone - popis hry	4
Grand Prix 128 - popis hry	5
Battle Command 128 - popis hry	6
Ano Gaia 48/128 - popis hry	7
Skool Daze - popis hry	8
Amaurote 48 - popis hry	9
Jak začínali ... Universum	10
File Manager - nový shareware od Proximy	12
Tolstoj - reklama	13
Zvukové generátory Best a Nicol	14
Universum's Top 10	14
Rozšířená grafika na ZX Spectru	15
Samcon '93 - Sam není mrtev	15
Listárna ZX magazínu	16
Bombardér - Basic program	17
ZX magazin a robotika (6)	18
Videodigitizer pro ZX Spectrum / Didaktik - reklama	19
Animace 2 (3)	20
Systémové proměnné	22
SAM - sériové rozhraní	24
Strojový kód Z-80 (7)	26
Nekonečné trápení s nekonečnými životy (1)	28
Ernie	31
Intro	32

Tomahawk

Máte rádi simulátory? Já moc. Svého času jsem si dost dlouhou vystačil se simulátorem vrtulníku TOMAHAWK. Protože se jedná o velmi zdařilý program, myslím si, že by stálo za to si jej připomenout, zvláště proto, že jeho ovládání není z nejjednodušších a mohlo by mnoho zájemců odradit ještě dřív, než by na něj přišli - a byla by to škoda.

vysoko se bude oblačná vrstva vyskytovat.

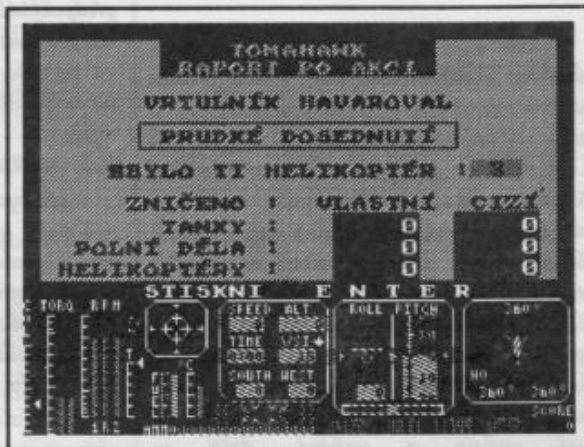
Vlivy větru a turbulence - pokud si zvolíte tuto možnost, ztíží se poněkud ovládání vrtulníku, bude však realističtější.

Zvuk - tady není co dodat.

Kvalita pilota - určujete "nebezpečnost" nepřátel.

Ovládání - ovlivňujete způsob řízení směru a střelení (INTERFACE 2 - klávesy 6, 7, 8, 9 a 0, KEMPSTON, Klávesy - 5, 6, 7, 8 a 0, nebo šipky)

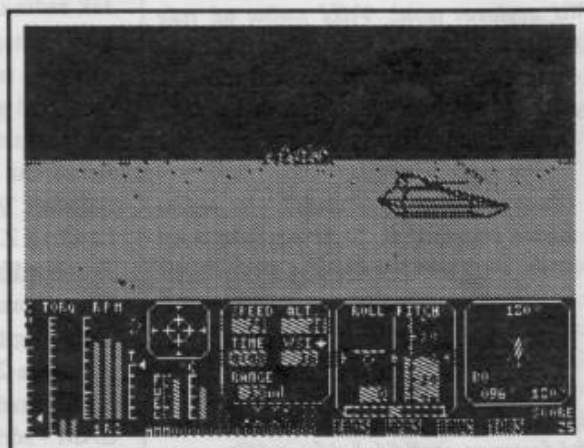
Do hry se dostanete stisknutím klávesy ENTER.



Nad ukazateli střeliva je asi nejdůležitější okénko - obsahuje informace o rychlosti letu v mílích za hodinu (SPEED), pokud je okénko bleděmodré, jde o rychlost zpětnou (couvání), vedle je ukazatel výšky ve stopách (ALT.), pod tím jsou hodiny (TIME - měří čas od začátku hry), ukazatel rychlosti stoupání nebo klesání ve stopách za sekundu (VSI se šipkou), pod tím ukazatel vzdálenosti vybraného objektu (RANGE), pokud se jedná o základnu a je blízko, je zde uvedena vzdálenost a zeměpisné směry.

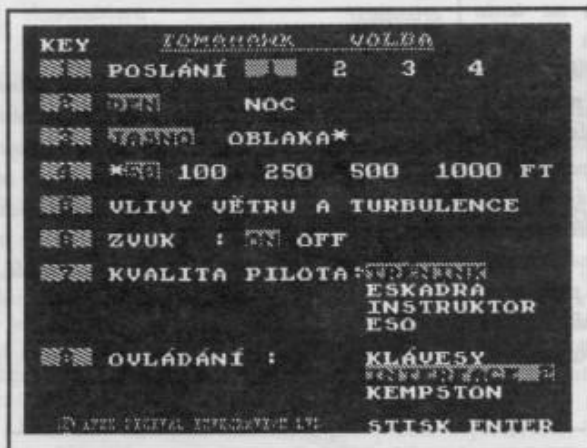
Další velké okénko obsahuje stranový (ROLL) a dopředný (PITCH) náklon vrtulníku. Ve spodní části okénka je ukazatel, o jehož smyslu nemám jasno - snad se jedná o vyvážení vrtulníku.

Pod tímto okénkem jsou zeleně vypsané hlavní systémy vrtulníku - motory (ENGS), zbraně (WPNS), navigační přístroje (NAVC) a vrtule (TADS). Pokud dojde k poškození některého z nich, poznáte to



podle změny barvy a blikání.

Úplně vpravo je okénko s různými čísly a obrázkem vrtulníku. Jsou tu: směr letu (azimut, úplně nahoře), pod ním je obrázek



Začneme od hlavního menu, je v něm pár věcí, které nejsou jasné na první pohled (budu popisovat českou verzi programu, kterou jsem kdysi vytvořil):

Hlavní menu

Poslání: 1 2 3 4 - volíte si okolnosti, za jakých mise probíhá. Jednička je nejspíš (manuál jsem neviděl, a tak to nemohu říci na 100%) mise na cvičném polygonu.

Dvojka se odehrává na hranicích dvou států, kdy došlo k napadení jednoho státu druhým - máte za úkol zlikvidovat útočící jednotky.

Trojka se odehrává na území, které je skoro celé obsazené nepřitelem - půjde asi o vzpouru, které se účastníte.

Čtyřka nás zavádí doprostřed války, územím prochází fronta a mění se podle momentálního stavu - musíte podporovat pozemní vojska.

Den - Noc - volíte si, jestli chcete létat ve dne, nebo v noci.

Jasno - Oblaka - určujete, jestli je jasno, nebo zataženo. Pokud bude zataženo, bude se od určité výšky rozprostírat oblačná vrstva a pokud tuto výšku překročíte, nevidíte nic.

Výška oblačné vrstvy - v případě, že si zvolíte "zataženo", můžete si zvolit, jak

Přístrojová deska

Nejprve si popíšeme přístrojovou desku vrtulníku, vezmeme to zleva doprava:

Nejprve je tu ukazatel nastavení kolektoru (sklon lopatek), čím je sloupec vyšší, tím větší záběr nosná vrtule má (tím větší je nadnášecí síla). Nad tímto ukazatelem je nápis COL.

Další v pořadí je ukazatel namáhání nosného rotoru - pokud vám sloupec překročí kritickou hranici (červená barva), ozve se varovný signál - snižte rychlost nebo změňte nastavení kolektoru. Na ukazatelem je nápis TORQ.

Následuje třísloupcový ukazatel otáček jednotlivých motorů - pravý, levý a rotor (nápis RPM).

Dále jsou ukazatele stavu paliva (FUEL) a teploty motoru (C). Nad nimi je zaměřovací obrazovka - zde občas vidíte typ cíle (kanón, tank nebo helikoptéra) a jeho příslušnost - zelení jsou "naši", červení nepřátelé.

Pak jsou tu ukazatele zbraní - dole (jakýsi plot) se signalizuje stav střeliva pro kulomet, šestiúhelníky z teček ukazují počet neřízených raket, křížky znamenají řízené střely.

vrtníku, poškození je na něm vyznačeno změnou barvy. Vlevo dole je písmeno a

CS, Z - ovládání zadní vrtule - při malé rychlosti letu se můžete s její pomocí otáčet na místě.

C - přepínání režimu naváděcího zařízení, navolený cíl je vypsán v pravém rámečku, v jeho levém dolním rohu (písmeno, číslo), podle nastavení se také na obrazovce objeví zaměřovač nebo ne.

N - přepínání mezi cíli (viz předchozí bod).

M - přepnutí na mapu a zpět. Na mapě vidíte svoji polohu, polohu nepřátelské helikoptéry, polohu osmi

rozložení sektorů a jejich obsazení nepřitelem. (červená), našimi vojsky (modrá) nebo momentálně s bojem (blikání).

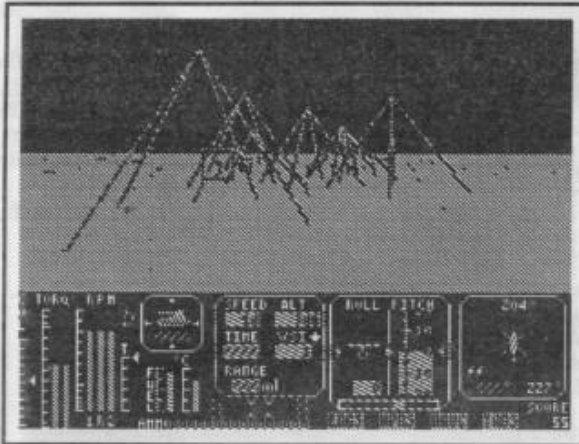
H - pauza.

P - volba zbraní - musíte mít navolen jako cíl T (target) nebo nepřátelskou helikoptéru (dva blesky). Uprostřed výhledu se pak objevuje jeden ze tří typů zaměřovačů. Šikmý kříž je zaměřovač pro kulomet, kříž je zaměřovač pro neřízené střely a čtvereček je zaměřovač pro řízené střely - čtvereček může být buď nakreslen plnou čarou (máte zaměřen nějaký cíl, je vidět v zaměřovači na přístrojové desce) nebo přerušovanou čarou (žádný cíl zaměřen nemáte).

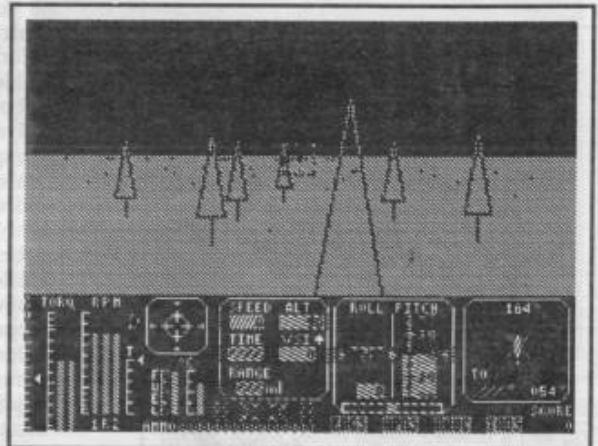
Když budete chtít změnit rychlost stoupání, nebo naopak začít klesat, měňte nastavení plynu nebo kolektoru.

Budete-li chtít zastavit nebo snížit rychlost, musíte vrtulník "zvednout na zadní", budete sice nejprve podobněji stoupat, ale snížíte rychlost. Také můžete střídat měnit směr letu - při velké rychlosti nakláněním, při menší (asi do 30 mph) pomocí zadního rotoru (CS, Z).

Při přistávání se snažte nejprve zastavit nad správným místem a pak pomalu



číslo, je to typ a číslo zaměřeného cíle (T-target, cíl; B-base možná také radiomaják, základna; H-heliport, přistávací plocha, dva blesky - nepřátelská helikoptéra). Na spodním okraji okénka jsou dvě čísla, první je azimut nastaveného cíle, druhé je opět směr letu (alespoň jsem nezjistil, že by se číslo nahore a číslo vpravo dole od sebe nějak lišilo. V tomto okénku také kolem obrázku vrtulníku bliká křížek - udává polohu zaměřeného cíle vzhledem k vrtulníku - pomůže vám určit, jestli se máte otáčet doleva nebo doprava.



klesejte. Všimněte si, že těsně nad zemí je "nosnost" vrtulníku vyšší - zřejmě působení odraženého vzduchu, musíte tedy více ubrat plyn.

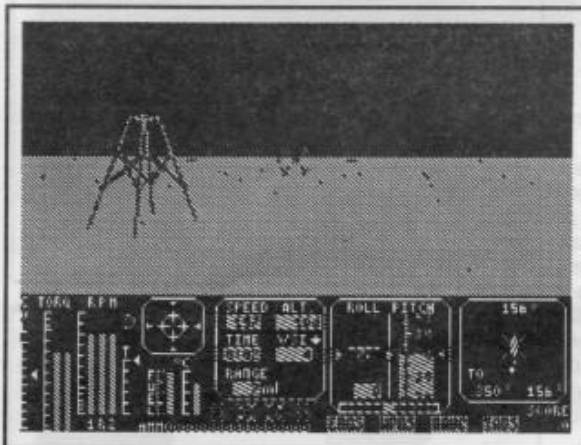
Závěr

Tomahawk je první simulátor vrtulníku, který byl pro ZX Spectrum napsán. Přestože je starý již 8 let, neexistuje program, který by ho výrazněji převyšoval (snad jedině program GUNSHIP). Simulace je velmi dokonalá, místy dokonce lepší než na podobných programech i na počítačích o třídu lepších.

Pokud už program neznáte, nenechte si jej ujít, stojí za vyzkoušení.

Universum

inaktivO



Občas tu také blikají dva blesky - to je signál toho, že nepřátelský vrtulník je příliš blízko.

V pravém dolním rohu přístrojového panelu najdete ukazatel skóre.

Ovládání vrtulníku:

Helikoptéra se ovládá několika klávesami, jsou to:

směry (podle volby ovládání) - vlevo a vpravo mění stranový náklon stroje, nahoru a dolů ovlivňují náklon podélný.

střílení (0 nebo fire na joysticku) - střelba nastaveným typem zbraně.

Q, A - nastavení kolektoru (lopatky vrtule), ovlivňují sílu nadnášení.

W, S - nastavení plynu.

Jak létat

Zkusím se stručně zmínit o tom, jak dostat vrtulník do vzduchu a nějakou dobu tam s ním vydržet.

Nejprve nastavte plný plyn (klávesa W), trojúhelníček se nastaví na maximum, počkejte, až se motory dostanou do otáček. Nyní nastavte kolektor tak asi do tří čtvrtin (klávesa Q), výsledek je po chvíli vidět na namáhání vrtule. První výsledek se již projevil, vrtulník se pomalu odlepuje od země a stoupá.

Když budete v dostatečné výšce, nakloňte stroj mírně dopředu (tah nosné vrtule se trošku přemístí i dopředu a bude vrtulníku udělovat dopřednou rychlost - SPEED).

Když budete chtít zatočit, nakloňte vrtulník trošku na příslušnou stranu a zatočení se dostaví samo.



CYCLONE

(Vortex Software 1985)

Další hra, kterou "vytáhneme" z šeré historie našeho 8-mi bitového miláčka, bude hra CYCLONE. Táhne jí sice už téměř na první křížek, ale autor by si s ní rozhodně ostudu neudělal ani dnes.

Přístrojový panel

Výškoměr (A), rychloměr (S), palivoměr (F) a ukazatel zbývajících času (T). Pod tím

je kompas, který ukazuje směr letu, pod ním pak informace o tom, jaký pohled máte zvolen (severní nebo jižní - je zvolen klávesou N), vlevo od kompasu jsou tři vrtulníčky (vaše životy). Dále můžete vidět nápis **Beware aircraft**, který bliká vždy, když hrozí nebezpečí srážky s letadlem. Následuje ukazatel skóre. Úplně dole je pět přepravních beden - jak je nalézáte a sbíráte, tmavnou, vpravo je pak asi nejdůležitější přístroj - ukazatel síly větru - pokud

jeho sloupec vystoupí až k nápisu **DANGER**, který se rozbliká, jste ve velkém nebezpečí a vrtulník vás přestává poslouchat.

Jak hrát - tipy a triky

Na začátku se objeví obrázek ostrova a na něm je helikoptéra na přistávací ploše, můžete hned startovat, ale počkejte, až se doplní palivová nádrž (jinak moc daleko nedoletíte). Zatím si prohlédněte mapu a zjistěte, kde se nalézá cyklon - tam samozřejmě nelétejte. Pokud cyklon zrovna nefáčí nad základnou a máte plnou palivovou nádrž, startujte.

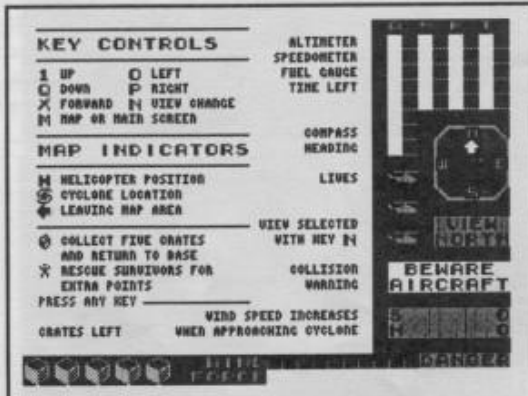
Když dorazíte k nějakému ostrovu, důkladně jej prozkoumejte a to z obou pohledů - přepravky totiž mohou být z jednoho pohledu neviditelné. Pokud nějakou najdete naleťte nad ni tak, aby byla přímo uprostřed vašeho stínu - objeví se lano - a opatrně klesejte, jakmile se lano dotkne přepravky, začne se zvedat. Stejným způsobem můžete nakládat i ty roztomilé lidičky, co na vás zdola mávají (jsou za to body navíc, ale zdržuje to).

Když budete chtít doplnit palivo, nemusíte se vracet na základnu - přistávací plošiny se vyskytují i na jiných ostrovech (jasně bílé čtverce). Přistávejte tak, že budete klesat těsně nad plochu (nebude vidět stín) a potom se velmi opatrně spouštějte níž, jakmile vrtulník dosedne,

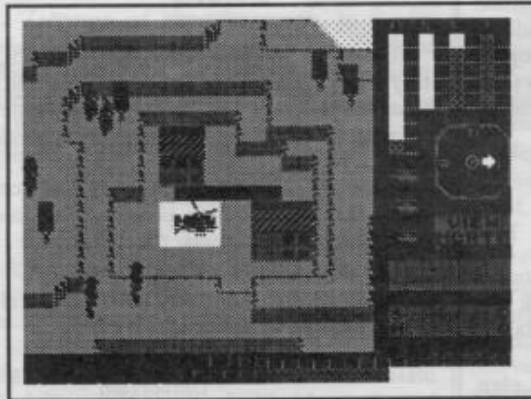
zpomalí se vrtule a začne se doplňovat palivo. Přistát můžete i jinde (když se bude blížit cyklon), ale není to moc dobré řešení, protože palivo ubývá i když "sedíte" na zemi.

Když se k vám přiblíží cyklon a síla větru dosáhne do nebezpečné velikosti, snažte se před cyklonem uletět (podívejte se na mapu, kde je, a leťte opačným směrem). Vrtulník vám přitom bude různě zlobit, ale při troše štěstí se to dá vyřešit.

Když se objeví signál **Beware aircraft**, je nejlepší okamžitě změnit výšku letu.



Jste pilotem vrtulníku a vaším úkolem je je hledat pět přepravních beden, které jsou rozmístěny (poschovávány) na ostrovech. Aby to nebylo tak jednoduché, odehrává se celá hra někde, kde zuří cyklon, tomu se musíte vyhýbat, jinak můžete přijít o helikoptéru.



Ovládání

Když po nahrání hry stisknete klávesu **I**, u které je nápis **Information**, objeví se obrázek s informacemi a na něm se dozvíte (anglicky) asi toto:

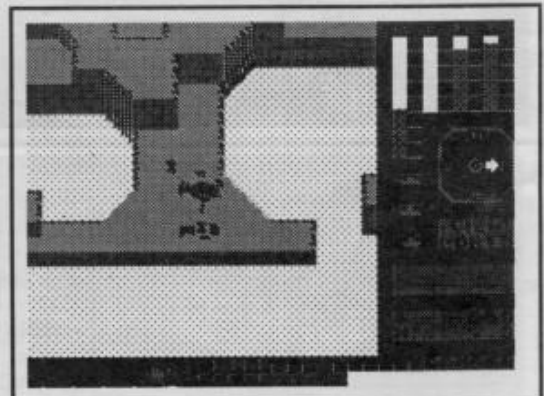
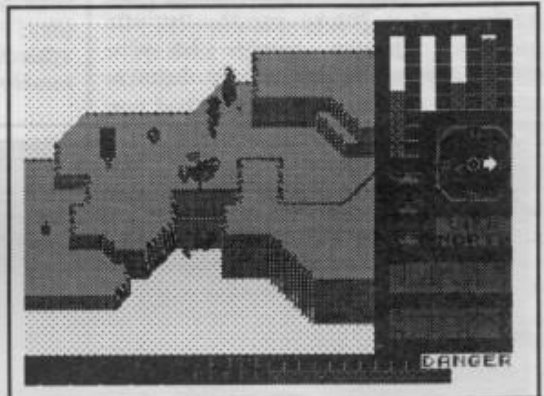
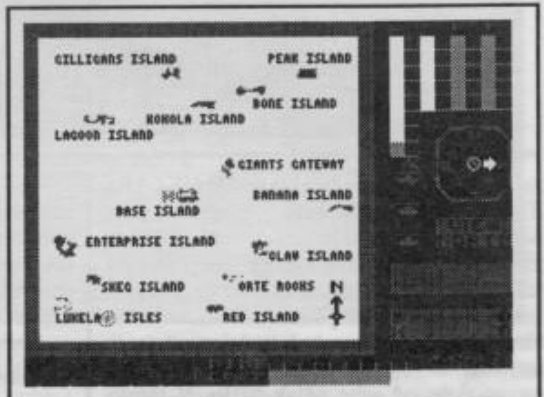
Ovládání klávesami:

I - nahoru, **Q** - dolů, **X** - dopředu, **M** - mapa nebo zpět, **O** - doleva, **P** - doprava, **N** - změna pohledu.

Indikátory na mapě:

H - pozice helikoptéry, obrázek cyklonu - poloha cyklonu, šipka - opouštíte oblast mapy.

Obrázek přepravní bedny a nápis - „shromážděte pět přepravek a vraťte se na základnu“, pod tím obrázek človíčka a nápis - „zachraňte trosečníky za body navíc“.



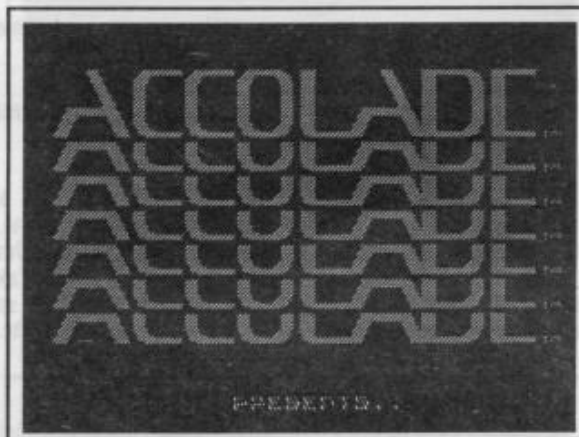
Závěr

Hra je jedna z mála, které používají trojrozměrné zobrazení a současně také všechny barvy. Po zvukové stránce také ujde (snad jenom hudba, kdyby nějaká byla) a jako nápad je vynikající - myslím, že vás rozhodně nezklame.

- Universum -

GRAND PRIX CIRCUIT 128

Accolade 1990



Grand Prix Circuit je další hrou od firmy Accolade s motoristickou tematikou. Jistě znáte: The Duel (Test Drive 2) a 750° cc.

hezký titulní obrázek.

Jako v Test Drive 2 si zvolíte ovládání, obtížnost, počet kol, dokonce se můžete podepsat a vybrat si ze tří typů jízdy.

Practise - trénink

Single race - jeden závod

Championship - šampionát na světových okruzích

Na výběr máte k dispozici několik vozů světových značek (Mc Larren, Ferrari a Williams). Každé má své výhody a

nevýhody.

Jedna "ef jednička" má automatickou převodovku (nemusíte řadit), ale nedosáhne rychlosti jako druhá F1 s ruční převodovkou - vše záleží na Vás, jak se zařídíte.

Spustili jste hru, vybrali jste si auto a ocitáte se na startovní čáře. Stiskem "dopředu" přidáváte plyn, FIRE slouží k přeřazení rychlostního stupně. "Dozadu" slouží k brzdění apod.

K samotnému průběhu hry snad není co dodat. Jen několik poznámek. Hra je celkem rychlá, grafika není špatná (ale byly tu

už lepší), zvuk je otřesný (viz. Test Drive). Protože nejsem jezdec F1, nevím, proč se v



jednou závodě kvalifikujete dvakrát nebo třikrát, jeden okruh jedete dvakrát a pak opět nějaký trénink do dalšího kola. Takže než projedete všechny okruhy, máte asi 120



Grand Prix Circuit byl mimo jiné naprogramován i na Amize a řada věcí byla použita na Spectru 128 (viz. dále). I když je hezké, že hra je pouze pro 128, nebylo to

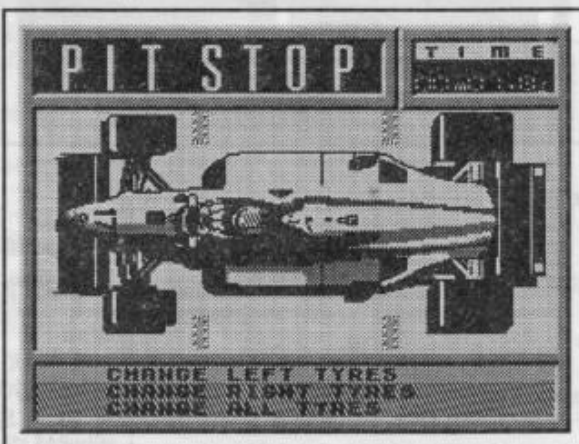


minut za sebou. Co Vás čeká na konci, nevím. Ale asi nic zvláštního - Game over. You are very good player.

- JSH -



stoprocentně nutné. Po jistých úpravách by bylo určitě možné hru vidět i na 48k (možná, že tato verze existuje), ale byla by o mnoho věcí chudší. (Takže ať mě 48-čkaři nelynují



BATTLE COMMAND 128

Ocean 1991

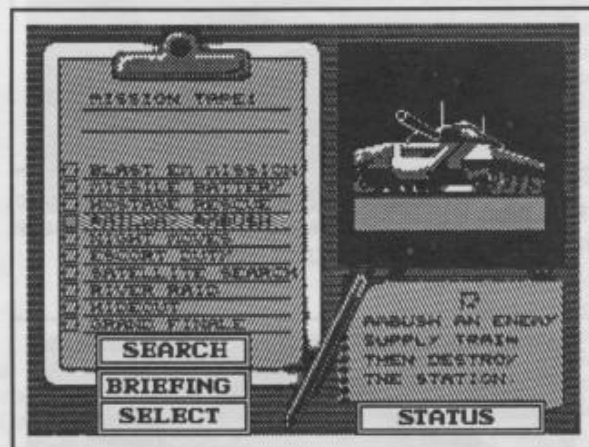
Pokračováním velmi úspěšné hry Carrier Command je tentokrát simulátor bojového tanku.



Jako obvykle, hru musíte nahrát, aby se po nahrání objevilo úvodní menu. (Ten rotující tank se pochopitelně počítá v reálném čase.)

**START GAME,
OPTIONS,
LOAD GAME**

Chcete-li si převolit ovládání - prosím. Po nahrání se šipka ovládá klasicky QAOP. Znáte-li kódy k již uhraným levelům, zadejte LOAD GAME a vložte je.



Po startu hry máte možnost vybrat si z deseti misí.

- 1 - *Blast em mission* - Zničte dvě palivové nádrže nepřítele na jihu
- 2 - *Missile battery* - najděte rakety a zničte je
- 3 - *Hostage rescue* - zachraňte jednoho ze zajatců Severu

4 - *Railway ambush* - přepadněte vlak a zničte stanici

5 - *Night moves* - zaútočte na nepřátelské letiště pod rouškou tmy

6 - *Escort duty* - cestujte s konvojem a zajistěte plnou bezpečnost

7 - *Satelite search* - najděte a znovu zajistěte ztracený satelit na cizím území

8 - *River raid* - zničte loď a nepřátelskou energetickou stanici

9 - *Hideout* - zdemolujte tajné skryté síly protivníka

10 - *Grand finale* - velké finále.

Vlevo dole je

menu:

SEARCH - slouží k nastavení aktuálního ukazatele na pásce (či na disku).

BRIEFING - podrobný text k žlutě označené misi.

SELECT - spuštění mise.

Vpravo dole je nápis STATUS. Když ho "odpálkujete", získáte informace o počtu uhraných misí, kódech atd.

Takže jste si zvolili misi a dostáváte se k vybavování tanku zbraněmi. Šipky slouží k posunování ukazatele.

1 - *Turret* - střelecká věž

2 - *Banshee RRSS* - raketa naváděná radiovými vlnami (typ 1)

3 - *Banshee IRSS* - raketa naváděná infračerveným zářením (typ 1)

4 - *Phoenix RRSA* - raketa naváděná radiovými vlnami (typ 2)

5 - *Phoenix IRSA* - raketa naváděná infračerveným zářením (typ 2)

6 - *K 40 Mortar* - minomet

7 - *Sleeper mine* - mina

8 - *Dragon fly* - ručně řízená střela

9 - *K 90 Cluster* - roj padáčkových střel

10 - *Spectre* - padáčková výbušná střela

11 - *Phantasm* - oklamání raket nepřítele (alobal)

12 - *Skeet bomb* - nenaváděná raketa

13 - *Laser* - laser, který automaticky střílí na nepřítele

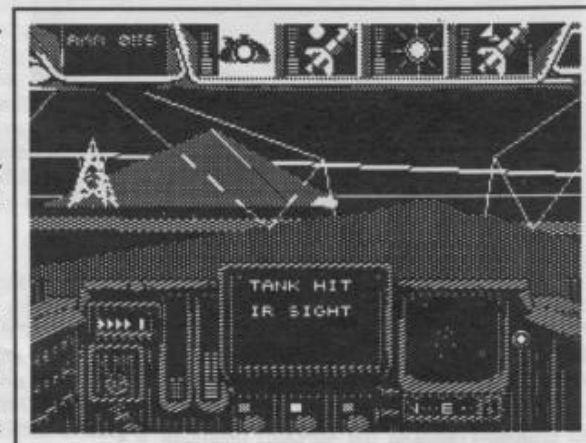
Když se na to tak podíváte, máte k dispozici velmi těžké a nebezpečné zbraně, ale jejich použití je omezená volným prostorem v tanku a jejich vahou. Tak například *Dragon fly* - po stisku FIRE se dostáváte do pozice rakety a ručně ji řídíte.



Vůči jiným střelám je toto skoro universální a nejlepší zbraň. Ovšem do výbavy dostanete pouze dvě rakety tohoto typu, které váží dohromady 500 kg a nosnost tanku je 1000 kg. Takže tady si už musíte poradit sami jak co použít.

No a jestli jste si už vybavili tank, tak stiskněte START GAME a ocitáte se se svým tankem na bojišti a zde máte funkční klávesy:

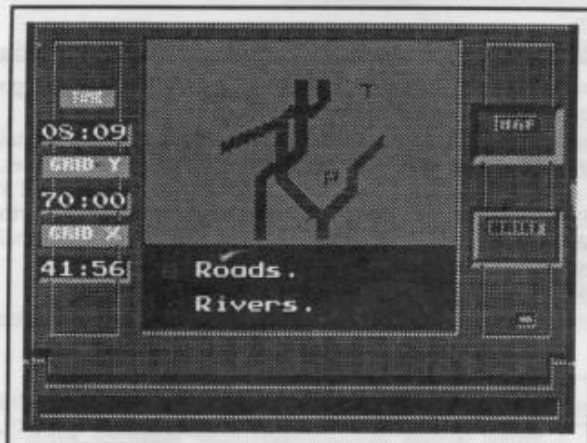
- 1 - 4 výběr zbraní
- E - ukončení hry (End)
- R - pohled do zadu (Rear)



K,J - Auto, Manual, použití pouze u některých zbraní

N - infračervená kamera (používá se v noci)

PICKUP je označen písmenem P, kam musíte po uhrání mise dojet a stisknout H k přivolání vrtulníku.



M - mapa hry (po stisku B se objeví text s informacemi o misi)

B - Binox 2, dalekohled

D - informace o poškození tanku (Damage)

S - rychlost tanku se automaticky sníží (Stop)

H - přivolání vrtulníku (Heli)

C - návrat z Mapy, Damage ...

Tak teď už víte jak tank ovládat a vrhněte se do boje. Směr dopředu slouží k přidávání rychlosti a směr k sobě zase naopak k jejímu ubírání. Když stisknete "M" objeví se mapa. Vaším cílem je dojet na pozici označenou T (TARGET) a udělat to, co Vám káže text, když v mapě stisknete B. Splníte-li misi, nad hráčím polem se napíše text "MISSION COMPLETE PROCEED TO PICKUP" což znamená, "MISE UKONČENA, POSTUPOJTE NA MÍSTO KDE VÁS VYZVEDNOU".

Úspěšně jsme se prokousali k závěru. Samotnou hru nechám na Vás. Dovolte mi ještě pár technických detailů hry;

1) Hra je opět zase a pouze jenom pro ZX Spectrum 128k, ale ti, kteří mají Spectrum 48k upravené na 128k nemusí zoufat. Hra nepoužívá druhou

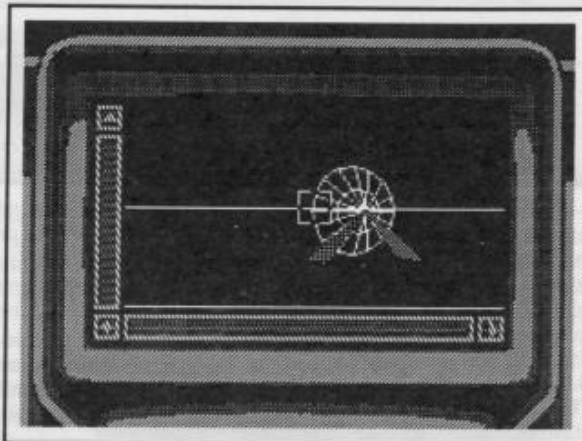
Videoram.

2) Zpracování prostoru je stylem freescape, ale poněkud jiné než u *incentive*.

3) Hratelnost je vynikající i při mnoha objektech na poli se rychlost téměř nemění.

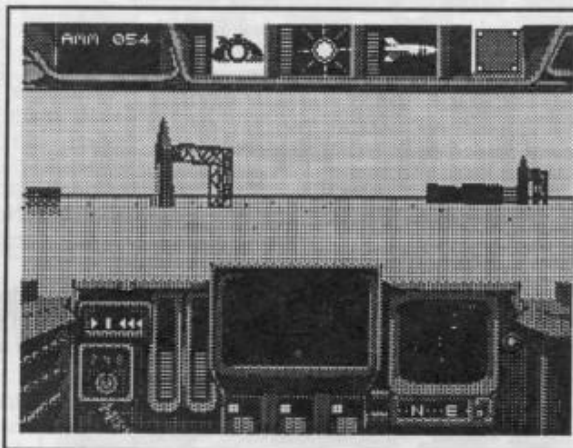
4) Hra obsahuje mnoho prvků dosud viděných pouze na 16ti bitových počítačích. Perfektně propracované elektrické vedení, postava člověka, který navíc mává (mise 3), železniční stanice, vlak, závory (pochopitelně se zavírají), semafor, most přes řeku a hodně jiných zajímavých věcí.

Hra je jak už jsem řekl pro 128k a vyčíst se jí dá pouze dlouhá časová smyčka po stisku fire při aktualizaci kazety. Tuto "chybu" můžete odstranit, tak že buď úplně odrušíte tuto smyčku, a nebo změníte její parametry v



šesté stránce (22). Víc Vám nepovím. Tu smyčku si už najdete sami.

Další drobný detail je hned v úvodu hry,



kdy se otáčí tank a hraje hudba. Rutiny, které počítají pozici tanku v prostoru používají instrukci DI (zakázání přerušení) což vede k tomu, že se hudba v některých momentech zpomaluje.

Takže to je konec

- JSH -

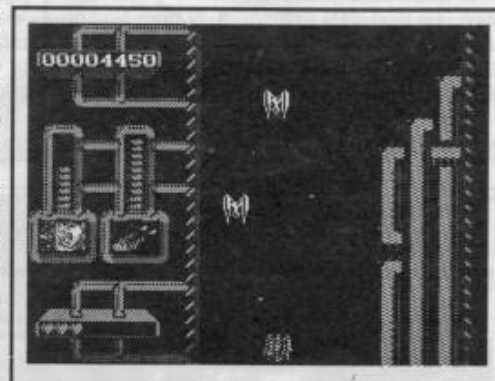
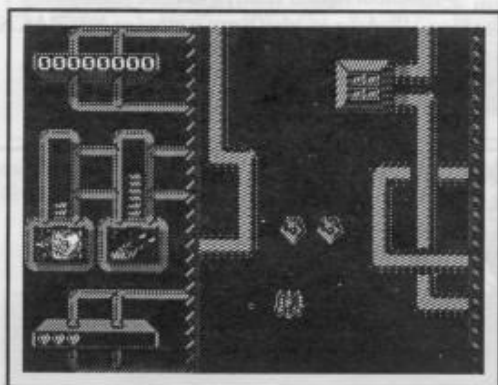
ANO GAIA 48

Simon C. Tillson 1992

A konečně tady máme novinku, určenou

majitelům Sinclairu 48. *Ano Gaia* je klasická střílečka, viz příložené obrázky. Dále k tomu není ani co dodat. Jen to, že hra obsahuje velice zajímavou grafiku, poprvé se v akci setkáváme s barvami, které Spectrum normálně neudělá.

Se třemi životy je hra téměř nehratelná, takže si ji musíte sami zapoukovat a pozor! Ani poté to nebude jednoduché! Druhá část prvního levelu je prakticky nehratelná, nemluvě o dalších dílech.



- JSH -

SKOOL DAZE

Autor ve hře neuveden

Pokud opomeneme některé slovenské textovky (Honba za klasákom) či Imaginovského MIKIEho, zjistíme, že skutečně velmi málo her na Spectru se odehrává ve škole (a pokud, tak má děj málo co společného se školním životem). Tuto citelnou mezeru zaplňuje právě hra **SKOOL DAZE** a její ještě lepší pokračování **BACK TO SKOOL**. Kdyby tuto hru vyrobili Code Masters, jistě by se jmenovala Super Student Simulator nebo nějak podobně. Bohužel, ve všech verzích Skool Daze, které se ke mně dostaly (byly dvě), chyběla jakákoliv zmínka o autorech, takže Vás zklamou - nevím to, ač bych rád věděl.

Poznámka korektora: Název hry SKOOL DAZE byste asi těžko překládali s pomocí slovníku, je to totiž zkomoleně napsaný nápis SCHOOL DAYS, neboli ŠKOLNÍ DNY. Stejně tak BACK TO SKOOL je vlastně BAG TO SCHOOL, neboli TAŠKA DO ŠKOLY.

(Universum)

Povězme si velmi stručně, o co ve hře jde. Byl jednou jeden puberták a jmenoval se **ERIC**. Ve škole byl poslední rok, takže jeho morálka byla poněkud uvolněnější. Navíc byl **ERIC** silný sadista - nejvíce ho těšilo, když za jeho zločiny pykali jiní. Jenže takových jedinců, jako byl **ERIC**, bylo na škole více a **ERIC**ovi silně znepríjemňovali život. A tak se **ERIC** rozhodl znepríjemňovat život jim, což mu přinášelo nebývalé uspokojení. A vy můžete jen hádat, kterou postavu ve hře budete ovládat.

Ano, je to **ERIC**. Klacek, který chodí pozdě na hodiny, neustále střílí prakem po učitelích, i když se na něj právě dívají, o hodinách bezdůvodně vstává či padá z lavice, čmárá na tabuli vulgární nápisy, leze tajně do ředitelny, mlátí mladší spolužáky i své vrstevníky a nakonec je za to všechno dočasně vyloučen ze školy. A to vše ve vašem podání. Gratuluji. Mně to také zpočátku nešlo. Ale do toho se dostanete...

Než si řekneme konkrétněji, co ve hře dělat, aby naskákali bodíci na vaše skóre, představíme si aktéry hry **SKOOL DAZE**.

O něco níže vidíte tabulku s postavami žáků. To jsou vaši vrstevníci. S nimi budete navštěvovat stejný ročník a právě tyto studenty musíte psychicky ničit tím, že na ně svedete vše, co se dá.

Boy Wander - na první pohled sympatický blondák, na další pohledy

vandal a násilník. Jeho oblíbenou zábavou je střílení prakem a čmárání na tabuli.

Angelface - poněkud prostoduchý rváč, který neumí nic jiného, než boxovat. Za to to dělá dobře...

Einstein - šprt, který ochotně odpovídá učitelům na všechny otázky a občas přidá jako prémii i nějaký ten bonz (obzvláště, když po něm často střílí prakem).



Kromě těchto h(r)ošků se na půdě školy vyskytují ještě žáci nižších ročníků, kteří vám sice příliš neublíží (maximálně vás vysunou z lavice - když vás uvidí učitel, máte smůlu) ale ani vám příliš nepomůžou. Mezi nimi je však jeden informátor, pracující pro vás. Občas za vámi o přestávce přijde a utrousí informaci, která může být užitečná. No a co by to bylo za školu, kdyby se na ní nenašel nějaký ten pedagog...

Přehlednou tabulku pedagogů najdete na následující stránce (snad...) ... a ne, je to ještě na této stránce (korektor).

Mr. Withit - učitel zeměpisu. Po zvonění se ještě chodí projít na terasu, takže na jeho hodiny můžete chodit trochu později.

Mr. Wacker - tvrdý a nekompromisní vládce zdejšího ústavu. Pozor na něj.

Mr. Rockitt - v pracovním oděvu a s tlustými brýlemi - učitel věd přírodních a technických.

Mr. Creak - osm let přesluhující dějepisec. Mějte úctu k šedinám...



Tito čtyři pánové Vám tedy budou znepríjemňovat studium na škole. A co vy? Vy budete provádět všechny lumpárny, o nichž jsem psal v úvodu, a to zejména podnikat útoky prakem na pedagogy, ovšem tak, aby se po jejich pádu na zem nacházel v jejich blízkosti jeden z vašich tří vrstevníků. V tu chvíli dostane poznámku on a vy si můžete připsat pár bodíků na skóre. Ovšem je také možné, že slíznete občas neprávem poznámku za cizí přestupek, nebo vám dokonce můžou učitelé ukřivdit - když Vás *Angelface* srazí tvrdou pěstí k zemi,

dostanete poznámku za to, že se válíte po zemi jako to zvíře.

Ano. Právě toto vás čeká, když vás učitel načapá při činu, nebo když si bude myslet, že vás načapal při činu. Trestry se udávají v "řádcích" a to od 100 do 800 řádků. Tyto řádky se vám neustále sčítají a až překročí 10000, budete dočasně vyloučen ze školy.

Učitelům se to nepodařilo. Vy to uděláte pomocí těchto kláves:

Cursor, INT2, KEMPSTON - tradiční ovládání. Vlevo a vpravo funguje klasicky, nahoru dolů je vlastně taky vlevo a vpravo - jdete-li však směrem ke schodům, podaří se vám po nich sejít (vyjít).

0 (nula, fire) - střelba z praku

H (hit) - boxování

S (sit/stand) - vstávání a sedání si (když nás sestřelí, musíme vstát)

W (write) - psaní na tabuli. Zpět do hry klávesou ENTER.

J (jump) - skáčete dočista jako nějaký kangaroo.

A ještě několik poznámčiček na závěr našeho povídání.

1) Choďte včas na hodiny. Učitelé vás vyhmátnou všude (pokud jim do toho nezazvoní)

2) O přestávce máte být na chodbě (ale když vás ve třídě nenajdou...)

3) Přestávka se řekne anglicky **PLAYTIME**.

4) V okénku dole se vždy píše jméno učitele, kterého máte tuto hodinu a třída, v níž se máte nacházet.

5) Snažte se nenasbírat moc "řádek" (**LINES**) tím, že se necháte nacytat u popsané tabule nebo při vašem střílení. I tak dostanete spoustu poznámek za to, že nesedíte v lavici.

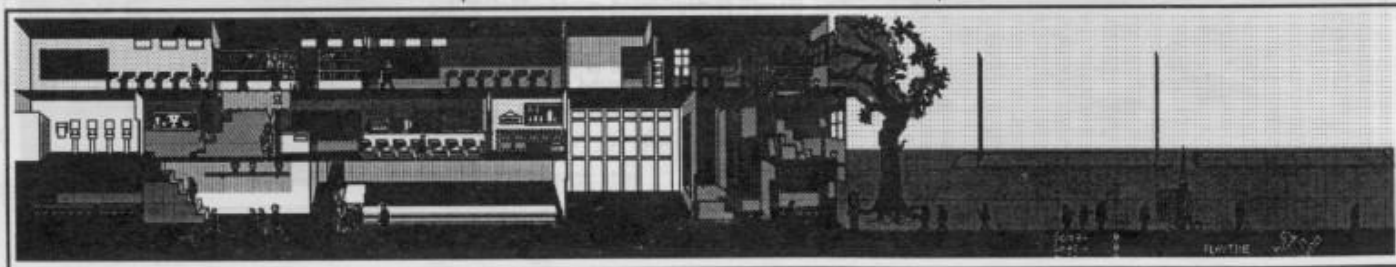
6) Střílet po učitelích ze zadu je jistější (a nejen po učitelích)...

7) Nelezte do "privátních" místností. Bude vám to jen ke škodě.

8) Pokud chcete, můžete si na začátku hry změnit jména postav podle vašich neoblíbenců (Petr Lukac, Pako Habera, Milouš de Břlek, Dr. Sladák, Scalex, Karel Gott, Bart Simpson a pod.) - program tuto možnost nabízí (i když to dost zdržuje).

9) A to je vše. Brzy u popisu hry **BACK TO SKOOL** na shledanou.

- Scalex nevlastní rukou -



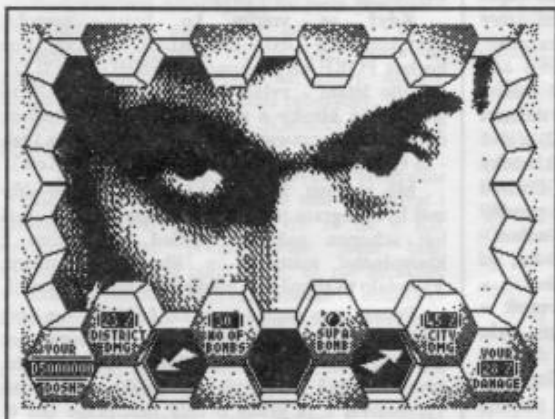
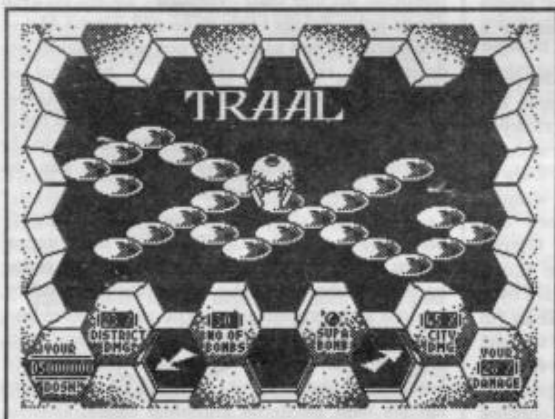
AMAUROTE 128/48

A binary design production 1987



Tentokrát tady máme poněkud starší (vlastně velmi starou) hru od známé trojice John Pickford (program), Ste Pickford (grafika) a David Whitaker (hudba a zvuky).

Vaším cílem je osvobodit rozsáhlý komplet měst od včelstva, které se ve



městech usadilo a vyhnalo původní obyvatele!

Od evakuovaných obyvatel dostáváte malé průzkumné vozítko, speciálně upravené pro vaši akci - je to jakýsi kráječ robot. Vše ale něco stojí a tak i koupě munice, speciální bomby, opravy vašeho stroje vám odčítá z konta peníze získané na začátku hry.

Ovládání je trochu atypické, ale brzy si na něj zvyknete. Dole uprostřed máte ikonu (včely, koruny a bomby), která spolu s šipkami signalizuje směr daného předmětu. Jeho vzdálenost od vás symbolizují barvy šipek.

Nyní několik rad do hry:

1) Napřed likvidujte včely a až pak zničte SUPRABOMBOU jejich královnu, která se vždy nachází uprostřed města.

2) Dávejte pozor na poškození vozidla (Your damage). Pokud přesáhne 70%, okamžitě si stiskem CAPS SHIFT vyvolejte menu a zvolte REPAIR. Vaše vozidlo bude uvedeno do původního stavu.

3) Neplýtvajte zbytečně bombami a mířte na včely přesně.

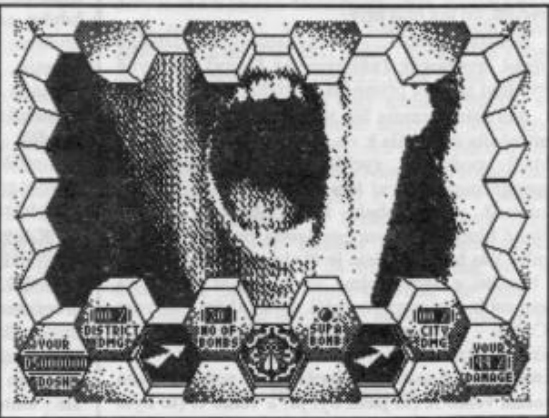
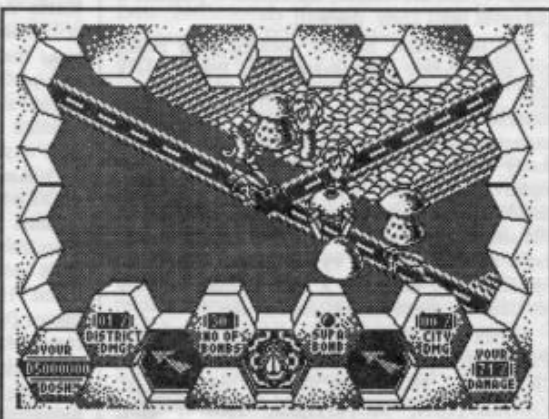
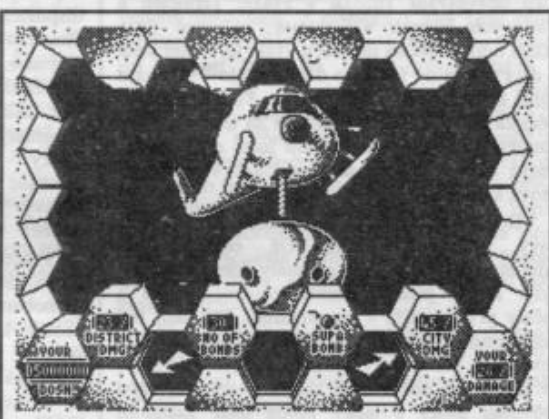
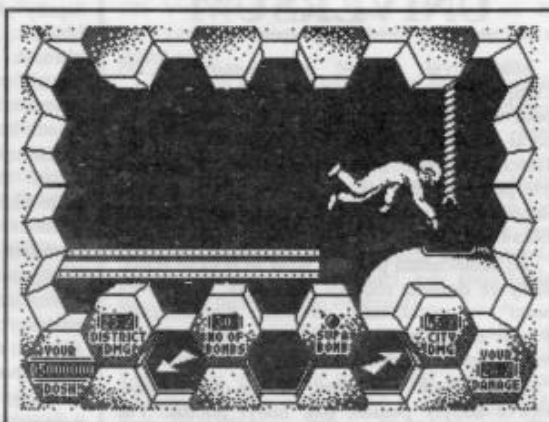
Když se vám podaří osvobodit město a zničit královnu, automaticky se přemístíte do dalšího města (můžete si vybrat).

Tato hra byla vytvořena pro Spectrum 128, 48 a dokonce existuje i na Atari 800/130 XL/XE.

Atarická verze se však na verzi 128 plnou digitalizovaných obrázků, výborné úvodní animace a perfektní stereo hudby vůbec nehrabe. Verze 48 neobsahuje z výše uvedených doplňků na 128 téměř nic. Na úplném konci hry na vás (48) čeká jeden digitalizovaný obráček a to je vše. Ale i tak, hra stojí za pozornost a proto pokud ji nemáte, obstarejte si ji, protože není špatná!

Jestli se vám zdá, že peníze ubývají rychleji, než je zdrávo, vložte do hry POKE 46364,33.

- JSH -



Jak začínali... (osobní zpověď)

UNIVERSUM

Můj první počítač se jmenoval CGS (Computer Game System) a vystříhl jsem si ho spolu s několika programy z časopisu ABC, zahrál jsem si s ním několik her - přistání na Měsíci a podobné. Neumíte si představit, jaká to byla práce, pořád dokola s kalkulačkou počítat šílené vzorcečky - ale přistál jsem mnohokrát a občas i úspěšně.

Ke skutečnému počítači jsem se poprvé dostal někdy ve druhém ročníku gymnázia, bylo to na "praxi" ve výpočetním středisku Keramických závodů. Stejnou přiležitost jsem si zopakoval ještě dvakrát a pak to přišlo....

Získal jsem Spectrum, trvalo to dlouho, nejprve nebyly peníze, pak nebyl počítač. Půl roku obvolávání Tuzexů v Praze, Teplicích a Karlových Varech a nakonec, nakonec jsem zkusil odpovědět na inzerát a vyšlo to. Dojel jsem si do Prahy a od jednoho dobrého člověka získal úplně nový počítač ZX Spectrum 48 KB. Zaplatil jsem za něj 6 000 Kčs, což byla na tehdejší dobu poměrně solidní cena. Později jsem si všiml, že tentýž člověk nabízí stejný počítač v novinách znovu a znovu....

Dodnes si pamatuji na zvláštní vůni, kterou nový počítač vydával. Už cestou jsem samozřejmě pročítal oba manuály, které v krabici byly. Doma jsem okamžitě přístroj zapnul a začaly první potíže - magnetofon. Ten můj byl stereofonní SHARP s rádiem, tedy naprosto nevhodný, jak se brzy ukázalo. Originální kazeta ale nahrát šla, dokonce ze sluchátkového výstupu (přívodní šňůra s jacky jinam totiž zasunut nešla). Výuka klávesnice, informace o počítači a také první hra - Thro' the wall.

Měl jsem štěstí, hned z počátku jsem získal čtyři plné kazety devadesátky her. Potíž byla s jejich zkopírováním - nakonec jsme to s kamarádem vyřešili pomocí dvou maďarských magnetofonů stejné značky a kopírování přímo z kazety na kazetu. Jediný problém byl, že při kopírování musel být regulátor hlasitosti zdrojového magnetofonu na maximum (ovlivňoval totiž sílu signálu), nakonec jsme to zvládli - tlustá vrstva peřin a polštářů vykonala své, ale přesto nejméně 4x90 minut se z našeho bytu ozývaly sice tlumené, ale přesto přišerné zvuky.

A pak to začalo - hry - COMBAT LYNX, SPORT HERO, CAULDRON, MANIC TWO, TIR NA NOG, SQUASH, FRANKIE, LOCOMOTION, MONSTERS, TRANS AM, ARMAGEDDON a spousta dalších. Člověk nevěděl, co má hrát dřív.

Neustálé problémy s magnetofonem částečně vyřešil šroubovák (nastavování hlavy) a hlavně vyvedení signálu přímo od reproduktorů.

Ale nejen hraním živ je člověk, jakási tajemná vnitřní síla mě hnala k vlastní tvorbě. BASIC jsem zvládl poměrně rychle a hned jsem si naprogramoval první hru - nevím jestli ji znáte, vzdáleně se podobala PAC MANovi, prostě nějaké bludiště, v tom jste jedna potvora vy a ta druhá Vás honí, úkolem je vysbírat všechny tečky v bludišti a samozřejmě se nenechat sežrat. Já jsem tuhle hru poprvé viděl na nějakém sálovém ECéčce - já jsem byl velké X a honily mě dvě osklivé hranaté závorčky. Nazval jsem to celé **Honička** a neustále vylepšoval - nejprve jsem změnil velké X (to jsem byl já) a velké O (to byla

ta potvora, co mě honila) na grafické znaky a vytvořil něco, čemu se při dobré vůli dalo říkat "sprity". Potom jsem použil barvy a přidal editor bludišť.. Nakreslil jsem k tomu celému úvodní obrázek (Kdepak Art Studio, hezky PLOT a DRAW a PLOT a DRAW a). No ale tím celé vylepšování neskončilo - naučil jsem se dělat vlastní znakové sady (odhalil tajemství systémové proměnné CHARS) a samozřejmě si nakreslil novou znakovou sadu - trochu se podobala písmečkům ze hry CAULDRON ale malá písma už byla stoprocentně moje. A dokonce jsem tam měl i češtinu.

Spolu s odhalením tajemství CHARS došlo k vytvoření prvního z poměrně dlouhé řady mých znakových editorů. Už ani nevím, co vlastně všechno uměl, moc toho asi nebylo ale znaky se s ním vytvářet daly.

Další pokus o zásah do HARDWARE byl naprostým fiaskem - nevím, jestli jste i vy měli k dispozici informace o zbytečně velké dimenzovanosti transformátoru pro ZX, přý dokonce může zničit i počítač - tak jsme stavěli stabilizátor napětí. Po jeho připojení se Spectrum sice tolik nehřálo, ale když běžely některé programy (FIGHTER PILOT), nevypadlo to hezky - pruhy přes obrazovku a podobně, nakonec jsme ho odpojili a zbyl z něj jenom takový "chladič - podložka", sloužil dloho a dobře.

Práce na **Honičce** pokračovaly dál. Přišel jsem na to, že by okolí bludiště vůbec nemuselo být jednobarevné a zkusil ho poskládat z různých znaků, které vytvářely obrázky - zdi, stromy, pobřeží, voda, kaktusy a co já vím co ještě. Daly se z toho skládat docela hezké ostrovy (tedy alespoň si to myslím). Vytvořil jsem jich skoro dvě desítky a různé pojmenoval - například **Ostrov Karla Záhorobského** (to byl spolužák ze školy, příběh sám pro sebe), **Hřbitovní ostrov** (tam byly samé malé hrobočky) a **Ostrov Pivní džbán** (to zase podle celkové tváru). Hra už se v té době jmenovala **Víla Amálka a Tlamichové**. Víla Amálka proto, že se nám s kamarádem podařilo najít v týdeníku Československá televize obrázek Víly Amálky za mřezemi a překreslit ho (opět PLOT a DRAW, PLOT a DRAW - program, který to dělal měl něco kolem 30 KB) na obrazovku, Tlamichové proto, že to, co jsme v rastru 8x8 bodů nakreslili jako potvora, mělo hlavně velkou tlamu. Hra dostala ještě úvodní menu, definici kláves (to sem byl už tehdy bourák, co?) a dvě různá ukončení - v jednom se nakreslil klasický hroboček (náhrobek s křížem) a zahrál Pochod padlých revolucionářů (ten jsme ale později nahradili jakousi nic nefikajícím smuteční hudbou - bylo to takové malé ideologické zakolísání, co kdyby na nás někdo vletěl, myslíme jsme si), v druhém se nakreslil jakýsi ohromný pomník a zahrála veselá melodie - na pomníku bylo napsáno cosi jako: *Pomník na počest hrdivé a statečné Víle Amálce za vítězství nad Tlamichy*. Později jsem hru ještě zkompiloval pomocí kompilátoru a značně tím zrychlil. Mohl jsem si dovolit přidat i dvě nebo tři potvory, ale nakonec jsem to zase zrušil - ty tři byly doopravdy moc.

Když už píšu o kompilátoru basicu - byl to M-CODER 2, ke mě se ovšem dostal jako **COMPILER**. Nejdřív jsem nevěděl, co mám s programem dělat - vždycky se nahrál, napsal na obrazovku nějaké podivné anglické nápisy a také stiskni nějakou klávesu - nu což, poslechl jsem a program se vymazal, tak jsem ho nahrál znovu, ale tentokrát jsem mu vymazání nedovolil,

BREAKnul jsem to! Z úvodního basicu jsem odstranil příkaz NEW, o kterém jsem věděl, že za všechno může, divil jsem se, že někdo může dát do programu takový destruktivní příkaz a to hned po nahrání. Pak jsem zkusil provést nabízený příkaz RANDOMIZE USR 60000, začaly se dít podivné věci, vypsal se nějaký text a hlavně osklivé hlášení ERROR a v listingu programu blikal otazník před příkazem LOAD. Nebudu Vás dlouho napínat, nakonec jsem se s M-CODERem 2 spátelil a vděčím mu za několik svých programů (z velké části už neexistují). Tento kompilátor značně ovlivnil mě hry - uměl totiž jenom celá čísla a tak si umíte představit příkaz BEEP (on ho uměl udělat i s kratším časem, jenže to jsem nevěděl, takže jsem BEEPal dlouze a nepřítli často).

Druhá hra, kterou jsem napsal, se jmenovala **Planeta pokladů** - inspiraci jsem bral ze stejnojmenného filmu, který zrovna běžel v kině. Skládalo se to ze tří hlavních částí - nejprve se člověk musel dostat na planetu pokladů - to bylo neustálé startování a přistání na Měsíci, jenom se to odehrávalo na různých planetách s různě velkou gravitací, mezitím byl vždy hyperprostorový skok, který ovšem mohl skončit v nějaké černé díře - to neznamenalo smrt ale odhození do jiné části "vesmíru", kromě černých děr na hráče čekaly i roje meteoritů. Druhá část se odehrávala přímo na planetě pokladů - museli jste doletět od rakety k základně a nenechat se sestřelit. Třetí část se odehrávala v základně (tohle byl klasický 3D labyrint v němž se na jistém místě nalézal "poklad" - zajímavostí této části bylo, že labyrint nebyl průchodný a museli jste na několika místech prorazit zeď).

Kromě her jsem programoval i "užitkové" programy - známý SUPERTAPEX jsem napsal znovu (z toho starého jsem použil strojovou rutinu na čtení souborů z pásky, zjišťování jejich délky, leaderu a správnosti nahrávy) a zkompiloval ho - důvodem byla pomalost původního programu, který obvykle nestačil po přečtení hlavičky začít kontrolovat data. Vznikl SUPRTAPEX, který můžete najít na třetím kompletu Public Domain ZXM.

Když se vrátím ke hrám, nemohu nezpomenout UNDERWURLDE (dohráli jsme ho bez POKE až do konce) a FIGHTER PILOT (ať žije létání - zvlášť když člověk neví, že to letadlo má klapy a přistává bez nich, dlouhou dobu mi byla runway krátká, ale přistál jsem i tak - na ty klapy jsem přišel až s TOMAHAWKem).

Mé největší "basicovské kompilované" dílo měl být program pro kreslení mapy krajiny - měl být schopen nakreslit pohled na krajinu z libovolného místa a v libovolném směru. Vypadlo to slibně, ale nikdy jsem ho nedokončil.

Potom jsem napsal program **LOGIC**, ten zase můžete vidět jako přímii na souboru her **EXPEDICE**. Vzniklo i druhé pokračování, jenomže když jsem byl dost daleko, počítač se vyresetoval a já přišel o celý den práce (od té doby ukládám rozpracovaný program nebo text raději častěji).

Když mluvím o tom ukládání, přidám i nahrávání - začal jsem mít čím dál tím větší potíže s nahráváním - to, co dříve šlo bez problémů šlo nyní stále hůř a hůř - příčina byla opět prostá, ale pobavte se nad mojí "blbostí" - na magnetofonové hlavě byl takový menší hnědý flek, dost tlustý - za celý rok mě ani jednou nenapadlo hlavu vyčistit.

Vzhledem k tomu, že se blížila maturita, musel jsem se zabývat i školou - vznikl program na zkoušení české a ruské literatury. Byl to takový dvířákový program, ale svůj účel splnil.

Ve stejné době jsem navštěvoval autoškolu - vzniklo celkem šest programů na zkoušení testových otázek. Testy jsem udělal bezchybně, poněkud horší to bylo s jízdami, no ale na třetí pokus mi to fidičské oprávnění přeci jen dali (ještě ho mám schovaný...).

Současně s těmito programy jsem vytvářel ještě grafický program LEONARDO, uměl to, co se dá naprogramovat pouze v BASICu. Možná si řeknete, proč jsem se s tím dělal, když existovalo Art Studio - existovalo, ale já jsem jej neměl a neměl jsem ani žádný jiný grafický editor.

Asi po roce došlo k zásadnímu obratu - dostala se mi do ruky příručka o strojovém kódu Z80 od Karolinky.

Po necelém roce provozu mi začala odcházet membrána v počítači - novou shánět se mi nechtělo (ani jsem nevěděl kde), tak jsem se pokusil si ji vyrobit sám. Vytvořil jsem asi pět prototypů membrány - všechny se vyznačovaly tím, že sice spínaly ale po krátkém čase přestávaly rozepínat. Nakonec jsem na "membrány" zanevřel a zkusil něco jiného, což bylo asi ve verzi 3.0 korunováno úspěchem - použil jsem cupretitovou destičku a na ní připájel měděné kontakty - plíšky z relétek (u "gumáka" to šlo, u "pluska" by to bylo horší), přidal nějaké drátky a světe div se, funguje to dodnes (už je to 6 let).

Žádný učený z nebe nespadá a tak jsem i já prováděl věci, nad kterými mi dnes zůstává rozum stát. Například u instrukce **sbh hl,de** se mi v jednom programu stávalo, že občas odečítala o jedničku víc, než měla. Dlouho jsem hledal příčinu, uvažoval jsem i o chybě počítače, dokonce jsem ho rozebral (přesněji řečeno otevřel) a zase složil, nic se nedělo. Skutečnost, že BASIC fungoval bez chyb, a strojový kód "zobil", mě dlouho pronásledovala... nemohl jsem ji totiž pochopit, zvlášť když jsem věděl, že BASICový interpret je naprogramován ve strojovém kódu.

Začal jsem se ho učit s pomocí programů GENS a MONS. Pokud jste s těmito programy měli někdy tu čest, víte, že nepatří mezi nejlepší. I vznikl projekt assembleru, který by používal basicový editor. I tento program - ASSEMBLER - můžete najít na disketě Public 3.

Po dokončení ASSEMBLERu jsem pracoval i na monitoru, ale se začátkem školy (Matematicko-fyzikální fakulta UK) práce na programu stále polevovaly až ustaly docela, kromě toho jsem získal i další programy, známý GENS ale tentokrát s FULL-SCREEN editorem a MON2, s tím už se dalo pracovat.

V té době jsem začal uvažovat o programu WLEZLEY (USER 1), původně měl obsahovat i disassembler, ale ukázalo se, že se do 4 KB nevejde. Kromě programování jsem také uvažoval o tom, co budu dělat, až mě ze školy vyhodí - blížilo se totiž první zkouškové období (s chmurnými pocity jsem prohlížel vyvěšené inzeráty typu *přijímáme programátora*). Nakonec první zkoušky dopadly příznivě - skóre 3,3,1 - ale další semestr na škole mohl začít.

Na hardwarovém poli došlo k velkým změnám - zakoupil jsem si *souřadnicový zapisovač Minigraf ARITMA 0507* za pouhých 6000 Kč - popsal jsem s ním spoustu papíru....

V letním (druhém) semestru jsem jako zápočtový program vytvořil první PRESSOR, program na kompresi obrázků (s jeho prapranukem se můžete setkat na kompletu USER 2).

Následovaly další verze programů WLEZLEY.

Prázdniny znamenaly vznik kompilátoru IUC-88. I ten můžete najít na kompletu Public 3, napsal jsem ho s pomocí ASSEMBLERu, byl to nejrozsáhlejší program, který jsem na něm vytvořil - a pak se samozřejmě objevil Hisoft BASIC - zase jedna téměř zbytečná práce.

Další v pořadí byly kopírovací programy GARGANTUA a PANTAGRUEL - měly počítatla při nahrávání, dokázaly komprimovat data při nahrávání a ještě další věci - jejich přímé následníky můžete najít opět na kompletu USER-1. Oběma programům předcházela úprava programu ALIENS, tam jsem si poprvé vyzkoušel využití časových smyček v nahrávací rutině - při nahrávání hry na obrazovce roloval text s informacemi o hře kombinovaný s obrázky - skrolovalo se v okénku o rozměrech asi čtvrtiny obrazovky - psaním tohoto loaderu jsem zaplnil celé tři dny (Velikonoce).

Když zavzpomínám na školu, vybaví se mi nejsilnější postava pana profesora Vopěnky (pokud se domníváte, že je to ten, co vedl dva roky naše školství do záhuby, tak máte pravdu). Jeho přednášky z Teorie množin byly zajímavé, počátku to byly spíše takové filozofické úvahy (vždycky na jeho přednáškách jsme uvažovali o přejmenování školy na Matematicko-filozofickou fakultu). Zkoušel zajímavým způsobem - do svého kabinetu vzal vždy tři studenty, když se trochu podusili v těžkém koufi, zadal prvnímu z nich (tomu nejvíce vlevo, jindy nejvíce vpravo a někdy i nejvíce uprostřed - pro každou trojici ale bylo rozvržení stejné) otázku, když ji nevěděl, přešla na druhého a posléze na třetího - výsledek tohoto způsobu byl ten, že třetí student dostával ty nejtěžší otázky, byly to ty, co první dva studenti nedokázali zodpovědět - jistě Vám nemusím říkat, který student v pořadí jsem byl (napodruhé ale podlehl a známku mi dal).

Začátek druhého roku ve škole (třetí semestr) mi způsobil několik problémů - ze čtyř zkoušek jsem čtyři (tedy plných 100%) musel skládat dvakrát. Po stránce programátorské se objevil základ dnešního kompletu USER 2 - první LOADERMAKER (dnes je tento program součástí PRESSORu) a jednoduchý pakovač hlavního bloku hry.

Čtvrtý semestr byl velice plodný - začal vznikat ladicí systém PROMETHEUS. Trvalo to většinu prázdnin a byl hotov assembler, monitor jsem dokončil až v semestru pátém. PROMETHEUS je symbolické jméno, samozřejmě, bájný PROMETHEUS přinesl lidem oheň, bůh ví, co ale vlastně přinesl PROMETHEUS programátorům? Radost mi udělal jeho recenzent ve FIFU, odhadli totiž proč se program jmenuje tak, jak se jmenuje.

Pohnuté listopadové dny mi umožnily vytvořit další verzi PRESSORu, tentokrát to byl PRESSOR V - pokud jej máte, můžete si všimnout odrazu vnějších událostí na program (Československá vlajka místo obrázku - normálně by mě to asi těžko napadlo, neboť míra mého nacionalismu se blíží nule).

Pátý semestr měl zkouškové období až po listopadu 89, profesori byli hodnější než obvykle (1, 1, 1) a tak bylo více času. Začala vznikat EXPEDICE - první program s textem v borderu - alespoň jsem si to myslel, ale pak se objevil SENTINEL. EXPEDICI jsem nabídl Samostatné drobné provozovně ONV Praha 4, dostal jsem na ní dokonce i smlouvu, při udávání (tak bych nazval nabízení programu distributorovi) jsem se setkal (kdybych napsal seznámil, bylo by to příliš troulalé, protože pochybuji, že si mě vůbec pamatuje) s Vladimírem Jiránkem - obrázky v EXPEDICI jsou od něj. Když jsem se ho ptal, jestli souhlasí se spoluautorstvím, musel jsem mu program předvést - souhlasil.

Téže firmě jsem se pokusil udát také PROMETHEA, ale bez úspěchu (prý "dělají" jenom hry). Tak jsem to zkusil u 602 ZO Svazarmu (Mikrobáze), tam by to byli zkusili ale nakonec jsem nechtěl já - na obzoru se totiž objevila PROXIMA.

PROXIMA vznikla v červnu 1990 (letos jsme slavili třetí výročí). Založili ji spolu Petr Podafil a Jiří Vondráček (s tím jsme se něco na ZX navývali), já jsem v té době byl programátor na volné noze.

To už se blížil ke konci šestý semestr a začal jsem pracovat na ORFEOVI, dokončil jsem ho o prázdninách, byl ty první program, který PROXIMA prodávala.

Na konci prázdnin a začátku sedmého semestru vznikl BAD DREAM, inspirace byla jasná - program SOKO-BAN na PC (ten jsem viděl první právě v pohnutých listopadových dnech - bylo to pozdě v noci, kdy jsme hlídali školu, aby nám ji nikdo neukradl).

Po BAD DREAMu jsem začal psát svůj nejznámější program - DESKTOP, byl hotový do konce roku, pak jsem musel psát ovladače pro spoustu tiskáren. Asi někdy v dubnu jsme ho (tedy PROXIMA) začali prodávat a to s poměrně velkým úspěchem. Z počátku s ním byly různé problémy - chybám se člověk nevyhne. Dodnes mi vstávají vlasy na hlavě hrůzou, když si vzpomenu na ovladače pro různé levné "socialistické" tiskárny - obzvlášť D100, PRT42G a NL2804, s těmi bylo nejvšelejší. Například PRT42G jsem dopravoval na kole z MOSTU do ÚSTÍ (a to jsem si v rámci prázdninového výletu cestu protáhl přes Roudnici nad Labem a kopce Českého Středohoří), zajímavé bylo, že před jízdou nepracovala vůbec, po jízdě sice špatně, ale alespoň nějak. Návrh na další projížďku jsem však již odmítl.

Od té doby jsme začali připravovat manuály na DESKTOPu, zvlášť po vzniku prvního ULTRA LQ (teď jsme jeho poslední vylepšenou verzi také zařadili do distribuce).

Objevila se disketová mechanika ze Skalice - D40. Následovaly úpravy programů pro ní - začalo to DESKTOPem. Disketové operace by se daly napsat i lépe, ale v té době jsme nevěděli jak, dneska už to (kvůli kompatibilitě) měnit nebudu (nešikovná je hlavně existence čtyř souborů pro jeden text).

V sedmém semestru (podzim 1990) jsme se seznámili s Miroslavem Hlavičkou (SCALEXem) a získali tak kvalitního hudebníka. Zajímavé jsou okolnosti, za nichž k tomu došlo - Mirek nám napsal sám, ale sám neví, odkud se o nás (a ORFEOVI) dozvěděl - prý našel ve schránce reklamní leták. V roce 1991 o Velikonočních jsme s Jirkou podnikli tzv. "Legendární cestu do Boleslavi" - vyrázili jsme na kole a bez přesné adresy - předpokládáný příjezd byl kolem poledne, dorazili jsme tam nakonec až někdy ve čtyři odpoledne a začali hledat, znali jsme jméno dotyčného a jméno ulice (nutno říct, že jsme s tou ulicí měli štěstí, tehdejší Marxova byla ulice, kterou jsme do Boleslavi přijeli), potíž byla v tom, že před Boleslaví je vesnička Debr a ona ulice je tam také, první hledání tedy nemělo úspěch, ale dozvěděli jsme se, že ještě nejsme v Boleslavi. Nakonec jsme hledanou adresu našli (poradil nám nějaký mladík), Mirek nebyl doma ale rodiče nám nabídli, že můžeme počkat a dokonce i přespat, což jsme vzhledem k fyzickému stavu Jiřího Vondráčka přijali. Poznali jsme i současného ilustrátora ZX Magazínu Miloše Bílka, stál ve dveřích a nechtěl Mirkovi věřit, že jsme z Proximy, později toho trpce litoval (o tom, že by nás pustil dovnitř ani nemluvě).

(pokračování na straně 21)

Nový shareware od PROXIMA...



GEORGE K.'S "FILE MANAGER"

(c) 93 Proxima, Software nové dimenze

Program "File Manager" (dále jen FM) postupně nahrazuje na distribučních discích firmy Proxima starý UniRun. Oproti němu poskytuje uživateli více funkcí, lepší vzhled a příjemnější ovládání. Následující text by Vám měl napomoci v prvních chvílích s FM a jeho zkrácenou verzi najdete na každé disketě.

Na distribučních discích je FM nastaven do české verze, zobrazuje pouze spustitelné soubory a k nim ilustrační obrázky. Překopírujete-li si ho na nezalepenou disketu, budete si moci nastavení pozměnit a uložit.

Při pohledu na obrazovku vidíte následující - v pravém horním rohu název založené diskety (zkuste na něj najet šipkou a vystřelit, třeba se něco dozvíte), o kousek níž řádek s příkazy a nakonec seznam vybraných souborů (podle nastavení). Při vystřelu na název disku se otevře "okno" systému KoZa Windows Projects - normálně jsou data okna čerpána z paměti (v tom případě počítač vypisuje "Žádné informace o tomto disku"), ale lze si vytvořit vlastní okno a rozstrkat ho do bootu na disku - bohužel, organizace okna a jeho uložení jsou tak složité, že jejich popis by zabral více místa, než celý tento článek, a proto Vám pouze poradím, že můžete použít TOOLS 40/80 a zkusit editovat nějaký distribuční disk s FM (všechny mají vlastní "okno" v bootu) a zkusit na to přijít sami...

Výpis souborů je formátován (do jednoho až tří sloupců) v závislosti na jejich počtu. Na obrazovku se vejde maximálně 43 souborů bez obrázků nebo 12 souborů s obrázkem; má-li se jich zobrazit víc, stránkujete výstřelem na funkci "Zpět/PgUp" a "Další/PgDn". Je-li před názvem souboru velké písmeno a dvojtečka, znamená to, že soubor je spustitelný. Spustit ho můžete buď tak, že na něj ukážete šipkou a vystřelíte (střelba na ilustrační obrázky nemá smysl, palte pouze po názvech), nebo

rovinnou stisknete CAPS SHIFT + písmeno před dvojtečkou (horká volba).

Pokud najdete na název souboru a stělu chvíli podržíte, nedojde ke spuštění souboru, ale bude vyvolána funkce VIEW, která pracuje stejně jako v TOOLSu 80; tj. zobrazuje úvodní obrázky ze SNAPů a obrázky v souborech BYTES (i pakované) a PROGRAM (jen pakované). Pokud má Váš joystick problémy a při delším stisku FIRE občas vrátí signál "nestisknuto", můžete



namísto delšího držení stělu udeřit do klávesy "3", která je horkou volbou pro VIEW.

V příkazovém řádku si povšimněte volby "Nastavení/SetUp". Výstřelem na ni otevřete okno, obsahující tyto položky (když je položka "nastavena", je před ní znak "x"):

Česky / Czech - česká verze

Anglicky / English - anglická verze

Klávesnice / Keyboard - ovládání šipky klávesnicí. Klávesy jsou definovány takto:

Q W E R T - nahoru, A S D F G - dolu,

I P - vpravo, U O - vlevo,

B N M S S P ENTER - střelba

Kempston IF - joystick přes interface v počítači

Kempston D40 - joystick přes D40/D80. I když máte nastaven kempston (IF, D40), můžete stále používat pro ovládání šipky klávesnici - to proto, aby, pokud byste

joystick neměli připojen, se program "nekousnul".

Spustitelné / Runables - zobrazit jen spustitelné soubory (Programy a SNAPy)

Všechny / All - zobrazit všechny soubory

Obrázky / Pictures - zobrazovat ilustrační obrázky u souborů. Téměř každý soubor může mít svůj ilustrační obrázek. Obrázek je na disketě uložen jako soubor BYTES, který má stejný název jako soubor, ke kterému patří. Např. u FM, který je na disku pod názvem "run", to bude vypadat takto (ověřte si basicovým CAT):

P run 8704 ---ARWED

B run 108 HS--RWED

Pokud soubor obrázek nemá mít (není nastaven atribut "A"), je místo něj zobrazen smutný obličej (barva podkladu je rozdílná podle typu souboru). Pokud soubor obrázek má mít (je nastaven atribut "A") a tento není na disku nalezen, je místo něj zobrazena lebka a dvě kosti. Obrázky mají rozměr 3x4 znaky a jsou organizovány následovně: 1. pixlový řádek (4 bajty), 2. pixlový řádek, ..., 8. pixlový řádek, 1. atributový řádek, 9. pixlový řádek, ..., 16. pixlový řádek, 2. atributový řádek, 17. pixlový řádek, ..., 24. pixlový řádek a 3. atributový řádek - dohromady 108 bajtů. Obrázek si můžete nakreslit např. v Art Studiu a pak pomocí jednoduchého programku v Basicu nebo strojovým kódem "přesypat" někam do paměti a tam odtud uložit na disk. Pokud Vám je organizace videoram záhadou, můžete si obrázek nakreslit na milimetrový papír, pak přepočítat a nakukat v číslech do počítače (108 čísel není tak mnoho, ale udělejte si padesát obrázků, co?). Souboru s obrázkem nastavte atributy "S" (aby ho FM dokázal identifikovat) a "H" (aby se zbytečně neprodužoval výpis CATALogu) a změňte mu jméno podle programu, ke kterému patří. Nezapomeňte u programu nastavit atribut "A".

Schované / Hidden - zobrazovat i soubory s nastaveným atributem "H"

Čekaj na FIRE / Waitfire - po nahrání FM čekat na stisk FIRE

Otevřít port / Open port - před spuštěním souboru otevřít port v disketové jednotce (umožňuje současně používat kempston připojený do počítače a tiskárnu připojenou přes mechaniku)

Další volbou v příkazovém řádku jsou "Příkazy/Commands". FM má následující direktiva:

Cat A - katalog disku A (přímá volba "1")

Cat B - katalog disku B (přímá volba "2") - pokud nemáte druhou mechaniku, dělá to, co Cat A

Save FM - uložit (zkopíruje) FM s daným nastavením (přímá volba "CS+SS")

Verify disk - obsah disku je "posčítán" a porovnán s hodnotou v bootu (první sektor disku). Během verifikace bliká border podobně jako při formátování, celá operace trvá různě dlouho podle velikosti formátu. Program čte jeden sektor disku za druhým a provádí dva součty: jeden pomocí instrukce

ADD a druhý pomocí XOR (podobně to dělají páskové operace LOAD a SAVE v Basicu). Zjištěné hodnoty porovnává s hodnotami v bootu na adresách 202 a 203; když se hodnoty nerovnájí, znamená to, že obsah disku se od minulé verifikace pozměnil - buď Vaším zásahem (na disk jste něco uložili nebo přejmenovali nějaký soubor, atd.) nebo zásahem operačního systému (disk byl jednoduše nakopnut, aniž by se Vás kdokoli v zeptal). Nové hodnoty kontrolních součtů dokáže FM na disk



uložit. Pokud je u distribučního disku při verifikaci zjištěna chyba, je nahrávka fyzicky vadná a je nejlepší disk reklamovat.

Quit - ukončení práce s FM (přímá volba "CS+0")

Poslední funkcí FM je "Čti mne/Read me". Tato funkce hledá v adresáři soubor s názvem "ReadMeGKFM" a pokud ho najde, zobrazí jej. Vlastní "čtimne-text" si můžete vytvořit, pokud dodržíte tato pravidla:

- Používejte pouze znaky s ASCII kódy 32-157. Znaky 32-127 jsou běžné podle tabulky ASCII (podívejte se do manuálu k počítači), znaky 128-157 jsou české znaky rozložené podle DESKTOPU (ěĚěŠšŠčřřžžýYáÁííďďňňóÓřřůůÚÚ).

- Nestarejte se o délku řádků, program si vše ohlídl sám. Konce řádků se v textu nijak neoznačují (asi proto, že vůbec neexistují) - prostě pořádk pište a až budete chtít ukončit odstavec, vložte do textu kód 13.

- Celý text ukončete kódem 255 a uložte na disketu pod názvem "ReadMeGKFM".

Oproti všem ostatním "unirunům" dokáže FM zobrazovat názvy souborů v češtině a ne jenom v cestine. Chcete-li, aby se Vás soubor jmenoval např. "Šemík", proveďte následující: nahrajte si TOOLS 40/80, založte disketu se souborem a zvolte editaci disku (CS+E). Mezi adresami 3072 a 7167 leží uloženy názvy souborů - dejte si trochu práce a najdete soubor "Semik". Zvolte editaci (CS+1), přejděte do módu zadávání čísel (Enter) a přepište na české znaky (opět podle Desktopu). V TOOLSu se pak sice název souboru "Šemík" stane částečně nečitelným, ale ve FM bude v plnohodnotné češtině...

Horně úspěchů při práci s File Managerem Vám přeje

George K.

Tolstoj

V ZXM 1/93 se objevila malá reklama na nový program z distribuční nabídky Proximy - na slovníkovou databázi Tolstoj. V tomto článku máte možnost dozvědět se o Tolstojovi více podrobností - neváhejte a čtěte dál.

Představte si úplně jednoduchý program: telefonní seznam, který má čtyři položky. První položka obsahuje jméno, je povinná a data jsou podle ní seříděna. Druhá položka obsahuje adresu, třetí telefonní číslo a čtvrtá libovolnou poznámku - tyto položky jsou nepovinné a není třeba je naplnit. Takto by se - úplně nejjednodušeji - dal charakterizovat základní princip Tolstoje... a uznejte sami, že by na tom nebylo nic lákavého, kdyby... kdyby tu nebylo pár maličkostí, které mohou několikanásobně pozvednout naznačenou úroveň. Jaké maličkosti to jsou...?

Základním stavebním prvkem databáze jsou "slovička", která mohou mít až čtyři rozdílné položky (v případě telefonního seznamu chápeme jako "slovičko" účastníka + všechno, co k němu patří). Každá položka má svoji vlastní znakovou sadu (jakoby náhodou se jedná o znakové sady z Desktopu), což umožňuje nejen snadné grafické rozlišení (použitím tučných fontů, kurzív, speciálních znaků, atd.), ale dává to i šanci opustit hranice naší latinky a vypracovat si databázi v azbuce nebo v runách. K tomu si můžete určit nový sled jednotlivých znaků - vytvořit si vlastní "abecedu" (např. pokud by Vám vyhovovalo mít "H" před "A", ale daly by se najít i rozumnější důvody). S položkami můžete v podstatě nakládat, jak uznáte za vhodné - libovolně je opakovat nebo vynechávat - a právě tohle je základ pro složitější slovníkové databáze (klidně můžete bez větších problémů opsat do Tolstoje kus jakéhokoliv slovníku).

Další důležitou vlastností Tolstoje je, že Vás dokáže na základě zadaných dat přezkoušet. Ne vždy to má smysl - asi by se stěžil našel člověk, který by se chtěl nechat otestovat ze znalostí telefonního seznamu nebo své knihovny, ale v případě slovníků (hlavně pokud se chcete zdokonalit v cizím jazyce) to svoji opodstatněnost má - jak vidíte, můžete Tolstoje použít nejen jako databázi, ale i jako výukový program. Samozřejmě, že se nemusíte omezovat pouze na jazyky - bez velkých problémů Tolstoje naplníte údaji z mnoha jiných oblastí (literatura, dějepis, zeměpis)... že by nová noční můra pro školou povinná dítka (do nového školního roku s novým Tolstojem)?!

Vraťme se k dalším vlastnostem Tolstoje: Tolstoj spolupracuje pouze s disketovou jednotkou, a proto máte zabezpečen celkem rychlý přístup ke spoustě dat - Vaše databáze může být až 1 megabyte dlouhá. Data jsou na disku uložena v souborech; každý soubor přísluší jednomu písmenu z definované abecedy (u zmíněného telefonního seznamu by to znamenalo, že v jednom souboru by se nalézali účastníci, jejichž příjmení začínají od "A", v druhém od "B", atd.). Díky této organizaci se dají jednotlivá "slovička" velice rychle hledat - nejprve je nahrán příslušný soubor a pak je teprve provedeno vyhledání.

Ovládání programu je řešeno pomocí šipky a "oken" - což je ten nejvíce uživatelsky příjemný způsob, běžně používaný hlavně na 16-bitových počítačích. Při práci s databází máte k dispozici kvalitní diskový ovladač (registruje všechny provedené změny a upozorňuje na opomenutí soubor uložit, ptá se na přepsání starých souborů (vida, nezduvojuje je!) a při chybách nevypadává

jen tak do Basicu), spoustu editačních funkcí (přidávání, vsouvání, opravování, umisťování a vyhledávání slovíček), možnost seřídění dat podle definované abecedy, atd. Na distribučním disku najdete kromě ukázkových aplikací (anglicko-český slovník, knihovna) i znakové sady, předdefinované abecedy (česká, anglická, německá, ruská), utility pro vstup textu z databáze do Desktopu a instalační program pro vytváření vlastních databází.

Soubor	Slovičko	Písmeno	Test	Ovládání
Zpět	Ulož	Ulož	Ulož	Nahoru Dolů
acced	Ulož	1.	shoda, souhlas	2.
ve sho	anej			
popudu,	Zpětně	2.	out of ~ with" v	
dohoda	Další	3.	jednomyslně @ v 1.	
rozporu	souhlasí,	shodovat se,	"with" s) hodit se	
(k) 2.	dát,	poskytnout	3. vyhovět ("a	
request"	žádosti) *	"~ praise" vzdát		
chválu;	"he was ned a t."	dostalo se mu čeho		
-ance	[-ns] shoda;	"in ~ with" podle čeho, v		
souhlasu s	-ant [-nt]	"~ with" souhlasný s,		

Slovičko	Písmeno	Test	Ovládání
Ulož	Ulož	Ulož	Nahoru Dolů
Konec	Ulož	1.	shoda, souhlas
Ulož	anej		
Ulož	Zpětně	2.	out of ~ with" v
Ulož	Další	3.	jednomyslně @ v 1.
Ulož	souhlasí,	shodovat se,	"with" s) hodit se
Ulož	dát,	poskytnout	3. vyhovět ("a
Ulož	žádosti) *	"~ praise" vzdát	
Ulož	"he was ned a t."	dostalo se mu čeho	
Ulož	[-ns] shoda;	"in ~ with" podle čeho, v	
Ulož	souhlasu s	-ant [-nt]	"~ with" souhlasný s,

Zvukové generátory BEST a NICOL IS-2

Tuzemský výrobce, firma Nicol electronic (dříve Best) nabízí zvukový generátor Sound Interface IS-2 pro ZX Spectrum, Didaktik a Kompakt. Na rozdíl od předchozího typu od firmy Best má IS-2 průchozí konektor a spínač ON - OFF.

Na oboustranné desce s plošnými spoji jsou z jedné strany umístěny IO 8400, 74LS02 a z druhé AY - 3 - 8912. Pro výstup zvuku je použit stereofonní jack 3.5 mm pro připojení sluchátek.

Po elektrické stránce se jedná v podstatě o nezměněné zapojení zvukového generátoru Best, které kdysi podle polského Bajtku přetisknul časopis Fifo. Jak již bylo na stranách časopisu ZXM popsáno, toto zapojení má několik chyb, nehlídá všechny potřebné signály a tak bohužel některé programy, zejména hry, s ním nebudou spolupracovat (resp. hrát) tak, jak by měly (naštěstí jich není mnoho). Spínač ON - OFF zajišťuje v poloze OFF trvalé nulování zvukového generátoru. Domnívám se, že by bylo lépe, kdyby spíše plnil funkci INHIBIT, tj. zamezoval kolizi zvukového generátoru s jinými současně připojenými zařízeními, např. s jednotkou D-40, protože pokud nechci poslouchat hudbu, stačí poněkud vysunout konektor sluchátek a nepotřebuji k tomu spínač.

Mechanické provedení skřínky je vzhledově uspokojivé (všechny mezery, kudy by mohlo být vidět dovnitř, jsou pečlivě zakryty). Z hlediska případných oprav nebo úprav je to však řešení nevhodné, neboť není rozebíratelné bez mechanického poškození skřínky. Ta je slepena z plastických dílů, nenajdeme zde jediný šroubek nebo rozebíratelný spoj. V praxi to znamená, že pokud si majitel bude chtít opravit chyby v zapojení, nebo jej jakkoliv upravit, (např. propojení nezapojených vývodů na průchozím konektoru), bude muset nejdříve přemýšlet, jak se do skřínky dostat a pak vyrábět skřínku novou.

K hezkému vzhledu skřínky rozhodně nepřispívají chyby v anglických nápisech, např. "slučitelný" má být "compatible" a nikoliv "compatibile".

K návodu na obsluhu se bohužel nemoho vyjádřit, neboť u dodaného vzorku nebyl k dispozici.

Ve srovnání s Melodikem je u IS-2 pouze několik málo výhod: menší rozměry, kratší délka spojů v průchozím konektoru a tím i menší zatížení sběrnice počítače, vyvedení konektoru sluchátek do boku, kde nepřekáží dalšímu připojenému zařízení a pokus o nulování zvukového generátoru spínačem ON-OFF (když už, tak mělo být spíše použito tlačítko). V mnoha ohledech však zvukový interface Nicol IS-2 za Melodikem pokulhává: chyby v zapojení (nehlídá všechny potřebné signály), nemá reproduktor ani regulaci hlasitosti, neobsahuje krystalem řízený generátor, takže někomu IS-2 hraje o něco výše, jinému o něco níže a kytaru si podle něj přesně nenaladíte, plus řada dalších nepřijemných drobností. Melodiku i IS-2 bych vytknul, že nemají vyvedenu paralelní bránu a uživatel za ní platí, zatímco v případě použití IO AY - 3 8913 bez této brány by generátor vyšel jednodušší i levnější.

Melodik ani IS-2 nemají průchozí všechny vývody konektoru, neobsahují spínač INHIBIT (vyřazení z funkce) obvyklý u kvalitních zahraničních periférií (např. Disciple) a tříkanálový zvuk degradují na dvoukanálový stereofonní (měly by tu být alespoň spínače pro jednotlivé kanály dovolující zjistit, co každý kanál hraje). U Melodiku je možné tyto nedostatky odstranit, u IS-2 také, ovšem za cenu mechanického poškození skřínky.

Věřím, že tyto poznámky výrobci nikterak neublíží, protože mezitím uběhnul rok, mnoha uživatelům již byl IS-2 prodán a po rozdělení ČSFR na dva státy již není konkurence (Melodik) tak velká. Přesto bych firmě Nicol poradil, aby u dalších verzí svých generátorů odstranila chyby a vybavila je jednoduchými doplňky (konektor MIDI-OUT, RS-232), které stojí pár korun a výrazně zlepšují prodejnost výrobku před konkurenčními (a to i zahraničními). Případným zájemcům jsem ochoten uvedené chyby odstranit.

- rex -



ROZŠÍŘENÁ GRAFIKA NA ZX SPECTRU

módy 512*192 a 256*384 aneb poklep na čelo se nekoná

Jistě zarážející titulek, že? Článků na téma grafika ZX Spectra bylo napsáno hodně. V BITu, Fifu, Excaliburu a jiných. Vzpomeňme si na naše pražské exspectristy popírající větší rozlišení na ZXS a o člověku, jenž vám řekne, že viděl program, který ukazuje obrázky v grafice 512*192 si uděláte úsudek jedním slovem: "BLÁZEN". Bohužel však s Vámi nemohu souhlasit. Rozšířená grafika na ZX Spectru jde udělat! Ve své podstatě celý systém zobrazování a tvorby této grafiky vychází z amigistického Interlace a dá se tato rozšířená grafika nazvat jako "zmenšený obrázek dvakrát (vždy z jedné ze dvou os) aniž by se z obrazovky ztratil jeden jediný bod". Tak toto byla přesná definice celého systému. Nyní si řekněme jak vše funguje.

Vezmeme obrázek ze hry Venom strikes back a provedeme s ním toto.

- 1 Vymažeme sudé sloupce, zbydou liché
- 2 Liché sloupce "stáhneme" k sobě.
- 3 Uložme do 2. Vram.
- 4 Opět si vyvoláme obrázek.
- 5 Vymažeme liché sloupce. Zbydou sudé
- 6 Sudé sloupce "stáhneme" k sobě.
- 7 Uložme do 1. Vram
- 8 Obě Vram střídáme pod frekvencí 25 Hz. Tímto programem:

1 Pause 1:out 32765,24:pause 1:out
32765,16:goto 1

anebo ze strojového kódu:

```

Ei
ld bc,32765
zzx hlt
ld a,24
out (c),a
hlt
ld a,16
out (c),a
jp zxx

```

Ještě jednou si zrekapitulujme, co jsme s obrázkem udělali

Vymazali jsme všechny sudé sloupce a zbylé (liché) jsme stáhli k sobě. Tím nám vzniknul dvakrát zmenšený obrázek. Jenže se v něm nevyskytují žádné sudé sloupce, takže se jedná o úplný hybrid.

Udělejme ještě jednou totéž, ale obráceně. Vymažeme všechny liché a zbylé (sudé) stáhneme k sobě. Vznikne nám opět dvakrát zmenšený obrázek, který obsahuje tentokrát všechny sudé sloupce.

Vznikly nám tak dva obrázky. Každý je dvakrát menší než byl originál a každý obsahuje sloupce, které naopak ten druhý

neobsahuje. Využijme očního klamu a to tak, že tyto dva výsledné obrázky budeme rychle za sebou přepínat. Co se stane? Na kvalitní obrazovce získáváme poměrně velmi dobrý obraz onoho obrázku, avšak s tím rozdílem, že bude dvakrát zmenšen. A co body? Jsou tam všechny? Ano. Všechny sudé sloupce a všechny liché sloupce jsme přece zachovali a nic jiného jsme s obrázkem nedělali, takže opravdu se jedná o zmenšený obrázek aniž by se z obrazovky ztratil jeden jediný bod. Ještě na jeden důležitý detail jsem zapomněl. Řekl jsem, že rychlým střídáním těchto obrazovek získáme tzv. optický klam a výsledný efekt se našim očím zdá (na kvalitních televizorech) statickým. Střídání těchto dvou obrázků však musí být synchronizováno (nebudu zde složitě vysvětlovat proč) pod 25 Hz, tak jako je klasické zobrazování televizoru synchronizováno na 50 Hz. Proč zrovna 25 Hz? Jinak to ani nejde. Střídáme dva obrázky a chceme-li dosáhnout kvalitního výsledného efektu, musíme jejich zobrazování synchronizovat s TV paprskem. To v praxi znamená, že každý obrázek musí být vždy stejnou dobu na obrazovce. Takže vše výše napsané platilo pro sloupce. Získali jsme obrázek zmenšený v ose X dvakrát. Jedná se tedy o pseudografiku 512*192. A jak to dopadne v ose Y? Úplně stejně. Budeme zde ale na rozdíl od sloupců konvertovat řádky a ne sloupce. Získáme tak pseudografiku 256*384.

Ještě jednou upozorňuji, že je nutné mít kvalitní televizor. Ne nějaký ruský krám. Program, který konvertuje obrázky a umožňuje jejich zobrazování skutečně již existuje a jmenuje se INT EXE 2.128.

Ještě je nutné upozornit, že program INT EXE 2.128 je určen jen pro 128, protože jak asi víte ZX Spectrum 128 má k dispozici dvě Videoram, které přepínáte prakticky jen jednou instrukcí o délce 11 nebo 12 taktů (OUT 0/32,24/16, nebo OUT 32765,24/16) což znamená, že obrázek nahraný do 2 Vram "eldírneme jakoby na obrazovku" jedním outíkem. Na 48 se nedá plnohodnotně tohoto módu využít. To by snad bylo k tomuto asi tak všechno.

Doufám, že jsem Vás tímto článkem konečně přesvědčil o tom, že to *staré dobré Spectrum* ještě něco dokáže. Celý systém funguje a byl otestován na několika typech televizorů a monitorů. Nejstatičtější obraz dosáhnete na monitoru se vstupem RGB, avšak ani televizor Tesla color 419 (který vlastním) či malý televizor Satelit nemají vůbec špatný obraz.

Ti zkušenější si jistě výše uvedené naprogramují a zjistí, že to opravdu funguje. Ti kteří nemají takové zkušenosti se strojovým kódem mohou podrobně informace získat u autora - dopis zašlete na adresu redakce.

Poklep na čelo se skutečně tentokrát nekoná.

- JSH -

Samcon '93 - SAM není mrtev!

Počátkem dubna tohoto roku se v základní škole v Kodaňské ulici v Praze konalo první mezinárodní setkání uživatelů počítače SAM Coupé - SAMCON '93. Celá akce se ke spokojenosti všech účastníků i organizátorů vydařila více než dobře.

Na SAMCONu nebyli jen majitelé SAMů, ale i spectristé (např. MATA, QJETA a MUF a Brna), exspectristé (František Fuka, Patrik Rak, Patrik dokonce přislíbil pomoc při tvorbě nového softwaru na SAMa), a dokonce i několik amigistů (např. FRACTALUS - Ondřej Kárný). Zaslouhou hosta z Polska (M. Szafranski, vydavatel polského diskového magazínu SAM PAPER) se návštěvníci seznámili s novinkami jako např. "střílečka" PARALAX a hudební program PRO-TRACKER. Bylo zde předvedeno připojení syntetizéru k SAMovi přes MIDI, spojení několika SAMů do sítě, konaly se přednášky na různá témata

(přenos dat z PC do SAMa, SAM v účetnictví a další).

Přestože v červnu minulého roku firma SAM Computers zkrachovala, SAMové se již opět vyrábějí a budou k dostání i v Čechách (SAM 512 KB s drivem 3,5" za cca 9000 korun). SAMCON se tedy přeže všechny obavy opravdu podařil a pro velký úspěch se bude přístí SAMCON konat už na podzim 1993 a zřejmě tentokrát bude (více či méně - podle případného zájmu spectristů) zaměřen i na ZX Spectrum

Jan Hanousek

Datum konání přístího SAMCONu je zatím stanoveno na 8. až 10. října. Bližší informace se dozvíte na adrese: KAPSA, Jiří Vávra, Ruská 24, 100 00 Praha 10; nebo: Jan Hanousek, Lázeňská 6, 118 00 Praha 1.



Isem majiteľom disketovej jednotky Didaktik 40 a zajímalo by mne,jakým spôsobom jsou ve snapu uloženy jednotlivé registry a jak bych mohl ze snapu získat jejich hodnoty.

J. Š., Bohušovice

Hodnoty registrů v okamžiku stisknutí tlačítka SNAP jsou uloženy v prvních 128 bytech souboru. Přesně jsou to pro jednotlivé registry tyto adresy:

16360	+104	přerušení (2. bit)
16361	+105	I
16362	+106	IY
16364	+108	IX
16366	+110	HL'
16368	+112	DE'
16370	+114	BC'
16372	+116	AF'
16374	+118	HL
16376	+120	DE
16378	+122	BC
16380	+124	AF
16382	+126	SP
16384	+128	PC (IP)

K tabulce ještě malé vysvětlení, první číslo je adresa, kde se hodnota vyskytuje, když je soubor nahráván M-DOsem, druhé číslo je relativní adresa vztažená k počátku souboru a poslední je vypsán registr, jehož hodnota toto číslo je. Na adrese +105 je zapsána hodnota registru I, na adrese +104 je ve druhém bitu informace o tom, jestli bylo přerušeno povoleno (1), nebo zakázáno (0). Při nahrávání SNAPu program ještě musí zjistit, jaký byl mód přerušování - to se nedá zjistit přesně, a tak si musí vypomáhat tím, že když je v I hodnota 63, předpokládá IM 1, když je tam něco jiného, předpokládá IM 2 (IM 0 se na Spectru nepoužívá) - tento způsob zjišťování může samozřejmě selhat a velice často je příčinou nepochopitelného pádání SNAPů.

Nyní jak můžete tyto hodnoty zjistit:

1) Nahrávejte si první (nulový) sektor SNAPu pomocí příkazu READ "*"jméno", 0, 40000. Na adresách 40104 - 40129 jsou požadované hodnoty.

2) Použijte TOOLS 40/80 a jeho funkci edituj soubor (E).

3) Použijte program CRACKSHOT - ten Vám nejen ukáže hodnoty registrů a stav přerušování, ale umožní i spoustu dalších věcí.

..... (1) Jakým způsobem se dá sestavit program ŠIPKA, aby při delším programu neblíkal?

(2) Proč se používají masky spritové grafiky a ne jenom předlohy?

(3) Jaké bude zaměření knihy Assembler a ZX Spectrum 3 a zda je to poslední díl této řady?

Libor M., Uherský Brod

1) Abych se přiznal, tak otázky, jako je ta první, mě přivádějí na pokraj šílenství - asi proto, že na takovou otázku se prostě nedá odpovědět. V otázce totiž chybí přesnější popis, co míníte výrazem "delší program", Vám to možná jasné je, mě ovšem nikoliv. Nicméně podívám se do textu kapitoly a pokusím se na to přijít. Pravděpodobně Vám jde o to, co je zmíněno na straně 33 v prvním odstavci - že do programu nemůžete vkládat cokoliv a kamkoliv. Pokud potřebujete při kreslení šipky provádět ještě nějaké další a hlavně (časově) delší akce, můžete je vložit před instrukci HALT. Tady si musíte dát pozor pouze na to, abyste neměnil tu část obrazovky, kde se zrovna šipka nachází.

2) Myslím si sice, že jsem to v knize vysvětlil ale zřejmě ne dost jasně. Použití masky je nutné proto, že když chcete rychle kreslit do obrazovky, musíte měnit najednou hned 8 bodů (jeden byte - to je dáno hardwarem a s tím nic nenaděláte). Stává se ale, že nechcete měnit všech 8 bodů, ale jenom některé z nich - nejjednodušší cesta je pomocí logických bitových operací ponechat to, co měnit nechcete. A informace o tom, co měnit nechcete, obsahuje maska.

3) Obsahové zaměření třetího dílu zatím neznám ani já, pravděpodobně bych se mohl zmínit o ovládání tiskárny (komunikace), o využití interpretu basicu, který je v ROM (výpočty s reálnými čísly), mohlo by tam být něco o datových strukturách a práci s nimi, možná i nějaký rozsáhlejší příklad programu, také něco o uživatelském prostředí, třeba roletová menu, také něco o

práci s disketovou jednotkou, něco, co by navázalo na připravovanou publikaci o D40/D80 od George K. a moje příspěvky do ZXM. Třetí díl bude s největší pravděpodobností také dílem posledním.

Vážená firma,
hromadným rozšířením Vašich pakovacích programů, ale i mnohých dalších si čoraz větší počet majitelů počítačů + kazetákov každý "druhý" program komprimuje. To je samozřejmě ale trochu nevhodné pro disketovou jednotku - ušetří se síce místo, ale dlouho se čeká na rozpakování. No a v tom spočívá můj problém. Preto by som sa od Vás rád dozvedel, ako si môžem zistiť v programe /po rozpakovaní/ adresu odkiaľ sa uložil a dĺžku bloku.

Peter B., Nitra

Nejprve bych Vám chtěl trochu oponovat, nemyslím si, že by dekomprese programu po nahrání trvala tak dlouho, aby Vám to muselo vadit - u našich programů (Mr. Pack) trvá nejvýše 10 sekund. Pokud tedy máte na mysli třeba hry, tak jediné použití, při kterém by komprese mohla vadit a které mě napadá, jsou závody kdo nahraje a potom vyřetuje více her za zvolený časový interval, jinak mi to nepřipadá tak tragické. Navíc nesmíte zapomínat, že i nahrávání z diskety trvá nějaký čas, zvláště když se jedná o více souborů. Poněkud jiná situace může nastat, pokud používáte komprimované uživatelské programy, a musíte je často střídát, tam může každá sekunda navíc vadit.

Podle našich zkušeností (z prodeje) kompresní programy používají i majitelé disketových jednotek, dokonce ve stejné míře jako "kazetoví" uživatelé, možná dokonce víc. Další výhodou komprimace pomocí kompletu (Mr.PACK + PRESSOR 6) je i jednodušost převedení programu z kazetové verze na disketovou (stačí použít program PACK->DISK). Tím bychom měli probrané filozofické aspekty daného problému, a teď se můžeme podívat na ty "technické".

Jestli dobře chápu otázku, chcete vědět, jak zjistit u rozkomprimovaného programu počáteční adresu a délku, abyste ho mohl uložit na disk v rozkomprimovaném stavu. Tak to je problém, na který se nedá přesně odpovědět jednoduše. Pokusím se tedy odpovědět alespoň přibližně.

Nejprve konkrétně pro programy, které jsou komprimovány pomocí našich pakovacích programů (Mr. PACK nebo starší Packmaker). Programy jsou většinou komprimované od nějaké adresy (standardně 24200) až do konce paměti (adresy 65535), prázdná místa se nevynechávají - kompresí se totiž zkrátí na několik bytů. Počáteční adresa je stejná jako adresa, ze které se spouští dekomprese. V této oblasti se tedy musí vyskytovat celý program - bude to všechno, co je vyplněno něčím jiným, než nulami. Pokud tedy program uložíte na disk příkazem SAVE "*"jméno" CODE 24200, 41336, nic nezkažte.

Kdybyste chtěl uložit jenom to, co je nezbytně nutné, prohlédněte si celou paměť nějakým monitorem (např. DEVASTACE) a

zjistíte, kde se nalézají nuly, takové bloky vynechte a uložte jenom to, co je potřeba - bloků pak může být i několik.

Protože tento postup je poněkud náročný na práci (musíte hledat části programu), použijte kompromis. Proveďte dekompresi programu a pak jej znovu zkomprimujte pomocí programu Mr. PACK, nyní ale s vypnutým Huffmanovým kódováním (v menu **Option** musíte nastavit **huffman no**). Jednoduchá komprimace, která bude použita, pouze zkrátí oblasti s opakujícími se byty - příslušná dekomprese netrvá déle než sekundu a neměla by Vás zdržovat.

Pokud se setkáte s jinými kompresními programy, zkuste vzít jako počátek bloku adresu, odkud se spouští dekompresní rutina nebo případně hodnotu systémové proměnné **RAMTOP** zvětšenou o jedničku (**RAMTOP** se nastavuje příkazem **CLEAR číslo**). Zjištění délky je stejné.

Pokud chcete zjistit délku opravdu přesně, nastavte **RAMTOP** na co nejmenší hodnotu (23999), vyplňte paměť od 24000 do konce nějakým číslem (ne nulou), pak nahrajte komprimovaný blok (tam, kam se nahrává loaderem) a proveďte dekompresi. Potom se podívejte, která část paměti se

změnila. To je nejobecnější způsob, jak to zjistit.

Když budete provádět úpravu programu, dejte si pozor, abyste z fungující komprimované verze nevyrobil nefungující verzi nekomprimovanou, byť by se rychleji nahrála, nebude k ničemu.



BOMBARDÉR

Následující program je jednoduchá hra napsaná v BASICu. Autorem je známý George K., podívejte se, jak před několika lety začínal.

Podtržená písmena A, B, C, D a E jsou odpovídající grafické znaky (zadávají se stiskem klávesy s tímto písmenem v grafickém módu):

```

50 LET HSCORE=0: GOTO 7000
90 INK 7: PAPER 0: BORDER 1:
CLS
98 GO SUB 4000
99 GO SUB 2000
100 GO SUB 700
106 PRINT AT 0,0;"SCORE:";score;
AT 0,13;"FUEL:";fuel
110 FOR u=2 TO 19
120 FOR v=0 TO 31
125 PRINT AT u,v;"ABCD";
160 IF INKEY$="p" OR INKEY$
="P" THEN GO SUB 800
170 IF INKEY$="o" OR INKEY$
="O" THEN GO SUB 900
180 IF v=29 THEN GO TO 185
182 GO TO 190
185 IF SCREEN$(u+1,1)="#" THEN
GO TO 3000
190 IF v=30 THEN GO TO 195
192 GO TO 200
195 IF SCREEN$(u+1,2)="#" THEN
GO TO 3000
200 IF v=31 THEN GO TO 205
202 GO TO 210
205 IF SCREEN$(u+1,3)="#" THEN
GO TO 3000
210 IF SCREEN$(u,v+4)="#" THEN
GO TO 3000
220 FOR r=1 TO 4
225 IF INKEY$="o" OR INKEY$
="O" THEN GO SUB 900
226 PRINT AT u,v;"ABCD";
230 IF INKEY$="p" OR INKEY$
="P" THEN GO SUB 800
240 NEXT r
280 PRINT AT u,v;" ";
290 NEXT v
300 NEXT u
310 PRINT AT 19,0;*****
312 PRINT AT 19,28;"ABCD"
314 PRINT AT 10,6;"PRISTANI SE
PODARILO"
316 BEEP .1,10: BEEP .1,14: BEEP
.1,18: BEEP .1,22: BEEP
.1,26
318 LET zx=zx-1
320 IF zx=3 THEN GO TO 6000
322 PAUSE 30: CLS
324 LET fuel=fuel+4: LET S=10
330 GO TO 100
700 FOR a=1 TO 30

```

```

702 INK RND *4+1
705 LET c= RND *zx+9
710 FOR b=19 TO c STEP -1
715 PRINT AT b,a;"#"
720 NEXT b
725 BEEP .001,a+20
730 NEXT a
735 INK 7
740 RETURN
800 INK 7
802 IF fuel=0 THEN RETURN
805 FOR d=1 TO 10
810 PRINT "-";
815 BEEP .001,d+20
840 NEXT d
850 LET fuel=fuel-1
852 PRINT AT 0,18;"0":fuel
855 RETURN
900 LET v1=v
901 IF u=10 THEN GO TO 925
902 IF u=12 THEN GO TO 926
903 IF u=15 THEN GO TO 927
904 IF u=18 THEN GO TO 928
905 FOR j=u+1 TO s+u
907 IF v=31 THEN LET v=0: LET
u=u+1: PRINT AT u-1,v;"
(32 mezer)
": IF u=19 THEN GO TO 3000
908 PRINT AT u,v;" "; AT
u,v+1;"ABCD"
909 LET v=v+1
910 PRINT AT j,v1;"E"
911 IF SCREEN$(j+1,v1)="#" THEN
LET score=score+10: PRINT AT
0,6;score
915 BEEP .001,j+30
917 FOR w=1 TO 8: NEXT w
920 PRINT AT j,v1;" "
923 NEXT j
924 RETURN
925 LET s=8: GO TO 905
926 LET s=6: GO TO 905
927 LET s=4: GO TO 905
928 LET s=2: GO TO 905
929 LET s=1: GO TO 905
995 RETURN
2000 LET u=2: LET v=0
2001 LET zx=10: LET s=10
2002 LET fuel=10: LET score=0
2090 RETURN
3000 PRINT AT 5,11; FLASH 1;"GAME
OVER"
3002 BEEP .1,10: BEEP .1,11: BEEP
.1,12: BEEP .1,14: BEEP
.1,13: BEEP .1,11: BEEP
.1,9: BEEP .1,0: BEEP .3,-5
3006 IF score>hscore THEN LET
hscore=score
3008 FOR f=1 TO 150: NEXT f
3010 GO TO 7000
3900 STOP
4000 POKE USR "e"+0, BIN 00000000
4001 POKE USR "e"+1, BIN 01000100
4002 POKE USR "e"+2, BIN 01101100
4003 POKE USR "e"+3, BIN 00111000
4004 POKE USR "e"+4, BIN 01111100
4005 POKE USR "e"+5, BIN 01111100
4006 POKE USR "e"+6, BIN 00111000
4007 POKE USR "e"+7, BIN 00010000
4008 POKE USR "a"+0, BIN 00111000
4009 POKE USR "a"+1, BIN 01111100

```

```

4010 POKE USR "a"+2, BIN 00111110
4011 POKE USR "a"+3, BIN 00111111
4012 POKE USR "a"+4, BIN 00011111
4013 POKE USR "a"+5, BIN 00011111
4014 POKE USR "a"+6, BIN 00001111
4015 POKE USR "a"+7, BIN 00000111
4016 POKE USR "b"+0, BIN 0
4017 POKE USR "b"+1, BIN 0
4018 POKE USR "b"+2, BIN 0
4019 POKE USR "b"+3, BIN 11111111
4020 POKE USR "b"+4, BIN 11111111
4021 POKE USR "b"+5, BIN 0
4022 POKE USR "b"+6, BIN 11111111
4023 POKE USR "b"+7, BIN 11111111
4024 POKE USR "c"+0, BIN 0
4025 POKE USR "c"+1, BIN 0
4026 POKE USR "c"+2, BIN 0
4027 POKE USR "c"+3, BIN 11111111
4028 POKE USR "c"+4, BIN 11111000
4029 POKE USR "c"+5, BIN 00111000
4030 POKE USR "c"+6, BIN 11111111
4031 POKE USR "c"+7, BIN 11111111
4032 POKE USR "d"+0, BIN 0
4033 POKE USR "d"+1, BIN 0
4034 POKE USR "d"+2, BIN 0
4035 POKE USR "d"+3, BIN 10000000
4036 POKE USR "d"+4, BIN 01000000
4037 POKE USR "d"+5, BIN 00100000
4038 POKE USR "d"+6, BIN 11111100
4039 POKE USR "d"+7, BIN 11100000
5000 RETURN
6000 IF score>hscore THEN LET
hscore=score
6010 PRINT AT 10,9;*** BRAVO
****
6020 BEEP .1,12: BEEP .1,14: BEEP
.1,16: BEEP .1,18: BEEP
.1,20: BEEP .1,22: BEEP
.1,24
6030 BEEP .1,12: BEEP .1,14: BEEP
.1,16: BEEP .1,18: BEEP
.1,20: BEEP .1,22: BEEP
.1,24
7000 REM
7001 PAPER 0: BORDER 1: CLS : INK
2
7002 PRINT AT 4,0; INVERSE
1,*****
7004 PAPER 2: INK 7
7006 PRINT AT 5,0;"
BOMBARDER "
7010 PAPER 0: PRINT AT 8,8;"High
Score:";hscore
7020 PRINT AT 10,11;"OVLADANI
-----"
7030 PRINT AT 12,11;"BOMBY..O
PAL....P"
7031 PRINT AT 16,0; INVERSE
1,*****
7032 INK 7: PAPER 2: PRINT AT
17,0;" " 1987 George
K. "
7034 PRINT #0;" " PRESS 'ENTER'
TO START "
7040 IF CODE INKEY$ =13 THEN GO
TO 90
7050 GO TO 7040

```

ZX magazín a robotika (6)

LOG 1, LOG 0 A CO JE MEZI TÍM?

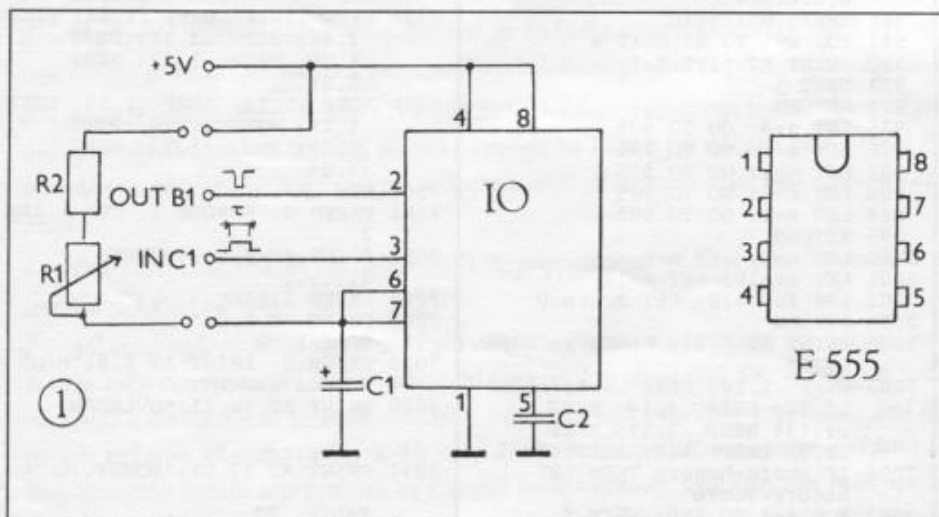
Tento díl našeho seriálu je posledním pokračováním věnovaným základním zapojením a řešením připojení ovládaných a snímaných periférií.

Doposud jsme se ale věnovali jen řízení a snímání signálů pouze pomocí dvou logických stavů, tedy nul a jedniček. V robotice však nalézáme mnoho aplikací, kde je potřeba plynule sladovat nějakou veličinu a nebo naopak něco spojitě ovládat. Takovýto signál se nazývá analogový a pro počítač, znající maximálně jedničky a nuly je nezpracovatelný. Abychom takové aplikace mohly připravit, je třeba použít tzv. analogově-digitální nebo digitálně-analogové převodníky (značí se A/D, resp. D/A). Jejich stavba je však poměrně náročná a těžko bychom ji mohli na tomto místě jednoduše popsat.

Nic však není ztraceno, stavbu A/D a D/A převodníků jednoduše obejdeme použitím jednodušších obvodů a jednoduchými programy, které budou pro naše potřeby vyhovovat.

"Převodník A/D" s použitím 555

Nápad na stavbu tohoto převodníku jsme si opět vypůjčili z časopisu Amatérské radio (č.6/88, str.220-221). Tento jednoduchý převodník využívá v zapojení IO časovače, známého pod typovým označením E555. Snímacím členem v zapojení je proměnný odpor, tedy trimr (ale také termistor nebo fotorezistor). Na velikosti tohoto odporu a velikosti kondenzátoru C1 v zapojení na obr.1 pak závisí délka impulsu, který E555 vygeneruje. Tento impuls je pak možno snímat na sběrnici počítače. Další zpracování impulsu je již jen pouze otázkou software, tedy jednoduchého programu ve strojovém kódu, který měří délku impulsu od jeho náběžné hrany.



Obr. 1. Schéma převodníku A/D: IO1=E555, C1=cca 300 až 700nF (provedení tantal), C2=47 nF, R1 je snímací proměnný odpor do 1M, R2=1k.

Obslužný program je stejně prostý jako zapojení a jistě jej rychle pochopíte a snadno sami upravíte pro registraci signálu na jiném portu než C.

```
LD A,154          ;nastavení režimu 8255
OUT A,127
-----
START:  DI
        LD BC,00   ;vynulování počítadla
        LD A,0
        OUT (63),A
        LD A,1     ;startovací impuls
        OUT (63),A
COUNTER: INC BC    ;zvýšování počítadla
         IN A,95   ;čtení pulzu
         RRA
         JR C,COUNTER ;další zvýšení počítadla
         ;pokud impuls neskončil
         EI
         RET
```

Hodnotu počítadla, které je realizováno dvojregistrem BC, zjistíme při spuštění programu příkazem LET HODNOTA=USR START, nebo složitěji RANDOMIZE USR START, kdy se hodnota dvojregistru BC uloží na adresu 23670. Odtud ji potom můžeme zjistit příkazem LET HODNOTA = PEEK 23670. (Pro méně znalé: Program ve strojovém kódu se spouští funkcí USR adresa. Hodnotu funkce pak musíme buď využít, nebo někam odložit. Právě k odložení slouží příkaz RANDOMIZE, který hodnotu funkce ukládá na adresu systémové proměnné SEED, z níž využijeme pouze nižší byte). Závislost mezi hodnotou v registru BC a měřenou veličinou je třeba zjistit experimentálně.

Nyní se blíže seznámíme s funkcí našeho zapojení. Impuls začne program snímat po vynulování obvodu E555, signálem z PB1 počítače. Doba impulsu, kterou budeme měřit, je přímo úměrná hodnotě snímaného odporu a kondenzátoru, podle vztahu $t = 1,1 \cdot R \cdot C1$. V případě, že bude doba impulsu t delší než 0,6 s, dojde k "přetečení" počítadla a znovučení snímaného impulsu. Tímto bychom však získali nesprávný výsledek, a proto musíme v závislosti na měřeném odporu vhodně měnit hodnotu kondenzátoru C1.

Pokud budete chtít toto zapojení nějak více rozvinout, bližší informace můžete čerpat v již zmíněném článku v AR.

Fázové řízení výstupu

Analogové řízení bude mnohem jednodušší než čtení analogových veličin, protože pro připojení ovládaného zařízení použijeme pouze v předchozích pokračováních seriálu popsané zesilovací obvody výstupu, tedy zapojení zesilovače s otevřeným kolektorem (ZXM č.4/92) nebo dvoustupňový zesilovač (ZXM č.5/92). Výstupní obvod zvolíme podle připojené zátěže!

Analogové ovládání bude v našem případě realizováno vlastně oklikou, čistě softwarovou cestou. Využijeme k tomu tzv. metodu

fázového řízení, což ve skutečnosti neznamená nic jiného, než ovládání výstupního zařízení prostou změnou poměru doby zapnutí a vypnutí obvodu při konstantní frekvenci. Schema fázového řízení je nakresleno na obr.2. V praxi to znamená, že např. dioda nebo žárovka budou v závislosti na době sepnutí svítit méně nebo více (čím déle je výstup zapnut než vypnut, tím více se žárovka rozsvítí). Frekvence bude relativně vysoká, takže nebudete vnímat blikání, ale různý jas.

```

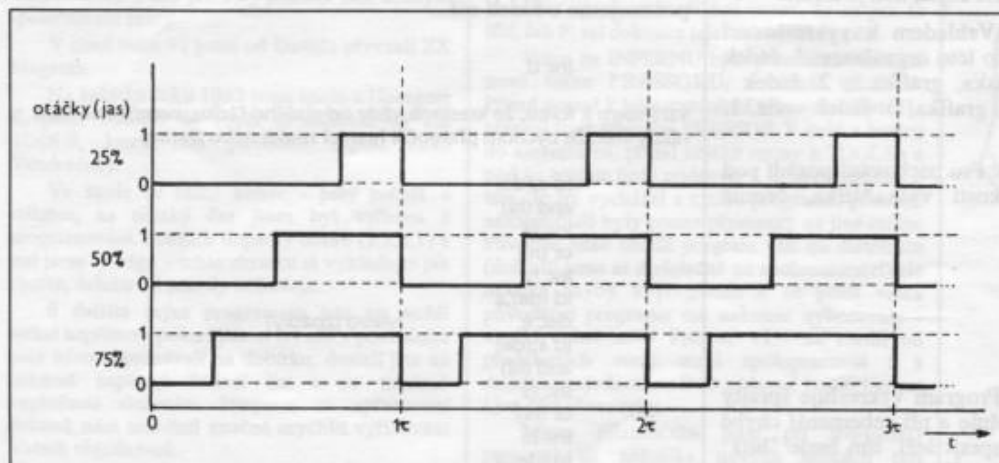
START:  LD HL,DELKA
        LD A,(23670)
        LD D,A      ;doba zapnutí do D
        CPL
        INC A
        LD E,A     ;doba vypnutí do E
OPAKUJ: INC D      ;je-li D nulové,
        DEC D     ;pak se sice časuje,
        JR Z,PRESKOC ;ale nic se nevysílá
        LD A,01  ;na výstup log.1
        OUT (65),A
PRESKOC: LD B,D
CASUJ1:  DJNZ CASUJ1 ;časuje zapnutí
        LD A,00  ;na výstup log.0
        OUT (65),A
        LD B,E
CASUJ2:  DJNZ CASUJ2
        DEC HL
        LD A,H
        OR A,L
        JRNZ OPAKUJ ;opakuje zap/vyp cyklus
        RET
        DEFW DELKA ;délka trvání signálu

```

Tento program pro fázové řízení se spustí příkazem RANDOMIZEUSR START, poté co příkazem RANDOMIZEKOLIK uložíme do proměnné SEED velikost poměru vypnuté a zapnuté fáze v rozsahu 0 až 255 (0 až 99 % zapnutí). Hodnoty proměnné DELKA je třeba vyzkoušet experimentálně.

Zatímco u žárovky se fázové řízení projevovalo proměnlivým jasnem, u motoru se projeví nižšími otáčkami. Je to proto, že motor při rychlém zapínání a vypínání nedosáhne maximálního počtu otáček. Otáčky však budou přímo úměrné poměru zapnuto / vypnuto. Protože však motor při zapnutí vždy dostane plné napětí, bude mít vždy dostatečný točivý moment. Tohoto jevu lze využít např. při ovládání modelové železnice.

Při softwarové obsluze těchto aplikací je nutné použít strojový kód. Pro přeložení programu ve strojovém kódu nemusíte používat



Obr.2: Schema fázového řízení.

Videodigitizér

pro ZX Spectrum / Didaktik

Software + hardware umožňující sejmout obraz z televizního přijímače, kamery, videa, přehrávače nebo jiného počítače a jeho přenos (jako SCREEN\$) do počítače ZX Spectrum / Didaktik.

- snímá obraz s rozlišením 192 x 256 bodů ve zlomku sekundy. Dochází také k určitému vystínování obrazu a rozlišení stupně šedi
- animace uložených obrázků s nastavitelnou rychlostí simuluje videorekorder
- možnost tisku obrázků na připojené tiskárně, záznam na vnější magnetické médium (MGF pásek, disketa)
- programovatelné nastavení výřezu obrazu (okénka) s možností pozdějšího zvětšení detailu obrazu (zoom)
- sw i hw pracuje současně s D40/D80, Didaktikem Gama, M, Kompakt
- neomezené možnosti experimentování: počítačové portréty na ZXS, rozpoznávání písma, obrázků a předmětů, měření vzdáleností a ploch, ověřování pravosti rukopisu, fax, videotelefon, pomaloběžný videorekorder, atd.

Nepotřebujete ani kameru, ani video, filmový obrázek si můžete sejmout do ZXS přímo z tel. přijímače a využít pro tvorbu vlastních programů či her. Přenos obrázků z Atari, PC nebo Amigy do Vašeho ZXS, zmrazení a uchování důležitých obrázků z televizního vysílání s možností je později zobrazit, vytisknout, atd. Věnujte své dříve (rodičům, přátelům) svůj počítačový portrét pořízený na Vašem počítači a tiskárně Vaším videodigitizérem.

Objednávky za zpětnou řádně vyplněnou a ofrankovanou obálku na adrese:

Jan Drexler, Jahodová 2889, 106 00 Praha 10

hned rozsáhlé programové balíky typu Prométheus, pro tyto účely zcela postačí jednoduchý a elegantní assembler, nabízený firmou Proxima na disketách Public Domain.

Schotek

(C) Studio 22 MCMXCIII

Příště: R.U.R., aneb nechte práci strojům; povídání o základních principech ovládání robota.

Připomínky a náměty k tomuto seriálu (ne dotazy!) zasílejte na adresu:

SINCLAIR CLUB, P.O.Box
132, OSTROV, 363 01

Pokud Vás náš seriál zaujal a nemáte možnost sehnat starší čísla ZXM, obraťte se rovněž na naši adresu. Rádi Vám předchozí díly poskytneme. Nezapomeňte však do obálky vložit 20 Kč bankovku.

George K.'s

AnimAce 2

Díl třetí: Legenda pokračuje

Nebudeme zbytečně protahovat úvod a vrhneme se rovnou k věci... zkusili jste si vytvořit rychlou sprajtovací rutinu podle naznačených parametrů...? - hlavně mi netvrďte, že ano, stejně bych Vám to nevěřil. Já jsem to udělal a výsledek vidíte o několik řádků níž. Abychom se pohli z místa, vzal jsem toho trochu víc najednou, ale však Vy to určitě zvládnete.

Nejprve si vytvoříme tabulku všech sprajtů, do které uložíme jednak jejich celkový počet a potom základní údaje o každém z nich: příznakový bajt pro sprajtovací rutiny (já používám příznaky: 0 - sprajt se nekreslí, 1 - odstraň sprajt, 2 - polož nový sprajt, 3 - kreslí sprajt), rezervní bajt, ve kterém sice zatím není nic, ale může se časem hodit, a adresa dat sprajtu. Do tabulky umístíte všechny sprajty, které chcete (např. v levelu) hry používat - ty, které se mají objevit až v určitou chvíli jednoduše označíte příznakem 0.

```
QTABLE defb 2      ;počet sprajtů v tabulce
        defb 3,0    ;kreslí sprajt, druhý bajt nevyužit
        defw SPT01 ;adresa dat prvního sprajtu
        defb 0,0    ;sprajt se nekreslí, nevyužito
        defw SPT02 ;adresa dat druhého sprajtu
```

Data sprajtu uspořádáme následovně: trojnásobek výšky, polovina výšky, adresa ukazující do tabulky adres VRAM, adresa grafických dat sprajtu, adresa volného místa pro uchování pozadí. Jednotlivé položky si trochu objasníme:

- výška (hh) je násobena třemi (to je šířka), protože výsledkem je počet bajtů grafiky sprajtu, což potřebuje rutina pro uschování pozadí. Chcete-li sprajty s jinou šířkou než 3, musíte si uzpůsobit rutiny; chcete-li sprajty různých šířek, doporučuji využít rezervní bajt v QTABLE a uchovávat si v něm šířku a podle ní rutiny dále větvit.

- výška (hh) je dělena dvěma, protože to je počet smyček, který musí proběhnout ve vykreslovací rutině. Výška sprajtu musí být sudá.

- ukazatel do tabulky adres VRAM. Tabulku vytvoříme podprogramem MKTABLE z minulého dílu. Bohužel se mi do něj vloudila chybička - opravte si: LD BC, HT*128+WT a TABLE defs HT*WT. Nejsou to sice destruktivní chyby, ale tabulka by Vám zbytečně zabírala dvojnásobný počet bajtů, než je nutné.

- ukazatel na grafická data (nn). Vzhledem k vykreslovací rutině je potřeba upravit grafická data do této organizace: 1. řádek - zleva doprava - maska, grafika, maska, grafika...; 2. řádek - zprava doleva - maska, grafika, maska, grafika; 3. řádek - viz. 1. řádek, 4. řádek viz. 2. řádek, atd.

- ukazatel do volného prostoru (mm). Pro uschování pozadí pod sprajtem je zapotřebí místo o velikosti výška*šířka, čemuž odpovídá velikost první položky.

```
SPT01  defb 3*hh,hh/2
        defw TABLE+aa
        defw nn
        defw mm
```

A teď už nás čeká samotná rutina. Program vykresluje sprajty zaznamenané v QTABLE, nic nekontroluje a při sebemenší chybě se může zhroutit. Čím dále v tabulce sprajt leží, tím bude "blíže" (bude překrývat ty pod ním) na obrazovky. Při běhu rutiny je nutné zakázat přerušování (pokud ho povolíte, musíte mít jistotu, že v době běhu nebude vyvoláno).

```
QDRAW  ld (QD09+1),sp ;ulož hodnotu SP
        ld hl,QTABLE ;do HL tabulka sprajtů
        ld b,(hl)     ;do B počet sprajtů
        inc hl        ;ukazuj na data 1. sprajtu
        ld de,4       ;délka položky v tabulce je 4 bajty
QD00   ld sp,hl       ;HL do SP
        exx           ;vyměň sadu registrů
        pop hl        ;vytáhni příznaky (L) a rezervu (H)
        inc l         ;testuj nulu - sprajt se nekreslí
        dec l
        jr z,QD08     ;skoč na další sprajt
```

zde by se dalo dopsat větvení rutiny podle šířky sprajtu, která by teď byla v registru H

```
        pop hl        ;vytáhni adresu dat sprajtu
        ld sp,hl      ;přendej ji do SP
        pop bc        ;vytáhni rozměry
        pop hl        ;vytáhni ukazatel do tabulky adres VRAM
        pop de        ;ukazatel na grafiku
        ld sp,hl      ;SP ukazuje do tabulky adres VRAM
        ex de,hl      ;HL=grafika
QD01   pop de        ;vytáhni adresu ve VRAM, kam kreslí
        ld a,(de)     ;notoricky známá část z minule
        and (hl)
        inc hl
        or (hl)
        inc hl
        ld (de),a
        inc e         ;vpravo
        ld a,(de)
        and (hl)
        inc hl
        or (hl)
        inc hl
        ld (de),a
        inc e         ;vpravo
        ld a,(de)
        and (hl)
        inc hl
        or (hl)
        inc hl
        ld (de),a
```

malá změna - zůstáváme ve stejném sloupci, ale zato se posunujeme o řádek níže...

```
        inc d
        ld a,(de)
        and (hl)
        inc hl
        or (hl)
        inc hl
        ld (de),a
        dec e         ;vlevo (zpátky)
        ld a,(de)
        and (hl)
        inc hl
        or (hl)
        inc hl
        ld (de),a
        dec e         ;vlevo
        ld a,(de)
```


and (hl)
inc hl
or (hl)
inc hl
ld (de),a
djnz QD01

smyčka se bude opakovat B-krát; vzhledem k tomu, že v jednom průchodu nakreslíme dva řádky sprajtu, je v B uložena pouze poloviční hodnota výšky

QD08	exx	;vrať původní sadu registrů
	add hl,de	;posuň se na další položku v QTABLE
	djnz QD00	;obsluž B sprajtu
QD09	ld sp,0	;obnov správnou hodnotu zásobníku
	ret	;a vrať se

Podobně jako tato rutina vypadají i rutiny na schování/obnovení pozadí. Zde už si ale musíte sami rozmyslet, jestli chcete vždy vykreslovat celou obrazovku znovu (pak nepotřebujete ukládat pozadí), nebo sprajty vždy sbírat a znovu pokládat (nejlepší je stáhnout vykreslování sprajtu a ukládání pozadí do jedné rutiny, ušetří se tím spousta času). Já se kloním k prvnímu řešení, protože se v něm jednodušeji ovládají operace s pozadím - např. jeho změny, posuny, atd. Druhé řešení je výhodné pro nízký počet sprajtů a pro hry, které nevyžadují rychlost (toto řešení je využíváno např. "plovoucí šipkou"). Kombinací vykreslovacích postupů je spousta - můžete vykreslovat jak pozadí, tak sprajty někam do paměti (vnitřní obrazovka) a pak všechno přenášet do VRAM, nebo kopírovat pozadí z vnitřní obrazovky do VRAM a potom dokreslovat sprajty až tam, atd.

Data jednotlivých sprajtů Vám doporučuji rozšířit o položku výkonného podprogramu, který ovládá pohyb (strategii, "myšlení") sprajtu. Váš hlavní ovládací algoritmus pak bude vypadat přibližně takto (v případě, že neuschováváte pozadí pod sprajty):

- 1) vykresli pozadí a sprajty
- 2) obsluž všechny sprajty (vypočítá nové polohy)
- 3) znovu od 1)

Máte-li statické pozadí a uschováváte si jeho překryté části sprajty, mohli byste to dělat takto:

- 1) schovej sprajty (=vrať pozadí)
- 2) obsluž sprajty (nové polohy)
- 3) kresli sprajty (+ukládej pozadí)
- 4) znovu od 1)

Má to ovšem jednu podstatnou vadu: než stihnete obsloužit všechny sprajty, uplyne obvykle tak dlouhá doba, že paprsek se dostane "před" místa, kde se bude kreslit - sprajty pak budou poblikávat, protože ULA je vykreslí až v příštím přerušení. Existuje celkem jednoduché řešení - prohodit kroky 1) a 2), čímž minimalizujete dobu mezi smazáním a vykreslením sprajtu. Musíte ale rozšířit tabulku dat sprajtu o novou položku - druhý ukazatel do tabulky adres VRAM; první ukazatel bude říkat, kam vrátit staré pozadí, druhý odkud vzít nové a kam vykreslit sprajt. Po vykreslení přepokopujete hodnotu druhého ukazatele do prvního a v prvním kroku si vypočtete novou pozici do prvního...

To by pro dnešek stačilo, příště se pokusím k té spoustě teorie sestavit konkrétní (možná i fungující) příklad.

(pokračování příště)

Jak začínali... Universum

O prázdninách jsem napsal první díl knížky **Assembler z ZX Spectrum**. Současně jsme navázali spolupráci s časopisem ZX Magazin, který v té době ještě vydával David Hertl.

Na konci prázdnin v roce 1991 jsem "zabředl" do citového dobrodružství, jehož následky (nutno říci, že poměrně příjemné) nesu dodnes. Nikdy bych nevěřil, jaký katastrofální vliv může mít na člověka láska, nerad bych si to zopakoval. Pozdní projevy dočasně vyšinutí mysli můžete najít v mé další hře MAH JONGG.

Ve stejné době jsme museli čelit potížím ve firmě - museli jsme se rozejít s Jirkou Vondráčkem a tak jsem se později stal druhým společníkem firmy.

V zimě roku 91 jsme od Davida převzali ZX Magazin.

Na začátku roku 1992 jsme spolu s Georgem K. psali TOOLS 40 (byla to náhrada za ZX TOOLS, který měl původně napsat Jirka Vondráček).

Ve škole se blížil konec - pátý ročník a státnice, na nějaký čas jsem byl vyřazen z programování. Státnice dopadly dobře (2,2,2,1) a stal jsem se Mgr. - tuhle zkratku si vykládejte jak chcete, daleko od pravdy nebudete.

S dalším mým programem jste se mohli setkat nepřímo - pokud jste si od nás v posledním roce něco objednávali na dobříku, dostali jste na tiskárně napsaný dodací list a na tiskárně vyplněnou složku. Program na zpracování dobříček nám umožnil značně urychlit vyřizování Vašich objednávek.

Poslední prázdniny znamenaly další práci na TOOLSu, tentokrát 80, na programu Apollon, který jsem pomáhal dokončovat. Objevil se také

KUD 1, ULTRA BT a disková verze PROMETHEA.

V říjnu jsem napsal druhý díl knížky **Assembler z ZX Spectrum**. Ve srovnání s prvním dílem obsahoval mnohem zajímavější příklady.

Na podzim jsem začal tvořit svoji další hru - **INFERNO**. Hru jsem dokončil poměrně nedávno, v květnu tohoto roku. Považuji ji za svůj nejlepší program. Reakce okolí na něj nebyla tak úplně jednoznačná - když jsem poslal do Boleslavi první verzi, setkala se dokonce s odporem (viz úvodní text v programu), SCALEX vyloženě odmítl planetu Fialka (navrhoval mi dokonce, abych raději vytvořil Červínku, poslal mi dokonce ideový návrh - byla to červená krajina se spoustou klasických obrázků prasátek). Nakonec ale museli všichni uznat, že se jim to líbí, Jan F. mi dokonce telefonicky blahopřál.

Práce na **INFERNU** byla přerušena vznikem nové verze **PRESSORU**, tentokrát už šesté. Přímý popud k jeho vytvoření mi poskytl Petr H., přepsal totiž původní **PRESSOR** v celý z básic do assembleru, přidal nějaké rutiny z **TOOLSu** a pod to celé se hrdě podepsal a to beze zmínky o tom, že by vycházel z cizího programu, přestože některé části byly pouze přesunuty na jiné místo. Původně jsme chtěli program vzít do distribuce (dohodli jsme se dodatečně na spoluautorství) ale některé chyby v programu a ne příliš velká původnost programu mě nakonec vyhecovaly - vznikl úplně nový **Pressor VI** - na rozdíl od předchozích verzí uměl spolupracovat i s disketovou jednotkou D40 a hlavně komprimovat i jen část obrazovky.

Mimo **PRESSORu** jsem při **INFERNU** pracoval na několika nových utilitách pro **DESKTOP - KUD 3**.

Momentálně pracuji na tabulkovém procesoru, už má dokonce jméno: **CALCULUS**.

Je to první program, který má jméno dřív, než je alespoň z poloviny hotov. Jestli se mi podaří tak, jak si jej představuji, bude to užitečný program - na disketě bude umět pracovat se soubory většími než je operační paměť počítače a bude s ním možné vést například peněžní deník u jednoduchého účetnictví nebo provádět složité výpočty, také jednodušší databáze kartotékového typu se v něm budou dávat realizovat.

Výhledově uvažuji o programu **APOLLON 2**, který by uměl totéž co **APOLLON**, ale navíc by mohl pracovat s velkými soubory na disku, měl čitelnější písma a další vylepšení.

- Universum -



SYSTÉMOVÉ PROMĚNNÉ

Question: What is on program ?

Answer: Systémové proměnné

(c) 1993.02.08. Pavel Rak

Paměť pro počítač je stejně potřebná, jako pro člověka peníze. Čím více, tím lépe. To je nesporné, což mi potvrdí každý z Vás.



Jak říká jeden z Murphyho zákonů: "Paměti není nikdy dost". Z vlastní zkušenosti to mohu jen dosvědčit. (Nyní pracuji na ATARI MEGA STE s 1 MB RAM, což je považováno opravdu za slabé minimum k práci - pozn. aut.) Všemi milované Spectrum s CPU (Central Processing Unit) Z-80 A vnútri "živěné" 3.5 MHz však může obhospodařovat najednou 64 Kb paměti. Ať je to RAM (Random Acces Memory) či ROM (Read Only Memory), Spectru je to jedno. Ale nám ne, protože my můžeme pro své náročné programy používat pouze RAM. ROMka slouží pouze ke čtení a v případě Spectra je v ní uložen BASIC, nutno dodat že docela dobrý.

Jen tak pro zajímavost uvedu, že RAM ve Spectru je paměť dynamická, proto se musí každé 2 ms obnovovat, jinak bychom o své data navždy přišli. O obnovu dat uvnitř počítače se stará chytrý "šváb" zvaný ULA. I procesor si na obnovu paměti zřídil jeden registr a to registr R (memory refresh), ale to už bych zacházel do přílišných podrobností. Data do ROMky byla uložena při výrobě a ta nelze tak jednoduše měnit. Pokud byste ale měli zájem o změnu ROMky, tak je nutné buď naprogramovat čistou EPROMku nebo tuto svoji smazat ultrafialovým zářením a naprogramovat. To jsou ale věci, které dělají odborníci na slovo vzatí. Tuto činnost radši přenechte jim. Obsah ROM je stálý, tzn., že i po vypnutí proudu tam data zůstanou, což u paměti RAM nehrozí. Jistě se ptáte, proč procesor může adresovat najednou pouze 65536 bajtů. Je to dáno šířkou jeho adresové sběrnice, která má 16 vývodů, jež jsou na schématech většinou označovány A0 - A15. Stačí se podívat na Obr. 4 v knize [1] na staně 16 a je to hned jasné. Pokud

vynásobíte kouzelné číslo 2 na 16 dostanete 65536 což odpovídá našim 64 Kb. U Spectra je prvních 16 Kb věnováno ROMce a zbylých 48 Kb je vyhrazeno RAMce. Majitelé Didaktiků Gama ale namítnou, že oni mají svých 80 Kb. Ano mají, ale nesmí zapomenout, že tato paměť není adresována vcelku, ale používá se tu finta s přepínáním paměťových bank po 32 Kb. Z toho plyne, že mohou použít buď bank A nebo bank B. Nikoliv současně! Pouze součet dostupné paměti typu RAM dává ono magické číslo 80 Kb. Chudák procesor už při této hodnotě jaksi nestíhá, je to nad jeho síly. Výrobce Didaktiků použil na přepínání banků port C0, což poté způsobilo částečnou nekompatibilitu s perifériemi, které na Spectru šlapaly bez problémů, např. tiskárny s připojením přes port C, atd.

Dnes se budu věnovat té malé části RAMky od adresy 23552 až po adresu 23733 o které bylo a ještě bude napsáno hodně. Pro začátečníky upřesňuji, že tato oblast je někým (výrobcem a programátory) označována jako Systémové proměnné. Také já zůstanu tomuto označení věrný.

Oblast RAM od adresy 23553 a délky 181 bajtů je místo kde můžeme najít Systémové Proměnné (SP) počítače (pro jednoduchost zkracuji na SP, ale jinak se takto označuje ukazatel zásobníku Stack Pointer). SP obsahují informace o okamžitém stavu systému. Jednotlivé SP zabírají jeden bajt, dva bajty (pro dámy a pány v assembleru slovo neboli anglicky word) nebo případně i více bajtů. Rekord drží proměnná STRMS, která okupuje 38 bajtů. Po resetu počítače se SP nastaví na standardní hodnoty, které jsou v ROMce. Pokud ale provedete restart příkazem NEW, tak se Vám zachovají 3 proměnné: pípní klávesnice, hodnota CLEAR a adresa UDG. Tohle způsobí 3 instrukce na adrese #11bf v ROM. Délka 181 bajtů platí v případě, že nemáte k svému "gumákovi" připojený Microdrive nebo INTERFACE I. Několik na sobě nezávislých zdrojů tvrdí, že s připojením těchto důmyslných vynálezů se posunuje i adresa konce SP. Dodnes jsem se s těmito zařízeními nesetkal (a snad už nesetkám, pozn. aut.), proto to nemohu objektivně posoudit. Zjistil jsem, že s připojením Interface I či Mikrodrive se SP prodlouží o 21 bajtů. Ale asi to tak bude, kdo to ví, ať odpoví... nejlépe písemně na adresu redakce, která mu to jistě ráda zveřejní.

Nemusíte se bát, že s připojením "interfejsu" UR-4 nebo D-40 se Vám v paměti něco radikálně změní. Tato zařízení Vám v podstatě rozšíří možnosti ubohého Spectra. Tak např. D-40 má v sobě zabudovanou vlastní statickou RAM

(protože statické paměti jsou velmi rychlé, za to ale pěkně hřejou a "sežerou") o velikosti 2 Kb, takže Vám nesebere ani bit z klasické RAMky. Do své statické RAMky si D-40 & spol. ukládá všechny potřebné informace o svém provozu. Namátkou vyberu: číslo následujícího snapu, formát diskety (počet stop, sektorů).

SP je oblast v paměti, proto je lze jednoduše číst z basicu příkazem PEEK (adresa) a měnit příkazem POKE adr,hodnota. Hodnota je v rozmezí 0-255. Je možné vložit třeba i -1. Počítač je chytrý on si to převede na 255. Lze samozřejmě dát i -2, Spectrum to přelouská. Hodnoty větší než 255 se ukládají pomocí drobné finty uvedené později. Jediné co asi nejde, je volat nějakou SP příkazem USR adresa. (Ha, Ha, Ha - smích v okolí) Pozor! Neuvážené změny SP mohou způsobit (někdy) havárii systému, proto měním jen ty SP, u kterých vím co se stane když to změním. Experimenty se doporučují jen v případě, že v paměti není žádný strategický program.

Pokud pracujete ve strojáku a nemíníte se už vůbec odkazovat na basic ani nepoužívat rutiny z ROMky, tak můžete oblast SP použít jako normální paměť, třeba jako buffer pro text. Většinou je ale dobré, když už programujete stylem STROJÁK - BASIC, když budete SP používat a veasele se na ně odkazovat a využívat jejich služeb. Je to výhodné, uspoří Vám to někdy dost námahy a času. S velmi pěkným využitím SP ze strojáku jsem se setkal (mimo svých vlastních programů samozřejmě) u programu DESKTOP fy PROXIMA ale i u jejich dalších programů. Musím uznat, že u DESKTOPU vyvinuli perfektní řešení. (Poznámka průměrného uživatele: "Oni ti maníci v Proximě jsou nějak moc chytří a dobří!").

Důležitá technická poznámka pro pracovníky ve strojáku: při nedestruktivním návratu ze svého programu ze strojáku do basicu musí (zdůrazňuji MUSÍ) být registr IY nastaven na hodnotu #5c3a, protože ROMka se na většinu SP odkazuje příkazy ld (iy+xx),a, kde xx je příslušný pevně zvolený odskok (displacement). Kdyby IY mělo jinou hodnotu, tak by interpret měnil jiné SP a systém by to asi psychicky nepřenesl přes srdce a kolaboval by. Dejte si na to veliký pozor.

Následuje přehled SP o kterých se mi podařilo něco získat. Na závěr je uvedena tabulka se stručnou charakteristikou.

KSTATE 23552 počítač používá při vyhodnocování stlačené klávesy nebo při stlačení dvou kláves

LAST_K 23560 obsahuje kód poslední stlačené klávesy s ohledem na stlačení SHITFu, rozlišuje tedy mexi x a X

REPDEL 23561 obsahuje dobu po kterou musí být stlačena klávesa, aby došlo k opakování, čas je v 1/50 sekundy, pokud chcete někomu ztřížit editování vašeho programu, tak dejte POKE 23561,1, nastavení v ROM na #1270 - hodnota je v registru H

REPPER 23562 doba v 1/50 sekundy která určuje interval opakování při stlačení klávesy, pro rychlejší editování basicu tam dejte 1

DEFADD 23563 adresa argumentu uživatelské funkce DEF FN

K_DATA 23565 ukládá se barevná informace vložená klávesnicí před dalším zpracováním

TVDATA 23566 obsahuje řídicí znaky o AT, TAB a informace o barvě

STRMS 23568 obsahuje adresy kanálů připojených k jednotlivým proudům, při inicializaci se prvních 15 bajtů přenesou z ROM, přenos na adr. #123b, tabulka je na #15af

CHARS 23606 říká, kde se nachází font pro výpis znaků na obrazovku, adresa je menší o 256, protože prvních 32 znaků z tabulky Spectrum nevyepisuje jako znaky, má tam totiž své řídicí znaky např. pro barvy, pozici AT ..., pokud máte svůj font (např. z ART-STUDIA) o délce 768 B a chce jej nahrát do paměti na adresu 32000, tak ho tam nahrajte příkazem LOAD "" CODE 32e3, před LOAD je lepší snížit ramtop příkazem CLEAR 31999, pokud jste si ho nenastavili jinak; pak stačí napsat RANDOMIZE 32e3-256: POKE 23606, PEEK 23670: POKE 23607, PEEK 23671, po odeslání řádku kl. ENTER by jste měli vidět ten svůj font, návrat k původnímu fontu v ROM udělá POKE 23606,0: POKE 23607,60; nastavení fontu v ROM je na #121c, font v ROM začíná na #3c00 a má délku 768 B

RASP 23608 udává délku varovného beepu např. při psaní řádku přes celou obrazovku, editaci řádku s REM ve kterém je nějaký stroják...

PIP 23609 délka pípnutí při stisku klávesy (vyjímaje CAPS SHIFT a SYMBOL SHIFT), ideální hodnota je 0; nastavení RASP a PIP je v ROM na #1212, dává se to do registru BC

ERR_NR 23610 na této adrese je hodnota o 1 menší než je kód chybového hlášení, např. při hlášení 0 OK, 0:1 je v této SP hodnota 255, pokud chcete program ukončit nějakým efektním hlášením, tak dejte na poslední řádek prováděného programu POKE 23612, XX (kde XX je kód chybového hlášení mínus jedna), Seznam chybových hlášení najdete v [2] v příloze A

FLAGS 23611 různá návěští pro basic, nespecifikovatelné

TVFLAG 23612 návěští pro řízení obrazovky

ERR_SP 23613 adresa položky zásobníku ve kterém je uložena adresa pro obsluhu chyby; stručně řečeno: pokud se interpret vrací s nějakým chybovým hlášením (nezapomeňte, že mezi chybová hlášení patří i OK, viz příloha A v [2]), tak sejme z ERR_SP nějakou adresu, z této adresy si vyzvedne hodnotu (od 0 po 65535) na kterou nastaví registr PC (Program Counter) a skočí na tuto adresu, pokud v paměti na adrese určené obsahem ERR_SP budou nuly, program při jakémkoliv pokusu o výpis chyb. hlášení provede reset - vtipné, že?

LIST_SP 23615 adresa kam skočí interpret pokud úspěšně ukončí listing programu, v praxi je totožná s obsahem ERR_SP, hodnota je vždy o 3 bajty menší než je hodnota RAMTOP

MODE 23617 obsahuje režim kurzoru při editaci řádku

NEWPPC 23618 obsahuje číslo řádku na který má skočit příkaz GO TO, GO SUB, změnou lze počítač donutit aby skákal na námi zvolenou řádku

NSPPC 23620 určuje číslo příkazu na který má skočit GO TO, GO SUB, kombinacemi NEWPPC a NSPPC je možno spustit program příkazy: POKE 23618,0: POKE 236190: POKE 23620,0

PPC 23621 číslo právě prováděného řádku (probíhá od 0-9999, protože basic větší čísla nezvládá)

SUBPPC 23623 číslo vykonávaného příkazu v PPC

BORDC 23624 barevná informace pro spodní část obrazovky (editační zóna), zkuste toto: POKE 23624,166: INPUT "": PAUSE 50: POKE 23624,7; pokud změníte BORDC změní se i barva okraje obrazovky (BORDER) a naopak, změníte-li BORDER stejným příkazem, nastaví se BORDC a tím spodní část obr. na barvu u BORDERu

E_PPC 23625 číslo řádku na kterém je kurzor

VARS 23627 adresa počátku proměnných, podrobný popis v [4] strany 4, 5, 6

DEST 23629 adresa proměnné při jejím vyhodnocování, jinak 0

CHANS 23631 adr. začátku informací o kanálech, bez INTERFACE I je tam 23734 (#5cb6)

CHURCHL 23633 obsahuje adr. z oblasti kanálových dat pro právě otevřený kanál

PROG 23635 adr. počátku basicového programu, bez INTERFACE I

23755 (#5ccb), lze využít k utajení prvního řádku programu příkazy: POKE 23755,0: POKE 23756,0; které ovšem již

nelze editovat, editaci umožní změna adresy 23755

NXTLIN 23637 začáteční adresa dalšího programového řádku, který má být programem proveden

DATADD 23639 obsahuje adresu čárky za poslední položkou v řádku DATA, která byla přečtena příkazem READ

E_LINE 23641 obs. adresu editované oblasti v RAM, kde se zrovna edituje řádek

K_CUR 23643 adresa kurzoru (že by řádek, sloupec ?)

CH_ADD 23645 adresa znaku v oblasti INPUT nebo oblasti proměnných, který bude jako příští vypsán na obrazovku

X_PTR 23647 adr. prvního znaku ve kterém je syntaktická chyba, t.j. znaku následujícím po "?"

WORKSP 23649 adr. dočasného pracovního prostoru

STKBOT 23651 adresa začátku oblasti zásobníku kalkulátoru

STKEND 23653 adr. vrcholu zásobníku kalkulátoru (začátek volné paměti)

BREG 23655 používá kalkulátor jako své počítadlo

MEM 23656 adr. začátku zásobníkové paměti kalkulátoru, většinou MEMBOT

FLAGS2 23658 další parametry pro basic, změnou na 8 lze programově přepnout na velká písmena při psaní textu (jednoduše řečeno zapne CAPS LOCK), zpětně lze přepnout na malá písmena změnou na 0



DF_SZ 23659 udává počet řádků v dolní části obrazovky; zajímavý je efekt při POKE 23659,1; v programu můžete dát POKE 23659,0 a bude vám fungovat i 22 pozice pro PRINT AT; nesmíte se ale vrátit do basicu s nějakou hláškou např. při INPUTu, scroll?, BREAK, OK, ..., protože počítač nemá kam vypisovat text a zakousne se (ne do Vás, ale sám do sebe, pozn. aut.)

S_TOP 23660 obsahuje číslo řádku, který bude zobrazen jako první při automatickém listingu

OLDPPC 23662 číslo řádku kterým bude pokračovat program po CONTINUE

OSPPC 23664 číslo příkazu na který skočí CONTINUE

FLAGX 23665 různé parametry, není o tom ani slovo v [2], [3], [4] ani v [5]

STRLEN 23666 délka právě vykonávaného řetězce

T_ADDR 23668 adresa další položky v syntaktické tabulce

SEED 23670 slouží pro generování náhodných čísel, náhodná čísla se negenerují náhodně, ale podle pevně daného programu, příkaz RANDOMIZE číslo (0-65535) rozloží na dvě osmibitová, výsledné číslo je pak dáno vztahem číslo=PEEK 23670+256*PEEK 23671

FRAMES 23672 při každém přerušení (50 krát za sekundu) se obsah FRAMES zvětší o 1, po resetu se nastaví na 0, lze spočítat kolik sekund je počítač zapnutý a to výrazem: TIME=(PEEK 23672+256*PEEK 23673*PEEK 23674)/50, přesnost je závislá na kmitočtu v el. sfti, který se mění den ze dne, z hodiny na hodinu, dělením to lze převést na formát HH:MM:SS, pokud je ale zakázáno přerušení (BEEP, SAVE, LOAD...), tak k inkrementaci nedochází

UDG 23675 adresa prvního znaku uživatelské grafiky (UDG), po resetu se nastaví na 65368 (#ff58), font pro UDG je v ROM na #3eaf

COORDS 23677 obsahuje x-ovou souřadnici posledního bodu nakresleného příkazem PLOT, DRAW

23678 totéž po y-ovou souřadnici

P_POSN 23679 číslo sloupce pozice LPRINT

PR_CC 23680 bajt příští pozice tisku při LPRINT, méně významný

noname 23681 většinou volný, občas je používán jako počítadlo vytisknutých řádků na tiskárně

ECHO_E 23682 adresa ve spodní části obrazovky za kterou nejde dál posunout kurzor směrem doprava

DF_CC 23684 adresa 1 bajtu psaného na obrazovku

DF_CCL 23686 platí totéž co bylo uvedeno předtím pro spodní část obrazovky

S_POSN 23688 obsahuje číslo sloupce 23558 (#5c88) a číslo řádku v 23689 (#5c89) běžné pozice pro PRINT, POZOR: řádky a sloupce mají vnitřně jiné číslování než jsme zvyklí z basicu, podrobnosti v tabulce č. 5 v [4]

S_POSNL 23690 totéž ale pro spodní část obrazovky

SCR_CT 23692 počet řádků vypsane příkazem LIST dokud se neobjeví hláška scroll?, zkuste tam dát 255 (nebo -1 jak je libo) a příkaz LIST

ATTR_P 23693 barevná informace pro právě tisknutý znak, nastavení v ROM je na #1265

MASK_P 23694 obsahuje masku pro ATTR_P

ATTR_T 23695 barevná informace o právě tisknutém znaku, nastavení v ROM je na #126a

MASK_T 23696 maska pro ATTR_T, pouze dočasně

P_FLAG 23697 další info o systému

MEMBOT 23698 zásobník kalkulátoru pro uložení hodnot se kterými pracuje, další informace jsou v [4]

MNI 23728 vektor pro NMI, lze využít pouze u Didaktiku Gama, protože Spectrum (Didaktik M) mají špatně v ROM ošetřenou obsluhu tohoto vektoru, jinak při přijetí signálu MNI procesor uloží poslední adresu hlavního programu z PC do zásobníku a PC napevno nastaví adresu 102 (#66), na této adrese začíná program pro obsluhu nemaskovaného přerušení, které vezme adresu z SP NMI a skočí na tuto adresu, více o NMI v [1] na str. 31, Spectristé tyto dva bajty mohou použít pro své potřeby

RAMTOP 23730 adresa posledního bajtu paměti využitelné pro basic

P-RAMT 23732 adresa posledního fyzického bytu RAM

Použitá literatura:

[1] Vladimír Dvořák: Mikroprocesorové systémy Z80 a I 8085, TESLA ELTOS, první vydání, 1986

[2] RNDr. František Palkovič, RNDr. Roman Iška: Didaktik M, uživatelská příručka, Výrobné družstvo DIDAKTIK SKALICA, druhé vydání, marec 1991

[3] DIDAKTIK GAMA, návod k obsluhu

[4] PROGRAMOVÁNÍ VE STROJNÍM KÓDU, 3. ČÁST

[5] vlastní zápisy

Příloha

Na následující straně najdete seznam systémových proměnných se stručnou charakteristikou. Vysvětlivky k tabulce a k jednotlivým kolonkám:

- 1) název SP
- 2) adresa SP v HEXu
- 3) adresa SP v DECU
- 4) délka této proměnné
- 5) jak je definována (předdefinována po resetu)
- 6) číslo xx při odkazu na SP příkazem ld a,(iy+xx) xx je uvedeno pouze v HEXu

Pavel Rak

Výpočetní technika

- ✓ prodej počítačů, periférií, tiskáren
- ✓ prodej software pro počítače ZX Spectrum a kompatibilní
- ✓ kompletní nabídka programů firmy PROXIMA

Firma CONSUL, Pálenická 28, 323 17 Pízeň
tel. 019 / 523 721

SAM - sériové rozhraní

Počítač SAM Coupé bývá zpravidla dodáván s paralelním nebo sériovým rozhraním (podle přání zákazníka). U nás se většinou vyskytují verze s paralelním rozhraním Centronics. Problém nastává, pokud k SAMovi potřebujeme připojit např. sériovou tiskárnu K6304, zapisovač Colorgraf 0512 či 0516 nebo XY4160 s RS-232.

Teoreticky by šlo k výstupu Centronics přidat převodníky TTL/RS-232 a vnější zdroj napětí +12 V a -12 V a napsat program. Běh programu ale u SAMa není rovnoměrný a stejně jako v případě Didaktiku M je zpomalován činností zákaznického obvodu.

Z těchto důvodů bylo sestaveno a v praxi vyzkoušeno jednoduché sériové rozhraní pro SAMa, které nevyžaduje ani zásah do SAMa, ani obslužný program a funguje prakticky ihned po zapnutí SAMa bez potřeby cokoliv nahraovat.

Bylo vyzkoušeno s výše uvedenými zapisovači, tiskárnami K6304 (sériová), GP500AS a dalšími. Vychází asi 2x levněji než firemní sériové rozhraní. Případní zájemci se mohou přes redakci obrátit na autora článku.

- rex -

název	hex	dec	délka	definice	iy	stručný popis
KSTATE	5c00	23552	8	ds 8	c6	vyhodnocování stlačené klávesy
LAST_K	5c08	23560	1	db 0	ce	uložen kód poslední stlačené klávesy
REPDEL	5c09	23561	1	db 35	cf	doba k repeat klávesy v 1/50 sekundy
REPPER	5c0a	23562	1	db 05	d0	opakování stlačení klávesy v 1/50 sekundy
DEFADD	5c0b	23563	2	dw 0	d1	adr. argumentu uživatelské funkce DEF FN
K_DATA	5c0d	23565	1	db 0	d3	barevné inf. z klávesnice před zpracováním
TVDATA	5c0f	23566	2	dw 0	d4	řídící znaky o AT, TAB, barvě
STRMS	5c10	23568	38	ds 38	d6	adresy kanálů k jednotlivým proudům
CHARS	5c36	23606	2	dw #3c00	fc	adresa fontu 8*8 pro výpis znaků na obr.
RASP	5c38	23608	1	db 64	fe	délka varovného beupu
PIP	5c39	23609	1	db 20	ff	délka pípnutí při stisku klávesy
ERR_NR	5c3d	23610	1	db 255	00	o 1 menší než kód chybového hlášení
FLAGS	5c3b	23611	1	ds 1	01	různé informace
TVFLAG	5c3c	23612	1	ds 1	02	parametry pro řízení obrazovky
ERR_SP	5c3d	23613	2	dw #ff54	03	návratová adresa při RST #08
LIST_SP	5c3f	23615	2	dw #ff54	05	návratová adresa při ukončení LISTU
MODE	5c41	23617	1	db 0	07	režim kurzoru (K,L,C,E,G)
NEWPPC	5c42	23618	2	ds 2	08	číslo řádku pro GO TO, GO SUB
NSPPC	5c44	23620	1	ds 1	0a	číslo příkazu v NEWPPC
PPC	5c45	23621	2	ds 2	0b	číslo právě vykonávaného řádku
SUBPPC	5c47	23623	1	ds 1	0d	číslo vykonávaného příkazu v PPC
BORDC	5c48	23624	1	db 56	0e	barevná informace pro spodní část obrazovky
E_PPC	5c49	23625	2	ds 2	0f	číslo řádku na kterém je kurzor
VARS	5c4b	23627	2	dw ?	11	adresa začátku proměnných
DEST	5c4d	23629	2	ds 2	13	adresa proměnné při zpracování
CHURCHL	5c51	23633	2	dw #5cbb	17	adresa právě otevřeného kanálu
PROG	5c53	23635	2	dw #5ccb	19	adresa počátku basicu bez INTERFACE 1
NXTLIN	5c55	23637	2	ds 2	1b	adresa dalšího řádku v programu
DATADD	5c57	23639	2	ds 2	1d	adresa naposledy čtené položky u DATA
E_LINE	5c59	23641	2	dw #5ccc	1f	adresa právě psaného příkazu
K_CUR	5c5b	23643	2	ds 2	21	adresa kurzoru
CH_ADD	5c5d	23645	2	ds 2	23	adresa znaku, který se bude zpracovávat
X_PTR	5c5f	23647	2	ds 2	25	adresa výskytu syntaktické chyby
WORKSP	5c61	23649	2	dw #5d08	27	adresa pracovního prostoru
STKBOT	5c63	23651	2	dw #5d08	29	adresa začátku zásobníku kalkulátoru
STKEND	5c65	23653	2	dw #5d08	2b	adresa vrcholu zásobníku kalkulátoru
BREG	5c67	23655	1	ds 1	2d	počítadlo kalkulátoru
MEM	5c68	23656	2	dw #5c92	2e	adresa paměti pro kalkulátor
FLAGS2	5c6a	23658	1	ds 1	30	další různé informace
DF_SZ	5c6b	23659	1	db 2	31	počet řádků v dolní části obrazovky
S_TOP	5c6c	23660	2	ds 2	32	číslo řádku zobrazené jako první při LISTU
OLDPPC	5c6e	23662	2	ds 2	34	číslo řádku na který skočí CONTINUE
OSPPC	5c70	23664	1	ds 1	36	číslo příkazu na který skočí CONTINUE
FLAGX	5c71	23665	1	ds 1	37	různé informace
STRLEN	5c72	23666	2	ds 2	38	délka vyhodnocovaného řetězce
T_ADDR	5c74	23668	2	ds 2	3a	adresa další položky v syntaktické tabulce
SEED	5c76	23670	2	dw 0	3c	SP pro generátor náhodných čísel
FRAMES	5c78	23672	3	ds 3	3e	informace o čase, inkrementace co 20 ms
UDG	5c7b	23675	2	dw #ff58	41	adresa prvního znaku UDG
COORDS	5c7d	23677	1	ds 1	43	x-ová souřadnice posledně nakresleného bodu
	5c7e	23678	1	ds 1	44	y-ová souřadnice posledně nakresleného bodu
P_POSN	5c7f	23679	1	ds 1	45	číslo sloupce u LPRINT
PR_CC	5c80	23680	2	ds 2	46	spodní bajt pro příští pozici tisku
noname	5c81	23681	1	ds 1	47	volno pro Vaše potřeby
ECHO_E	5c82	23682	2	ds 2	48	poslední adr. kurz. spodní části obrazovky
DF_CC	5c84	23684	2	ds 2	4a	adresa 1 bajtu psaného na obrazovku
DF_CCL	5c86	23686	2	ds 2	4c	totéž co předtím, ale pro spodní část obr.
S_POSN	5c88	23688	2	ds 2	4e	souřadnice pozice pro PRINT
S_POSNL	5c8a	23690	2	ds 2	50	totéž pro spodní část obrazovky
SCR_CT	5c8c	23692	1	db 23	52	počet n+1 řádku vypsane před hl. scroll?
ATTR_P	5c8d	23693	1	db 56	53	nastavené barvy pro tisknutý znak
MASK_P	5c8e	23694	1	db 0	54	maska pro ATTR_P
ATTR_T	5c8f	23695	1	db 56	55	barevná informace o tisknutém znaku
MASK_T	5c90	23696	1	db 0	56	maska pro ATTR_T
P_FLAG	5c91	23697	1	ds 1	57	informace pro systém
MEMBOT	5c92	23698	30	ds 30	58	30bajtů pro zásobník kalkulátoru
NMI	5cb0	23728	2	dw 0	76	adresa obslužného programu NMI
RAMTOP	5cb2	23730	2	dw #ff57	78	adresa posledního bajtu pro basic
P_RAMT	5cb4	23732	2	dw -1	7a	adresa posledního bajtu fyzické paměti

STROJOVÝ KÓD Z-80

(trochu jinak, 7. díl)

Tím si přehrajeme kousek ROMky. Chceme-li skladbu s efekty BORDERu delší, lze z basicu pokračovat vyvoláváním přes příkaz RANDOMIZEUSR s adresou návěští VYBER.

A protože to nebyla žádná krása, zkusíme si udělat alespoň sirénu (než budeme mít zvukový procesor).

```

10      ORG 50000
20 START
30      LD C,#FF      ; čas od #FF do 0
40 ZVUK
50      XOR A        ; místo LD A,0
60      OUT (254),A  ; port s reproduktorkem
70      CALL WAIT    ; počká dobu dle C-reg.
80      LD A,16      ; ovládání membrány reproduktoru
90      OUT (254),A  ; port
10     CALL WAIT    ; počká dle C
110     DEC C        ; zkrácení času po každém cyklu
120     JR NZ,ZVUK   ; není-li C nulové zpět na ZVUK
130     RET          ; konec podprogramu
140 WAIT
150     LD B,C        ; plní B pro prodlevu
160 SMYCKA
170     DJNZ SMYCKA  ; čeká
180     RET

```

Podprogram po překladu vyzkoušejte v basicu např. takto:

```
300 RANDOMIZEUSR 50000:GOTO 300
```

Konečně vyzkoušíme rutinu BEEPER v paměti ROM:

```

LD DE,délka tónu ; např. 262
LD HL,výška tónu ; např. 1642
CALL #03B5
RET

```

Přičemž:

```

DE = frekvence tónu * délka tónu v sekundách
HL = 437500 / frekvence - 30.125
(oba výsledky zaokrouhlit na celé číslo)

```

Pozor! Během rutin BEEP, BEEPER je zakázáno přerušení.

Pokud v basicu zapíšeme BEEP x,y, pak x je délka tónu v sekundách a y je výška tónu vzhledem ke střednímu C v mezích -60 až +69. Pro strojový kód použijeme tento přepočít pro frekvenci f:

$$f = 440 * \text{EXP}((y-9) * \ln 2/12)$$

Z uvedeného lze vypočítat hodnoty DE a HL pro 1 sekundu trvající střední C:

```

DE = 262
HL = 1642

```

Druhá uvedená rutina BEEP na adrese #03F8 má vstupní parametry na vrcholu zásobníku kalkulátoru, proto podrobněji bude popsána později.

4. Restarty

V této kapitole se seznámíme s podprogramy, které se vyvolávají pomocí instrukce RST x. Jsou to častěji používané rutiny, u kterých je výhodné je vyvolávat přes RST x, protože proti klasickému CALL adr ušetříme 2 byty paměti.

Obecně se v systémech používá instrukci RST X jako instrukci programového přerušení. Ve Spectru je běžně užito módu přerušení IM 1, což znamená, že mikroprocesor Z80 neočekává žádnou reakci ze strany periferních zařízení při požadavku na maskovatelné přerušení a vždy přechází přímo na adresu #38. To znamená, že interně generuje instrukci RST #38.

4.1 START RST 0

Je vyvoláván po zapnutí systému, resp. po stisku tlačítka "reset", a samozřejmě instrukcí RST 0 nebo JP #0000.

Tento studený start systému začíná zakázáním přerušení a přechodem na inicializační rutinu na adrese #11CB. Je nastaven bílý border a testována paměť RAM, nastaveny systémové proměnné, tabulka grafických znaků naplněna písmeny A až U, definován SP registr, kanálové informace, základní černobílé nastavení, vypsáno hlášení (copyright). Rutina končí přechodem do hlavní smyčky prováděcího programu (interpretru). Volná paměť RAM je naplněna nulami, poslední správná adresa RAM po testu je zapsána do systémové proměnné P_RAMT, od ní je odvozena oblast pro UDG grafiku a RAMTOP.

4.2 ERROR_1 RST #08

Podprogram byl již popsán. Je volán vždy, je-li důvod k vypsání sdělení ve spodní části obrazovky při ukončení činnosti basicu nebo při zjištění chyby při jeho provádění. Po instrukci RST #08 musí následovat byt, ve kterém je zakódováno číslo hlášení, zmenšené o jedničku. Pro tabulku hlášení platí:

byt	kód	chybové hlášení
255	0	OK
00	1	NEXT without FOR
01	2	Variable not found
02	3	Subscript wrong
03	4	Out of memory
04	5	Out of Screen
05	6	Number too big
06	7	RETURN without GO SUB
07	8	End of file
08	9	STOP statement
09	A	Invalid argument
10	B	Integer out of range
11	C	Nonsense in BASIC
12	D	BREAK - CONT repeats
13	E	OUT of DATA
14	F	Invalid file name
15	G	No room for line
16	H	STOP in INPUT
17	I	FOR without NEXT
18	J	Invalid I/O device
19	K	Invalid colour

20	L	BREAK into program
21	M	RAMTOP no good
22	N	Statement lost
23	O	Invalid stream
24	P	FN without DEF
25	Q	Parametr error
26	R	Tape loading error

4.3 PRINT_A_1 RST #10

Rutina byla popsána již dříve.

4.4 GET-CHAR RST #18

Podprogram přisune aktuální znak (adresovaný systémovou proměnnou CH_ADD - #5C5D, 23645, IY+35) do registru A. Mezery (#20) a řídicí kódy (hodnota ASCII kodu < #20) rutina přeskakuje (neplatí pro #0D = "ENTER").

4.5 NEXT_CHAR RST #20

Podprogram přisune následující znak po znaku, který by přisunul podprogram RST #18. Znak je přesunut do A registru.

Napišme v basicu:

```
300 PRINT USR 50000,::: PRINT USR 50100,::: STOP
```

a zdrojové programy:

```
10      ORG 50000
20 GET  RST #18 ; GET_CHAR
30      RST #10 ; PRINT_A_1
40      RET

50      ORG 50100
60 NEXT RST #20 ; NEXT_CHAR
70      RST #10
80      RET
```

Po odstartování příkazů basicu (řádek 300) vypíše první příkaz čárku a obsah BC registru, t.j. adresu 50000 a druhý příkaz vypíše středník a adresu 50100.

4.6 FP_CALC RST #28

Podprogram zajistuje "přepnutí" na kalkulátor, který je umístěn na adrese #335B a je nejuniverzálnějším podprogramem ROM ZX Spectra. Jeho pomocí lze provádět veškeré funkce, které jsou v počítači zabudovány, a to včetně rozhodování a skoků.

Podprogram kalkulátoru vyvoláme instrukcí RST #28, po které následují kódy požadovaných funkcí (operací) kalkulátoru. Instrukce RST #28 předpokládá, že na vrcholu zásobníku kalkulátoru je již daná hodnota, se kterou bude provádět předepsanou operaci. I pro tuto úlohu jsou v ROMce potřebné podprogramy.

Pro složitost a souvislosti s jinými rutinami budeme kalkulátoru věnovat zvláštní kapitolu.

4.7 BC_SPACES RST #30

Tato rutina vytvoří v pracovní oblasti (t.j. mezi adresami WORKSP a STKBOT) pracovní prostor, jehož délka je dána obsahem BC registru.

Výstupní parametry rutiny:

DE obsahuje adresu 1. bytu nového pracovního prostoru
HL posledního bytu
BC dále podrží délku bloku

4.8 INTERR RST #38

Podprogram INTERRupt je 50x za vteřinu vyvoláván obvodem ULA. Během tohoto podprogramu je inkrementována tříbytová proměnná FRAMES (23672, #5C78, IY+62), která tak obsahuje informaci o čase. Dále je prováděn dotaz na klávesnici, což provádí vyvoláním rutiny na #02BF. Tato se odvolává na již známou rutinu KEY_SCAN na adrese #028E.

K vyhodnocení klávesnice používá 2*4 byty proměnné KSTATE, aby mohla zpracovat 2 následně stisknuté klávesy:

byt 0: co stlačeno (#FF = nic)
byt 1: počítadlo autorepeatu
byt 2: počítadlo pro spuštění autorepeatu
byt 3: kod stisknuté klávesy (ASCII nebo token)

5. FP-CALC RST #28

Podprogram kalkulátoru voláme instrukcí RST #28, po které musí následovat skupina bytů, kódů požadovaných operací kalkulátoru dle tabulky v uvedené příloze.

Vlastní operace kalkulátoru pracují se zásobníkem kalkulátoru, což je oblast paměti RAM mezi systémovými proměnnými STKBOT a STKEND. STKEND ukazuje za poslední byte z pětibytového vyjádření čísla na vrcholu zásobníku kalkulátoru.

K operacím kalkulátoru je využívána 30-ti bytová oblast RAM systémové proměnné MEMBOT. Zde lze uložit až 6 pětibytových čísel.

Systémové proměnné, které používá kalkulátor:

STKBOT 2B 23651 #5C63 IY+41 počáteční adresa zásobníku kalkulátoru
STKEND 2B 23653 #5C65 IY+43 vrchol zásobníku kalkulátoru
BREG 1B 23655 #5C67 IY+45 počítadlo kalkulátoru
MEM 2B 23656 #5C68 IY+46 obvykle adresa MEMBOTu
MEMBOT 30B 23698 #5C92 IY+88 až 6 x 5-ti bytové vyjádření čísla pro práci kalkulátoru (dtt0 jako registry u Z80CPU)

Před spuštěním instrukce RST #28 musíme nejprve na vrchol zásobníku kalkulátoru nachystat hodnoty, se kterými bude kalkulátor pracovat (Obdoba jazyka FORTH).

Proto probereme nejprve podprogramy, které dokáží zásobník kalkulátoru (dále jen ZK) plnit či z něj odebrat výsledky.

5.1 STACK_A CALL #2D28

Tento podprogram přenesou hodnotu z registru A na vrchol ZK a převede ji na typ floating point (FP).

Příklad:

```
10      ORG 50000
20 START
30      LD A,65
40      CALL #2D28 ; přenesou hodnotu 65 z A_reg. na
50                          ; vrchol ZK jako 5-ti bytové číslo
60      RET ; typu FP
```

(napsal Ing. Jiří Vacek, pokračování příště)

Nekonečné trápení s nekonečnými životy

I.ČÁST

Zadání Pouků do loaderů v Basicu

První část je určena všem, kteří již alespoň částečně ovládají programovací jazyk BASIC na ZX Spectru a kompatibilních počítačích (dále jen ZX Spectrum). Je zde podrobně popsáno, jak postupovat při vkládání pouků do loaderů v BASICu, nejčastější způsoby kódování včetně mnoha příkladů a najdete zde i spoustu dalších cenných informací, které vám pomohou nejen při vkládání pouků.

1.1. Nahrání BASICu bez autostartu

V této kapitole se budeme zabývat postupy, pomocí kterých lze nahrát úvodní BASIC hry tak, aniž by se spustil a pokračoval v nahrávání vlastní hry. Zabránění spuštění BASICu je první důležitá operace s hrou, bez které se při vkládání pouků dál nedostanete. Existuje spousta možností, jak toho dosáhnout. Pokusím se zde popsat nejpoužívanější a nejefektivnější metody. Jako ve všech kapitolách jsou seřazeny podle složitosti - od jednoduchých k složitým.

1.1.1. Nahrávání příkazem MERGE

Jedním z nejjednodušších způsobů je nahrání úvodního BASICu příkazem MERGE. Tento příkaz pracuje jako LOAD, ale neruší starý program a proměnné, pokud nemají řádky stejná čísla a proměnné stejné názvy. Říkáme tomu, že program přihráváme. Zjednodušeně to vypadá tak, že si počítač přihrávaný program nahrává do volné paměti (pokud je příliš malá, hlásí hned po nahrání jména chybu *Out of memory*), a pak přesunuje řádky do paměti tam, kam patří. Dělá to tak proto, aby byly řádky uloženy v paměti ve správném pořadí tj. řádek 10 před řádkem 20 atd.

Příklad:

Dejme tomu, že máme v paměti následující program:

```
10 CLS
30 FOR J=0 TO 500: PRINT "POCITAC ": NEXT J
```

Nyní použijeme příkaz MERGE a přihrajeme tento program:

```
5 BORDER 0: PAPER 0: INK 9
20 BEEP 1,1
```

Počítač si tento program uloží do volné paměti a potom přesune řádek 5 před řádek 10 a řádek 20 mezi řádky 10 a 30. Protože jsou oba programy krátké, je počítač hotový okamžitě, ale u delších programů se doba přenosu prodlužuje i na minuty.

Je samozřejmé, že po přihrávání programu příkazem MERGE počítač program nespustí a na tom je právě založeno jeho využití v našem případě: Vymažeme počítač a pomocí MERGE zkusíme nahrát úvodní BASIC hry. Pokud se objeví hlášení *OK 0:1*, jsme hotovi a máme program připravený k rozkódování. V případě, že se objeví například *Out of memory* nebo *Nonsense in BASIC*, může se stát, že se BASIC nahrál a pokud není použita oblast proměnných (viz. slovníček) například pro strojový program nebo pro nějaké důležité proměnné, můžeme také pokračovat v dekódování. Pokud je nahráváný BASIC chráněn proti MERGE, nezbyvá nám, než použít jiný způsob (jak takto zabezpečit program si vysvětlíme později).

1.1.2. Break

Další snadnou metodou jak nahrát a zastavit úvodní BASIC asi všichni dobře znají. Jedná se o přerušení běžícího programu současným stisknutím kláves CAPS SHIFT a SPACE. Tato operace se nazývá BREAK a odtud pochází často používaný slangový výraz "brejknout".

Break se používá ihned po nahrání a spuštění úvodního BASICu (nezapomeňte zastavit magnetofon) a pokud se nevyíše příslušné hlášení (*BREAK - CONT repeats* nebo *BREAK into program*), musíme použít jinou metodu, protože program je proti breaku zabezpečen. Ochrana je nejčastěji uskutečňována pomocí změny hodnot systémových proměnných (viz. příloha A).

Použití breaku sebou přináší tu nevýhodu, že program, který chceme přerušit, se již spustil a mohl tedy pozměnit hodnoty různých systémových proměnných barev nebo mohl zmodifikovat i vlastní BASICový program. I když se nám breakem podaří program zastavit (což se nám většinou nepodaří), dá nám pak práci dostat program do požadovaného tvaru (není pravidlem - především u starších programů).

1.1.3. Podvrh hlavičky

Třetí možnost je použitelná pouze v případě, že program nemá uložena nějaká data v oblasti proměnných, to znamená, že nevdává, když se tato oblast smaže nebo nepoužije (viz. kapitola 1.1.1. po výpisu hlášení *Nonsense in BASIC* nebo *Out of memory*). Je totiž založená na tom, že interpret (viz. slovníček) v podstatě nepozná, jestli po nahrání hlavičky nějakého programu (viz. slovníček) nahráváte i jeho tělo (tělo = hlavní část nahrávaná po vypsaní jména). On to pozná, ale až když délka uložená v hlavičce nesouhlasí s délkou následujícího těla. V případech, že program data v oblasti proměnných potřebuje, využívá se tento způsob pouze k prohlédnutí BASICového programu.

Pro využití této metody musíme mít po ruce program v BASICu, jehož délka je určitě větší než délka programu, který chceme rozkódovat. Nejedná se však o délku celkovou, nýbrž pouze o délku vlastního programu (vysvětleno později). Nyní zadáme příkaz LOAD a nahrajeme hlavičku dlouhého programu. Po nahrání vyměníme kazety a dohrajeme tělo (pozor tělo! nesmíte znovu nahrát hlavičku) programu, který si chceme prohlédnout. Program se nahraje a vypíše se chybové hlášení *Tape loading error*. Tím nám chce počítač oznámit, že délka uložená v hlavičce nesouhlasí s délkou těla. Program však nesmaže a my si ho tudíž můžeme prohlédnout. V případě, že k tomu nedojde, tak jsme někde udělali chybu (například znova nahráli hlavičku nebo vzali hlavičku od příliš krátkého programu), anebo má program nějakou zvláštní strukturu. S tím jsem se zatím nesetkal, takže to považuji za velice nepravděpodobné.

Co se týče délky programu, ze kterého použijeme hlavičku:

Ve hlavičce programu (viz. slovníček) je mimo jeho názvu uložena také jeho typ, ale hlavně jeho délka. U programů typu BASIC je zde uložena délka programu bez proměnných a délka programu s proměnnými. Počítač si po nahrání zjednodušeně řečeno kód rozdělí podle těchto délek a tak s ním pracuje. Proto pokud by byla délka bez proměnných programu, ze kterého použijeme hlavičku, kratší než délka bez proměnných programu, který chceme rozkódovat, dojde k tomu, že počítač bude část BASICu tohoto programu považovat za proměnné, čímž o něho přijdeme.

Podvrh hlavičky se nezdívka používá při nahrávání SCREENŮ (tj. obrázků), které nemají hlavičku, ale pouze tělo. Pokud takovéto obrázky chceme nahrát například do Art Studia, musíme použít hlavičku od jiného obrázku. Tento postup je velice častý, protože všechny obrázky, které mají délku 6912 bajtů (délka VideoRAM - viz slovníček) mají podobnou i hlavičku a nahráný obrázek se potom nijak nepoškodí a ani se nevyíše hlášení *Tape loading error*.

1.1.4. Dohrání BASICu disassemblerem

V této kapitole si ukážeme jak lze dosáhnout toho, abychom dostali úplný program i když se při použití příkazu MERGE vypíše hlášení *Out of memory* či *Nonsense in BASIC*. Náš postup je založen na tom, že počítač sice vymaže oblast proměnných, ale systémové proměnné (pozor, rozlišujte termíny proměnná a systémová proměnná - viz. slovníček) nastaví podle hlavičky. Potom stačí nahrát nějaký disassembler a obnovit oblast proměnných tím, že příkazem LOAD nahrajeme tělo BASICového programu znovu od adresy 23755.

Příklad:

Pomocí příkazu MERGE "" nahrajeme BASIC nějaké hry. Počítač vypíše hlášení *Out of memory*. Od adresy 60000 si nahrajeme program DEVASTACE a spustíme ho příkazem RANDOMIZE USR 6e4. Nyní stiskneme klávesy pro LOAD (tuším CAPS SHIFT + J). Jako začátek zadáme 23755, jako délku například 60000 (délka může být libovolná větší - nevadí nám, když DEVASTACE zahlásí chybu na konci nahrávání) a jako LEADER bajt zadáme 255 (to je leader pro hlavní tělo programu, pro hlavičku je leader 0 - viz slovníček). Adresa 23755 je první adresa BASICového programu a je uložena v systémové proměnné PROG - může být samozřejmě i jiná, ale málokterý uživatel používá periferie, které tuto adresu mění.

Nyní máme program nahráný, můžeme se vrátit do BASICu a zjistit, zda-li naše finta podařila (někdy se samozřejmě může stát, že po návratu do BASICu dojde k nějaké nečekané reakci - například se vymaže počítač).

1.1.5. Změna hlavičky

Poslední a nejlepší metoda, která se používá, je také nejnáročnější na čas. Jedná se o změnu čísla řádku, na který se skáče po nahrání. Číslo tohoto řádku je uloženo v hlavičce a abychom ho tedy změnili, musíme nějakým způsobem hlavičku vyeditovat. Těchto způsobů je více a jedny z nejpoužívanějších vám zde předkládám:

a) Nahrajeme si program HEADER EDIT nebo jiný, který slouží k editování hlaviček. Volbou LOAD nahrajeme danou hlavičku a pomocí volby EDIT změníme číslo řádku například na 65535. To je řádek, který se nemůže v programu vyskytovat a podle toho interpret BASICu pozná, že program spustit nechceme. Potom si tuto hlavičku nějak uložíme volbou SAVE a vymažeme HEADER EDIT. Nyní nahrajeme program, který chceme rozkódovat, avšak místo původní hlavičky programu použijeme hlavičku, kterou jsme si zmodifikovali programem HEADER EDIT.

b) K modifikování hlavičky lze také použít nějaké dobré kopírovací programy, které mají zabudovanou tuto funkci. Jsou to například COPY COPY nebo GAMA COPY.

c) Samozřejmě si můžeme změnit strukturu hlavičky i sami. Pokud opět použijeme program DEVASTACE a volbu LOAD, zadáme jako start například adresu 23296 (je to jedno, potřebujeme pouze nějakou adresu, na které je místo) a délku 17. Nahrajeme hlavičku a na adresu 23296+13 a na adresu 23296+14 zadáme 255. Použijeme volbu SAVE, vložíme stejné parametry jako při LOAD a leader 0, hlavičku si uložíme. Teď už můžeme pokračovat stejně jako v bodě a).

1.2. Zadávání pouků v BASICu

Cílem této kapitoly je ukázat kdy, kam a jak je v BASICu možné zadat pouky. Zadáváním pouků ve strojovém kódu se zabývá část II.

1.2.1. Kdy je v BASICu možné zadat pouk ?

Odpověď je zcela jednoduchá. Pouk je možné zadat pouze tehdy, pokud máme BASIC rozkódovaný a lze vkládat příkazy před příkaz spouštějící hru. Může se však stát, že tento příkaz je hned za prvním příkazem LOAD a hra má blok CODE, obrázek a ještě jeden blok CODE. Potom to znamená, že první blok je loader ve strojovém kódu, který má za úkol nahrát obrázek a hlavní část a potom hru spustit. V takovém případě jsme s BASICem v koncích, protože zadávat POKE před spuštěním loaderu je zhořle zbytečné. Jestli bychom pouk před jeho spuštěním, tak by se loader (kdybychom ho náhodou nepoškodili) spustil, nahrál by obrázek a nahrál by hru, která by naše pouky přehrála. Pouk je tedy většinou možné zadat pouze pokud se hra nahrává i spouští v BASICu (pokud neuvažujeme použití strojového kódu). I zde však existuje výjimka. Je to ta možnost, že máte, například z nějakého časopisu, opsán program, který modifikuje loader tak, aby vložil poke sám až po nahrání. Většinou je to pár řádků dat plus nějaký cyklus, který je uloží na správné místo. V podstatě však existuje i tento způsob použití strojového kódu, ovšem přepsaného do dat.

Občas se v BASICovém programu můžeme setkat s tím, že program nahrává obrázek a pak hned spouští nějaký strojový program. Buďto má tento program uložený někde v BASICu anebo je loader připojen k obrázku nebo je dokonce v něm. I v tomto případě však s BASICem nic nepofídíme.

Hry mohou mít i část nahrávanou v BASICu a část ve strojovém kódu, atd., ale na všechny tyto figle si s tím, co znáte, za čas sami přijdete a nebudou vám dělat žádný problém.

1.2.2. Kam je možné zadat pouk?

Aby pouk způsobil určený efekt, musí být jednak správný a jednak musí být zadán v tom místě programu, kde je jisté, že bude uložen tam, kam patří, a nebude již ničím přepsán. Proto se pouky zadávají až před příkaz spouštějící vlastní hru. Víme, že v BASICu se strojový program spouští pomocí funkce USR, jenž vrací číslo a může se tedy pojit s těmi řídicími příkazy, které číselný tvar podporují (PRINT, RANDOMIZE, PAUSE, ale ne RETURN, CLS, atd.). Číslo, které USR vrací, je návratová hodnota registru BC (jeden z párových registrů mikroprocesoru Z-80) a právě toto číslo je předáváno řídicímu příkazu.

Příklad:

RANDOMIZE USR adresa - výsledek strojového programu uloží do systémové proměnné SEED

PRINT USR adresa - výsledek vypíše na obrazovku

LET m=USR adresa - výsledek uloží do proměnné m

Samozřejmě, že po spuštění hry se již běh programu do BASICu většinou nevrací, takže někdy můžeme v programech najít i takovéto příkazy: PAUSE USR adresa, DRAW USR adresa,1 (jednička je druhá souřadnice pro příkaz DRAW), GO TO adresa, atd.

Příkaz USR, který hledáme, je téměř vždy až na konci BASICového loaderu (pokud za ním nejsou nějaké podprogramy apod.). V případě, že bychom chtěli zjistit, zda-li je to ten pravý, můžeme před něho umístit příkaz STOP. Jestliže se program zastaví dříve než po nahrání prvního bloku, znamená to, že tento USR spouštěl například nějaký loader. Nezapomeňte však oddělat všechny pouky do systémových proměnných, které by mohly příkazu STOP zabránit v návratu do BASICu.

Je nutné upozornit, že ne vždy nám pomůže vložit pouk před poslední příkaz USR, protože ten může skákat na nějaký podprogram, který nejdříve přenesení celou hru v paměti a teprve potom ji spustí. To znamená, že my jsme pouk uložili úplně na nesprávné místo, a je velice nepravděpodobné, že nám bude fungovat (pouze v tom případě, že ten, kdo pouk našel, s tímto počítal). Podobná situace nastává i tom případě, že je hra zkomprimována a na dekomprimační rutinu se skáče ze strojového kódu (v těchto případech je nutné ovládat strojový kód). Pokud se na dekomprimační rutinu skáče z BASICu, bývají na konci úvodního BASICu dva příkazy USR (první spustí dekompresi a druhý vlastní hru). Je to většinou u novějších a přepracovaných her, které jsou "stlačeny" tzv. kompresovými programy a které co nejvíce zkracují jejich délku, aniž by je nějak změnily nebo poškodily (viz. slovníček).

Často se také setkáme se zkomprimovanými obrázky, které se nejčastěji spouští sledem příkazů jako:

LOAD " CODE 4E4: RANDOMIZE USR 4E4

Příkaz LOAD nahraje zkomprimovaný obrázek na jehož začátku je rutina, která ho dekomprimuje a přesune ho do VideoRAM. Tato rutina se spouští následujícím příkazem USR.

1.2.3. Jak zadávat pouky

Pouky se zadávají normálně ve tvaru **POKE adresa, hodnota** a to buďto před USR spouštějící hru a nebo z dialogového řádku v případě, že již máte celou hru správně v paměti (například pokud jste před správným USR zadali příkaz STOP a ten se po nahrání celé hry vrátil do BASICu).

1.2.4. Rozkódování BASICu

Kódování BASICu je obvyklý jev, který velice ztěžuje vkládání pouků. Programátoři mezi sebou soutěží, kdo vymyslí složitější ochranu, a tou pak vybaví každou hru, na kterou narazí, aniž by zapomněli připsat text typu ORIGINAL FROM XXXSOFT nebo MODIFIED BY XXXSOFT (možná bych neměl používat slova programátoři, protože ne každý takovýto šourálek již něco umí). Ubohému uživateli, který si potom chce zahrát s nekonečnými životy, nezbyvá než prostudovat kdejakou příručku, aby tyto ochrany odstranil. Neříkám tak ani tak, protože jsem samozřejmě také provozoval tuto činnost, i když jsem hry aspoň vybavoval možnostmi volby pouků.

Kapitola obsahuje několik podkapitolek, které vás seznámí s nepoužívanějšími způsoby kódování a postupy, jak je dekódovat. Metody jsem se opět snažil seřadit podle složitosti od lehkých k nejtěžším. Nezapomeňte, že u většiny programů bude použito více řídlů najednou, a proto nejdějte a nebo naopak nezapomejte, když se vám podaří jeden odstranit.

1.2.4.1. Použití barev

První způsob je velice jednoduchý, ale spousta začátečníků zde narazí. Pokud program zastavíte breakem, program si stihne nastavit barvu podkladu i papíru na stejnou hodnotu a program pak není vidět. Jak si poradit, je nabílední. Změníme barvu inkoustu nebo papíru a stiskneme ENTER, aby se přepsala obrazovka s novou barvou.

Další metodou, jak schovat program pomocí barev, je využití možnosti měnit barvu pouze v BASICovém řádku. Aby se v řádku změnila barva, musí se do něho umístit příslušný řídicí kód pro danou barvu (16-21) a za něho číslo barvy. Tohoto dosáhneme stisknutím příslušných kláves: Barvy papíru můžete zadávat když stisknete EXTEND MODE (CS+SS najednou - kurzor je blikající E) a potom číslici, která odpovídá příslušné barvě - například EXT 0 nastaví černý papír. Barvu inkoustu můžete zadávat obdobně, navíc však musíte při zadávání barvy stisknout ještě CAPS SHIFT - EXT CS+7 nastaví bílý inkoust. Ještě nám zbývají klávesy 8 a 9, ty neodpovídají žádné barvě, a tak jsou použity pro nastavení nebo zrušení jasu (BRIGHT) nebo blikání (FLASH).

V programech jsou často schovávány řádky právě tímto trikem, a proto je vždy dobré se přesvědčit, jestli náhodou v programu není nějaká jeho část (například řádek) schovaná. Barva lze odstranit jednoduše použitím DELETE. Pokud program po příkazu LIST vypisuje hlášení *Invalid Colour* a při pohybu na řádku, kterého se toto hlášení týká, se ozývá varovný signál. Znamená to, že za řídicím kódem pro barvu je číslo barvy, kterou nelze zobrazit. Takovou barvu lze vymazat také pomocí DELETE, avšak někdy je těžké dojet s kurzorem až k danému místu, protože počítač nenesitelně a dlouze pípá. Doporučuji proto změnit systémovou proměnnou RASP, která udává délku varovného signálu, docílíte toho jednoduše příkazem POKE 23608, 0.

1.2.4.2. Nastavení RAMTOP

Jistě má každý doma alespoň jednu hru firmy ULTIMATE. Všechny hry této firmy mají (pokud je někdo neupravil) BASICový loader, skládající se z jednoho jediného řádku, na jehož konci je příkaz na spuštění hry. Pokud hru zastavíme breakem, nelze tento řádek vyeditovat. Je to tím, že program nastavuje RAMTOP na příliš nízkou adresu, proto se zmenší velikost editační paměti a tak dlouhý řádek se do ní prostě nevejde. Změníme tedy RAMTOP o trochu výš (je jedno na jakou adresu, protože po spuštění si program svůj RAMTOP zase nastaví zpět) a můžeme editovat. Je velice pravděpodobné, že po vložení pouků a spuštění programu bude počítač hlásit nedostatek paměti. Je to proto, že pouky příliš prodloužily program a ten se nyní pod RAMTOP nevejde. Tudíž je lepší, když hned při vkládání pouků vymažeme z řádku všelijaké zbytečné příkazy jako BEEP nebo PRINT.

1.2.4.3. RANDOMIZE USR 0 ?

Dost často se můžeme setkat s hrami, které jsou spouštěny příkazy jako RANDOMIZE USR 0 nebo PRINT USR 1. Hry samozřejmě nelze spustit od těchto adres a to znamená, že počítač musí zobrazovat jiná čísla než používá. Tato možnost je způsobena formou ukládání čísel v BASICu do paměti. Abychom lépe pochopili, jak to dělá, vysvětlíme si nejdříve, jak se ukládá řádek. Počítač si pro každý BASICový řádek ukládá do paměti jeho číslo, délku, vlastní obsah a označení konce. Tyto informace jsou v paměti uloženy následovně:

2 bajty	2 bajty	? bajtů	1 bajt
číslo řádku	délka řádku	obsah řádku	Enter

Víme, že číslo řádku v BASICu může být v rozmezí mezi 1-9999. To znamená, že počítač si musí jeho číslo ukládat do dvou bajtů (v jednom bajtu mohou být hodnoty od 0-255). Dělá to tak, že čísla, která jsou větší než 255, si rozkládá na tzv. vyšší a nižší bajt a ty si pak uloží v tomto pořadí do paměti (pozor, u délky programu je to naopak). Znamená to, že například číslo 1000 si rozloží na 3 (vyšší bajt) a 232 (nižší bajt) a v tomto pořadí je uloží. Číslo řádku potom vypočítáme tak, že přičteme 256-násobek čísla 3 k číslu 232 (256 x 3 + 232 = 1000), neboli:

$$\text{řádek} = \text{vyšší bajt} \times 256 + \text{nižší bajt}$$

Délka řádku

Délku řádku si počítač také rozkládá na vyšší a nižší bajt, ovšem do paměti je ukládá naopak, to znamená nejprve nižší bajt a potom vyšší. Délku řádku určuje její *obsah + znak enter*.

Obsah řádku

V této části jsou uloženy ASCII kódy (viz. slovníček) všech znaků a příkazů, které vidíme na obrazovce a ostatní důležité údaje.

Enter

Řádek je vždy ukončen bajtem s hodnotou 13 (enter).

Jak tedy dochází k tomu, že počítač používá jiná čísla než vypisuje? Aby byla práce s čísly rychlejší, převádí si je to tzv. floating point tvaru (dále jen FP). FP je umístěn vždy hned za číslem a bajtem s hodnotou 14, který označuje jeho začátek. Jeho délka v paměti je 5 bajtů. Počítač potom neber ohled na to jak vypadá číslo ve znakovém tvaru a používá pouze FP. To znamená, že my si můžeme pomocí příkazu POKE změnit znakové vyjádření čísla, aniž bychom změnil jeho hodnotu. Aby se dala jednoduše vypočítat hodnota takového čísla, musí to být celé číslo v rozsahu 0-65535 (což většinou u her je, protože žádný program nemůžeme spustit od jiné adresy než 0-65535). Potom je první, druhá a poslední pozice FP nulová a velikost vypočítáme podobně jako délku řádku. Na třetí pozici od čtrnáctky se nachází nižší bajt a na čtvrté bajt vyšší - viz. příklad:

(Pokračování příště)



ERNIE™

NEBEZPEČNÝ ŠILENEC

"VSTUPUJE DO JINÉHO SVĚTA" by Pils

NO NE! ERNIE NAVŠTÍVIL MÍSTNÍ
VÝROČNÍ POČÍTAČOVÝ TRH!



DÁMY A PĀNOVĚ, VÍTEJTE
VE SVĚTĚ VIRTUÁLNÍ
REALITY!



VYZKOUŠEJTE SI TEN NEJ-
MODERNĚJŠÍ ŠPIČKOVÝ HARDWARE!
DO VIRTUÁLNÍ REALITY!

VIRTUÁLNÍ REALITA JE TAK
REALNÁ, ŽE SI BUDETE MYSLET,
ŽE JSTE SKUTEČNĚ VNÍ!



AACH! NE! MONSTRA! DEJTE
TO PŘEČ! AACH!



TEBĚ POTŘEBUJEME JEDNOHO
DOBROVOLNÍKA! AH! VY, PANE!

VIRTUÁLNÍ



STAČÍ SI JENOM NĀSADIT TUHLE
HELMU A JSTE TAM!!!



NO NENÍ SNAD SKVĚLEJ?
PODEJTE MU NĚKDO RUKU!

HMM! NEVYPADÁ TO, ŽE
BYCH JE MOH! TREFIT...!



HA HA HA HA
HA HA HA
HA HA HA
HA



A - TADY. ZDA SE MI,
ŽE JSEM NĚKO ZASAHI!

TEĎ UŽ NEJDE
TREFIT VŐBEC NIC...



...ALE TŘEBA ZĀBERE TOHLE!!!



ACH! NE!



HMM... ZATRAČENÍ VĚTRĚLCI!
POŘÁD PŘEDĚ MNOU UHĀBAJ!

VYPNETE
NĚKDO TEN
KRAN!

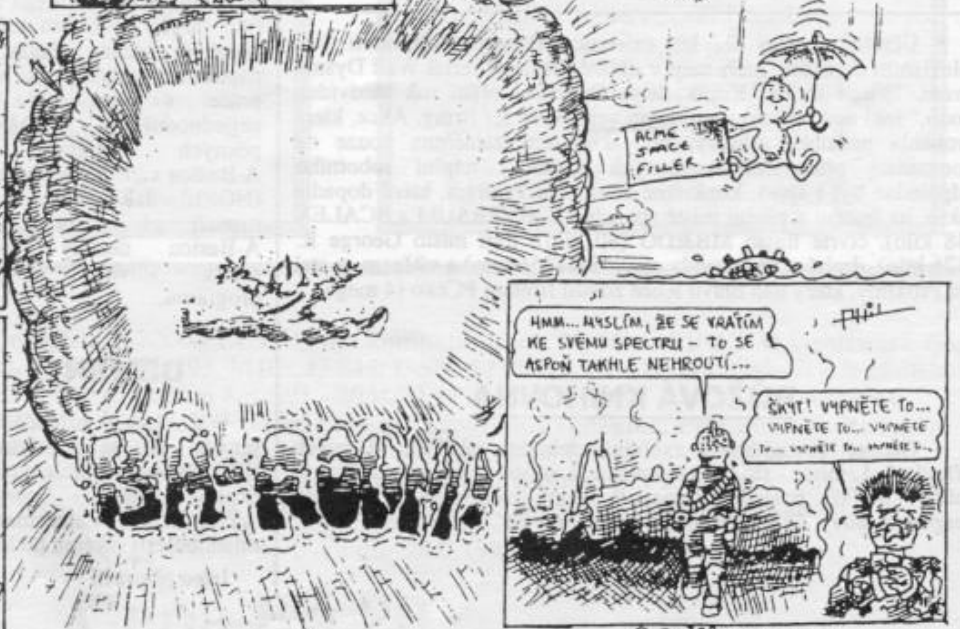


GRRR! NO TAK POJĚTE, VY
DĚVČANKY! MONSTRA!



UVIDĪME, JAK BUDOU
UHĀBAT PŘED TĀMhle... TAK!

NEEEEEEE!!!



HMM... MYSLĪM, ŽE SE VRÁTĪM
KE SVĚTU SPECTRI - TO SE
ASPOŇ TAKHLE NEHROUTI...

ŠKYT! VYPNETE TO...
VYPNETE TO... VYPNETE
TO... VYPNETE TO... VYPNETE TO...



Počítače a humor • Zajímavosti • Zprávy z domova a ze světa

ČERNÁ KRONIKA

• Jistý M.B. z M.B. (pouze shoda okolností) formátoval disketu na 43 stop a utrl si hlavičku (samozřejmě u mechaniky), následkem toho pak musel mechaniku poslat na návštěvu do Skalice a tři (a možná i více) týdny byl bez ní!

• V polovině června se na neutrální půdě v Praze sešli brněnský hudebník Qjeta (MQM Team), mladoboleslavský hudebník Scalex (Depeche Code) a ústecký programátor George K. (Depeche Code). Pro všechny případy byla přítomna i ochranka "depešáků" (Alex Cossac, A. R. C.), ale k žádnému původně plánovanému incidentu nedošlo. Cílem setkání bylo navrhnout koncepci nového programu na tvorbu AY-hudby, který by měl řešit všechny nedostatky Bzykova Soundcrackeru. Na počest svých dvou nejoblíbenějších AY-hudebníků Scalexe a Qjety, George K. pojmenoval program "SQ-Tracker" namísto původně zamýšleného Soundcracker. Schůzka byla slavnostně zakončena návštěvou B-clubu, kde se občas používá Scalexova hudba na vyhánění hostů po zavrací hodině.

• Haberman se prozatím rozhodl opustit svůj Totální výkal a chce na SHARPa předělat nějakou hru ze ZXS. Odborníci naštěstí nepředpokládají, že by se mu to podařilo častěji než 2x ročně.



• Účastníci oslav 3. let existence Proximy připravili šok televizním divákům, kteří mají v oblíbené kreslený seriál Walt Dyesni uvádí. "Vloni to byl Králík, letos Prasátko, příští rok medvídek Pooh," řekl nad talířem v pečeném vepřovém šéfu firmy. Akce, která proběhla nedaleko Jihlavy, nebyla ovšem zaměřena pouze na konzumaci postaviček kreslených filmů - náplní sobotního odpoledne byl i sport, konkrétně soutěž ve vzpírání, která dopadla takto: na šestém a pátém místě společně UNIVERSUM a SCALEX (48 kilo), čtvrté místo MB&DG (80 kilo), třetí místo George K. (176 kilo), druhé místo Freddy alias JSH (256 kilo) a vítězem se stal šéf Proximy, který nad hlavu lehce zdvihl firemní PCčko (4 mega).

RŮŽOVÁ KNIHOVNA

• George K. nedávno zveřejnil žebříček oblíbenosti svých přítelkyň: 1. místo - stodvacetosmička; 2. místo - čtyřicetosmička; 3. místo - M (viz. poslední strana). Posledně jmenovaná na žebříček reagovala slovy, že jeho autorovi asi už brzy dá ...co proto.

CO SE PŘIPRAVUJE...

• Většina Proximáků a jejich blízkých se připravuje na dovolenou. Místa odpočinkového pobytu jsou pečlivě utajována, protože zejména programátoři se bojí, aby je ve vytoužených dnech klidu neobjevily davy všudepřítomných obdivovatelek, které by od nich očekávaly na večer nějaký ten program.

• UNIVERSUM zahájil práci na tabulkovém procesoru (zatím beze jména). Jisté je jenom to, že program bude v diskové i páskové verzi a možná ještě letos.

• Práce na novém demu od Depeche Code byla prozatím odložena a Scalex čeká, až bude hotový SQ-Tracker, aby v něm mohl udělat hafo hudby. Demo se bude pravděpodobně jmenovat Nothing!, Estas Muerto nebo SQ-demo, ale také není vyloučeno, že třeba nějak úplně jinak.



NESOUTĚŽ!

Znovu je tu NESOUTĚŽ! a znovu je tu program od J. Flašky. Pokud se Vám jeho minulý program zdál geniální a předminulý téměř geniální, nevíme, co budete říkat o tomhle! Věřili byste, že pár následujících řádků v sobě ukrývá celý programovací jazyk?! Je to skutečně tak - opište a vyzkoušejte:

```
10 RESTORE 100: CLEAR 64999
20 FOR x=65e3 TO 65064
30 READ a: POKE x,a
40 NEXT x
50 RANDOMIZE USR 65e3
100 DATA 62,7,205,155,34,50
101 DATA 141,92,205,107,13,213
102 DATA 17,36,254,1,5,0,205,2
103 DATA 254,17,25,254,14,11
104 DATA 205,60,32,205,142,2
105 DATA 28,40,250,84,253,94
106 DATA 255,46,200,205,181,3
107 DATA 65,118,16,253,201,22
108 DATA 1,0,81,85,73,84,32,18
109 DATA 1,76,18,1,75,18,0
```

Sám autor nám k programu řekl: "A-Basic (absolute basic) vznikl po dlouhém studiu Mega basicu a Beta basicu. Přišel jsem na to, že tyto jazyky jsou pro neobohého uživatele příliš složité. A-Basic obsahuje všechny příkazy, které by i úplný amatér měl zvládnout. Je napsán tak, aby byl co nejkratší a přitom uživatelsky příjemný a práce s ním byla co nejjednodušší. Mnoho pěkných programů v A-Basicu vám přeje Matlasoft (NOP)." Jak autor dodal, napsali už s bratrem v A-Basicu desítky (podle bratra stovky) dobrých programů.

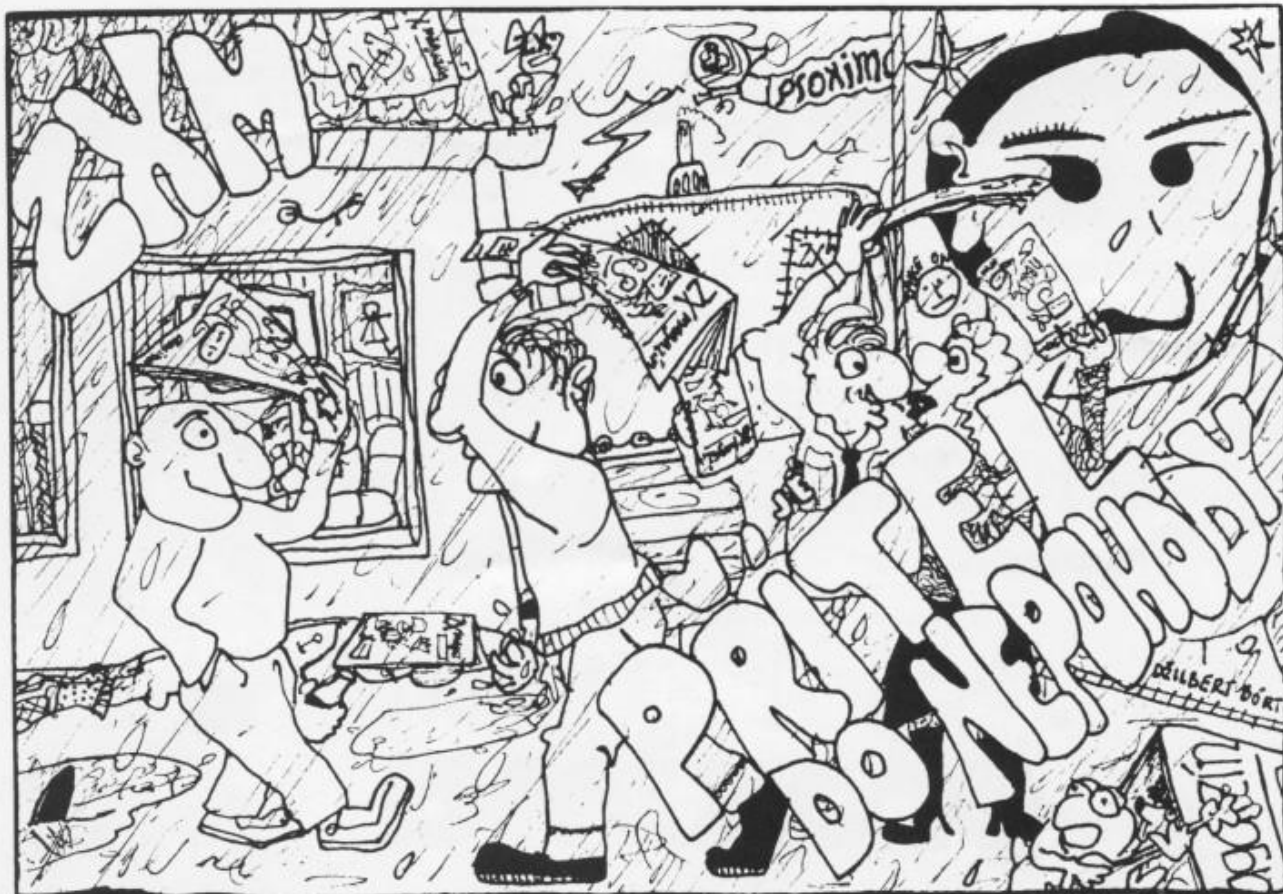


DEMENTI

• V žádném případě není pravda, že se zaměstnanci B-klubu pokusili kolem páté hodiny ranní předčasně ukončit setkání Scalex - Qjeta - George K. puštěním kazety se Scalexovou AY-hudbou. Zaměstnanci B-klubu chtěli pouze udělat mladoboleslavskému hudebníkovi radost.

Intro připravil

George K.



Ve kterých prodejnách můžete zakoupit programy od PROXIMY?

PROXIMA, obchodní dům Labe, Ústí nad Labem, Klub elektroniky 602, Martinská 5, Praha 1, DIDAKTIK market, Gorkého 4, Skalica na Slovensku, PRECISOFT v. o. s., Ulrichovo nám. 810, Hradec Králové, BONO s. r. o., obchodní dům Dargov, Štúrova 1, Košice, BONO s. r. o., Hlavní 134, Prešov, EL - COM, Budova pošty 12, Košice, RAMAT electronic, kulturní dům ODRA, Ostrava - Výškovice, ELEKTROSERVIS Kocman, SNP 1443, Považská Bystrica, ALFA, 28 října 243, Ostrava - Mariánské hory, OMEGA, Radniční 27, Hranice, ELEKTRO Zručký, Kobylí 221, PSČ 691 10, CONSUL, Pálenická 28, Plzeň, JINTES, Husova 45, České Budějovice, SHINY STAR, Kostelní 163, Český Krumlov, NALIM, Voříškova 4/22, Brno, Lampárna ECU, nám. 3. května 13, Otrokovice.

Prodejcem těchto programů (dealerem) se může stát jakákoli firma po podepsání dealerské smlouvy. Rabat činí 10 až 30%, dle odběru. Podrobnosti zašleme na požádání.

PROXIMA, box 24, 400 21 Ústí nad Labem.

Chcete, aby měl ZX magazín více stránek?

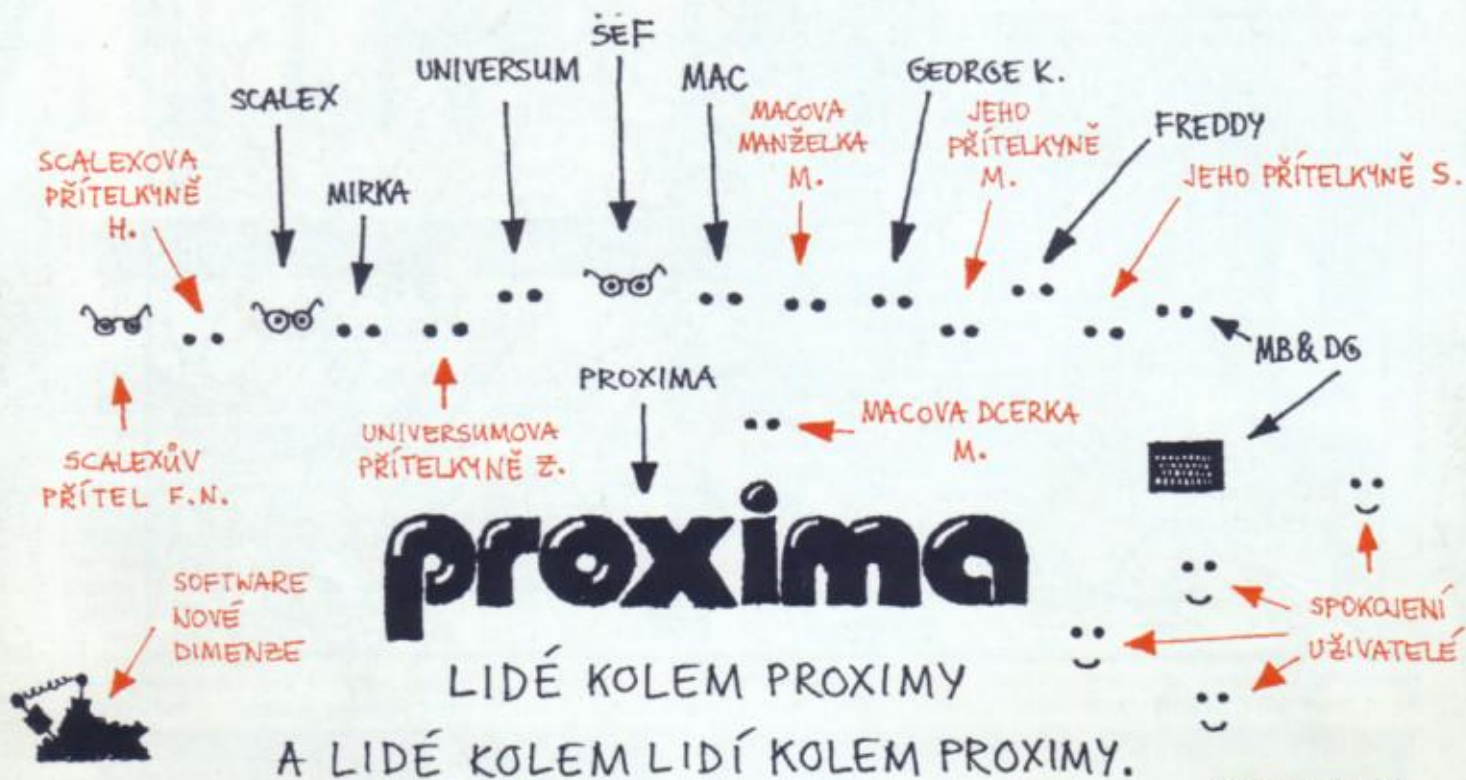
K tomu je třeba, abychom získali co nejvíce předplatitelů a mohli tak zvýšit náklad časopisu. Pomozte nám proto: rekněte o existenci ZX magazínu svým kamarádům a známým, nebo přesvědčte majitele obchodu s elektronikou a doplňky k počítačům DIDAKTIK ve Vašem městě, aby ZX magazín objednal do prodeje!

ZX magazín, časopis pro uživatele počítačů ZX-SPECTRUM a kompatibilních. Vydává: PROXIMA-software v Ústí nad Labem. Povoleno pod číslem MK ČR 5293. MÍČ: 47 845. ISSN: 1210-4833. Podávání novinových zásilek povoleno Oblastní správou pošt Ústí nad Labem č. j. P/1 - 2034/92 ze dne 24. 9. 1992. Adresa pro veškerou korespondenci: PROXIMA, box 24, pošta 2, 400 21 Ústí nad Labem. Odpovědný redaktor: Petr Podařil. Redakční rada: UNIVERSUM, GEORGE K., Oldřich Páleníček. Obrázky kreslí Miloš Bílek. Za původnost a obsah příspěvků ručí autor. Nevyžádané příspěvky se nevracejí. Distribuce: PNS. Předplatné: PROXIMA, box 24, pošta 2, 400 21 Ústí nad Labem. V roce 1993 vyjde celkem 6 čísel, min. rozsah každého čísla je 32 stran.

Toto číslo bylo dáno do tisku dne 28.7.1993

Cena Kč 20,-

3 ROKY...



PROXIMA-software v. o. s.

post box 24

400 21 Ústí nad Labem

