

ZX magazín

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

4/93

Z obsahu čísla:

Popisy her: Back to Skool, Sito Pons, WNB Game, Advanced Flight Trainer,
Navy Seals — Hudební editory pro Spectrum 48 — Tabulkové procesory —
Něco pro AV... — Robotika — Hledání nejkratší cesty (trocha teorie) —
Nekonečné trápení s nekonečnými životy (2) — Animace — George K.
vzpomíná — Horror — Komiks — Intro (počítače a humor)

20 Kč

KLÍČOVÁ
ZÁLEŽITOST

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 poukážete **pětidílnou** složenkou typu A (poku ji neseženete, můžete si ji vyžádat u naší firmy), na konto číslo 28846-411/0100, Komerční banka v Ústí nad Labem. Jeden ze dvou dílů složenky, které Vám na poště po zaplacení dají, nalepte na objednávací kupón (můžete jej pořídít zkopírováním tohoto listu) a 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 objednáváte, 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 ve stáncích PNS a také ušetříte 2 Kč na každém čísle!

Místo pro nalepení
kontrolního
ústrižku složenky

Objednávám předplatné ZX magazínu,
ročník 1993, od čísla _____

(pokud neuvedete, od kterého čísla chcete předplatit ZXM,
dostanete jej od nejbližšího čísla které vyjde)

Adresa odesílatele:

PSČ:

Adresa příjemce:
PROXIMA, box 24, 400 21 Ústí n/Lab.

Vážení čtenáři,

tak už jsem tu zase a mám hned několik novinek.

Ta první je, že mi spadla na hlavu větší část práce s přípravou ZX Magazínu - doufejme, že to bude změna k lepšímu - hlavně pravidelnost vycházení časopisu musíme podstatně zlepšit.

Další souvisí s mým záměrem zavzpomínat na to, co jsem si odnesl ze školy a o zajímavé záležitosti se s Vámi podělit - ale zbytečně se netěšte, budou to jen informace (nic jiného jsem si tam odtud samozřejmě neodnesl, ale hlavně by mi z toho vůbec nic nezbylo).

Také bych Vám chtěl představit Proximu. V současné době má veškerou činnost, kterou Proxima vyvíjí v oblasti 8-mi bitových počítačů (distribuce programů pro Sinclair, Didaktik a kompatibilní, vydávání ZX Magazínu), na starost celkem 5 lidí:

Na přípravě ZX Magazínu pracují obvykle tři lidé - Petr Dodařil, já a externě nám pomáhá nám George K., obrázky pro nás kreslí Miloš Bílek.

Na distribuci programů pro ZX a kompatibilní počítače (kopírování disket, reklamace, dotazy a dobírky) pracují též tři lidé.

Na přípravě distribuce pracují čtyři lidé.

Jak vidíte, máme každý na starost spoustu věcí a někdy toho je až nad hlavu...

Pak bych si chtěl trošku postěžovat na poštu - občas nám nějaká Vaše zásilka nebo objednávka nedoručí (při tom množství se tomu nelze příliš divit) - pak se stává, že si stěžujete a urgujete, občas dokonce ne příliš vybiravým způsobem. Nedávno se nám například stalo, že přišla už druhá urgence ZX Magazínu, a my jsme dotyčného předplatitele vůbec neměli v databázi (což znamená, že nám nepřišla objednávka spolu s ústřížkem zaplacené složenkou), ale ani jeho první urgenci jsme nedostali. Podobně jsme měli nedávno už třetí objednávku na programy od jednoho uživatele (první poslal v květnu, další koncem června a poslední přišla na začátku srpna), mezi vyřízenými objednávkami (ty máme všechny v

počítači) ani mezi dosud nevyřízenými objednávkami nic nebylo - možná si myslíte, že tu musíme mít pěkný bordel - ale objednávky se vyřizují z došlé pošty a nejpozději do týdne vkládají do počítače, ten vytiskne dodací list se složenkou a jde to do dobírkového oddělení (Mirka), kde se to zabalí a odnese na poštu. Možnost, že by se objednávka ztratila u nás sice není stoprocentně vyloučena, ale je mizivá - jistě chápete, že nemáme nejmenší zájem, ztrácet objednávky (zákazníky).

Nedávno jsem četl v Lidových Novinách (ze soboty 7. srpna, na poslední straně) o poště, že v průměru 90% zásilek dojde včas a zbylých 10% se zpožděním - o tom, kolik zásilek nedoručí nikdy, se tam nepsalo, ale zjevně to nebude zrovna 0%. Takže, když k něčemu takovému dojde, nemyslete si o nás hned, že Vás chceme okrást a napište nám, že jste si

předplatili (v tom případě pošlete raději rovnou kopii ústřížku složenkou, řešení se tím značně urychlí) nebo objednali to a to a my zjednáme nápravu - ještě jednou: Vaše spokojenost nezáleží jenom na nás, bohužel.

Některé problémy si ovšem způsobujete sami - když například dostaneme předplatné, kde chybí ulice a číslo popisné (jako např. u pana Václava Valenty ze Štětí, pokud ho znáte, řekněte mu, ať napíše), nenaděláme nic ani při nejlepším vůli. Také se nám stalo, že pošta zásilku dvakrát vrátila jako nedoručitelnou - opět nemůžeme než čekat, až se postižený ozve.

Berte předchozí řádky jako takové postesknutí si, protože jsou to problémy, které nás pálí.

- Univerzoom -

Obsah čísla 4/93

Vážení čtenáři	1
Back to skool - manuál ke hře	2
Sito Pons 500cc - Grand Prix - manuál ke hře	3
Whole New Ball Game - manuál ke hře	4
Advanced Flight Trainer 128 - manuál ke hře	5
Navy Seals 128 - manuál ke hře	6
George K. vzpomíná	8
Kluby	10
Animace 2 (4)	11
Basicovská hra	12
Listárna	13
Hudební editory pro Spectrum 48KB	14
Tabulkové procesory	16
Jak na zvukový interface s obvodem AY (1)	18
Ovládání D40	19
Máte rádi vlaky? (aneb hledání nejkratší cesty)	20
Softwarové využití druhé banky u Didaktiku GAMA	22
Nekonečné trápení s nekonečnými životy (2)	23
Strojový kód Z-80	26
Komiks	27
Softwarové novinky	28
ZX Magazín a robotika (7)	29
UNIVERSUMální klávesnice pro „gumáka“	30
Jedna poněkud temná historie (děsivá, ale pravdivá)	31
Intro	32

BACK TO SKOOL

Pokud se nemýlím, tak už jsme se setkali u návodu na starou, ale skvělou hru **SKOOL DAZE** (pokud jste jej nečetli, tak to rychle učiňte, protože některé informace, uvedené posledně, nebudu opakovat, zejména ohledně ovládání, protože to zůstalo nachlup stejné, pouze něco přibýlo) a dnes si povíme něco o jejím druhém dílu, který nese název **BACK TO SKOOL** - jde vlastně o slovní hříčky (v obou případech), protože fonetický přepis "SKOOL DAZE" je naprosto stejný jako "SKOOL DAYS" (přeložte si to) a "BACK TO SKOOL" se dá číst jako "BAG TO SKOOL" (přeložte si to). Navíc je v obou názvech gramatická chyba (najděte si ji). *Nutno dodat, že MB&DG, který tento článek jazykově lektoril, si oba názvy přeložil a chybu našel.*

Poznámka korektora: v minulém návodu (k SKOOL DAZE) jsem jako poznámku vysvětlil jména obou her, autor o tom nevěděl a tudíž netrpí ani sklerózou ani tím, že by se rád četl - Universum.

Stručný obsah minulého dílu a úvod do dílu nového:

Pro neukázněné chování a poškozování cti školy byl **ERIC** potrestán úkolem napsat něco přes 10 000 řádků s textem, který nám bohužel autoři hry nesdělili (ale jistě to nebylo jen nějaké nenápadité "Nebudu ve škole zlobit." - jako člověk s jednoletou pedagogickou praxí v oboru základního školství mohu potvrdit, že tak stupidní zvláštní úkoly už dávno vyšly z módy, stejně jako tělesné tresty (bohužel)). **MB&DG** tvrdí, že on by **ERICa** zabil - zde je vidět, jak negativně působí počítačové hry na naši mládež. Od zabíjení tu jsou přece školníci (nikoliv však v této hře. Zahrejte si **BEERLAND** a uvidíte).

Když se **ERIC** druhý den vrátil do školy (**BACK TO SKOOL**) se svojí brašnou (**BAG TO SKOOL**) plnou střeliva do praku, plesnivých svačin a erotických magazínů (čertví, proč právě v tuto chvíli **MB&DG** zpozorněl...), čekalo ho velké překvapení. Progresivní německá stavební firma (plná levných českých a polských pracovních sil) školu zgruntu přebudovala, před ní postavila hřiště a o kus dál vyrostlo krásné nové gymnázium plné krásných (možná nových) dívek (*reakce MB&DG na tuto skutečnost snad ani nebudu popisovat*).

Leč kantoři zůstali staří a oškliví - **Mr. WACKER** - ředitel ústavu, **Mr. Creak** - starý dějinozpytec, **Mr. Withit** - geographoman a **Mr. Rockitt** - profesor fyziky, chemie a biologie (**MB&DG** si jej přejmenovává na **MENDĚLEJEVA** či **PAVLOVA**).

K personálu se přidal ještě školník **Albert**, který benevolentně přehlíží všechny vaše hříčky, a ředitelka gymnázia

Miss Take (podobnost s anglickým slůvkem **MISTAKE** je čistě náhodná), která nepřehlíží vůbec nic, zvláště nepřehlíží váš pobyt mezi dívkami (což mrzí zejména pana **MB&DG**).

Ani vaši vrstevníci se příliš nezměnili. **Boy Wander** se s vámi nakonec spřátelil, a proto za jeho poznámky žádné bodky nedostáváte. **Angelface** stále boxuje, navíc od minula dost zesurověl a mlátí kolem sebe vlastně permanentně. **Einstein** je stále velký šprt a především bonzák. Bonzuje mnohem více než v prvním díle. A i zde nám přibyla jedna postava - slečna **Hailey** (**MB&DG** ji přejmenovává na **MARFUSU**) - **ERICova** velká láska. Navštěvuje dívčí gymnázium a pokud se k ní **ERIC** chová něžně (klávesou **K** (kiss-polibek)), dokonce mu pomáhá s psaním trestů (ubýdou **LINES**). Ale má to malý háček.

Krásná **Hailey** (alespoň doufám, že je krásná. Oni jsou ti sprajitci mrňaví...) má určité meze - políbíte ji šestkrát a potom už se neopovažujete ji olizovat, protože se při hodinách tělesné výchovy naučila od **Miss Take** několik antiznásilňovacích chvatů a vyzkouší je právě na Vás. A pak už si u ní neškrtnete. Takže si nevyplývejte polibky hned na začátku a šetřete si je, až těch **LINES** bude opravdu dost. (Je to svinstvo, takhle vypočítavě zacházet s něžným stvořením.) **MB&DG** má tradičně odlišný názor - tvrdí, že **Hailey** už má dost toho věčného ocucávání a chtěla by, aby (cenzurováno - toto není erotický magazín) a vynucuje si to fyzickým napadáním (ve skutečnosti sadismus - pozn. **SCALEX**).

Podstata hry se vlastně nezměnila, pouze přibyla spousta nových věcí (viz dále), textů (viz dále) a tak všelijak podobně. Stejně jako v prvním díle, i zde se snažíte všechny své lumpárny svádět na **Angelfacea** a **Einsteina**. Hlavně nic nesvádějte na chudinku **Hailey**, pak byste totiž její trest stejně slízlí vy - a vůbec, vy byste jí to udělali??? (**MB&DG** tvrdí, že on by jí to udělal určitě...)

Několik rad, co všechno se dá ve hře dělat:

- chytit myš (**C**) a pustit v dívčím gymnáziu (**R**) - holky pak hezky hopsají
- jezdit s kolem (stojí před školou) po chodbách (až se vám to podaří, napište mi)
- dupat na kytky (vyskočíte na ně)
- stříkat vodu z pistolky (až ji seženete...)

- a ještě jedna rada: když začne **Einstein** mluvit, aniž by se učitel na něco ptal, je to určitě bonz na vás. Proto se o hodinách vyplatí sedět v poslední lavici s **Einsteinem** a v takové chvíli neustále držte **S** - **Einsteina** to bude vyhazovat z lavice,

jeho řeč se silně zpomalí a pokud zazvoní dříve, než se vykoktá, byl bonz neúspěšný (zato vy jste byl úspěšný). **MB&DG** vyhazuje z lavice každého, i když vůbec nebonzuje (inu - dva roky seděl v lavici se **SCALEXem** a leccemu se přiučil).

Ten, kdo umí anglicky a má smysl pro humor a pro horor, jistě ocení vtipné texty, které se objevují na tabulích a v bublinách učitelů. Některé vtipy vznikly vlastně neošetřením programu - učitel pracně smaže z tabule nápis **HEAVY WATER** a pak jej tam opět napíše (a pak se zeptá na něco, co nemá s těžkou vodou vůbec nic společného). Otázky učitelů a odpovědi **Einsteina** také nemají chybu (jednu za všechny: "**Kde žije veš?**", "**V chybném programu.**" - to je další slovní hříčka - programům pro ladění programů se říká **debuggery** - odvšivovače). Nápisy na tabulích vás také pobaví. Kromě toho, že tam uvidíte **Boy Wanderovy** texty (**Nenávidím školu, nemám rád holky, nesnáším Mr. Wackera** apod.) můžete narazit například na název lekce "**BACTERIA AS PETS**", neboli "**Baktérie - domácí miláček**". Pokud sledujete při hře pana **MB&DG**, můžete narazit maximálně na nápisy "**Fuck Miss Take, Fuck YOU, Fuck off, Fuck yourself**" apod., což je důsledek toho, že i vy můžete psát na tabuli (viz návod na **SKOOL DAZE**). Výrazy učitelů ("**Vy moji malí anarchisté**") rovněž patří k humorným stránkám hry; jistě vás potěší i jejich požadavek "**Napište esej na téma 'Proč miluji školu'**".

Doufám, že vás tento článek (spíše reklamní než návodný) příjemně naladil a že si co nejdříve hru **BACK TO SKOOL** zahrajete. A pokud objevíte něco nového, můžete mi napsat, rád si to přečtu a budu-li mít na poštovní i odepíšu.

- Pouze pro XZM bez DPH napsal **SCALEX** -
(with kibicing by **MB&DG**)

Nakonec ještě dvě malé poznámky, které se vztahují k popisu hry **SKOOL DAZE**:

1) Jak jste si zajisté všimli, celá škola je vyzdobena řadou erbů. Pokud se vám podaří "sestřelit" učitele právě pod některým z těchto erbů a z vhodné vzdálenosti vystřelíte znovu, střela se odrazí od hlavy nic netušícího kantora a zasáhne erb. Podaří se vám sestřelit všechny erby? (- **SCALEX** -)

2) Pokud jste si zkoušeli hru **SKOOL DAZE** zahrát a srovnávali jste obrázek v XZM 3/93 s tím, co jste viděli na obrazovce, tak vám musím sdělit, že šlo o obrázek ze hry **BACK TO SKOOL** (**SKOOL DAZE** jsme neměli k dispozici).

(- **Universum** -)

Sito Pons 500^{cc} Grand Prix

(C) Zigurat, 1991

V roce 1991 vydala španělská firma Zigurat hru Sito Pons 500^{cc} Grand Prix. Po přečtení názvu jistě zaplesá srdce každého milovníka silných strojů a jme se dotýcnou hru shánět (pokud ji už ovšem nemá - rok 1991 je pro některé z vás jistě počítačový dávnovek).



Když se mu to podaří a program konečně nahraje, narazí pravděpodobně na dvě věci - za 1) se na něho vyhrne přímo záplava španělských textů (tento "nedostatek" odstranila jistá nejmenovaná firma zabývající se mrzačením her jaksi programově; je-li ctěná libost a máte-li na to žaludek, opatřete si českou verzi...) a za 2) zjistí, že hra není jen tak pro někoho - ovládnout dokonale svěřený stroj dokáže totiž jen skutečný profesionál a my ostatní se rozdělíme na dva tábory: jedni hru znechuceně odloží do archivu (jejich škoda) a druzí se znovu a znovu pokoušejí udržet motocykl (vidá, dosud jsem neuvedl, že se bude jezdit na motocyklu - omlouvám se těm, kterým to nedošlo...) na silnici a hle - posléze se nám to také daří! První krok je učiněn, jsme na dobré cestě... leč nepřeskakujme.

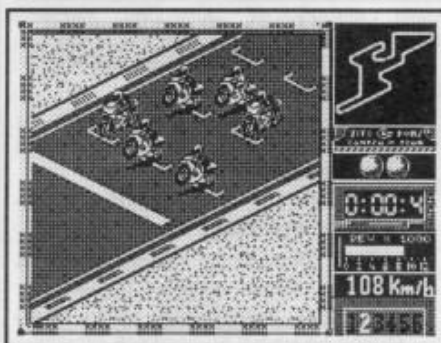
První, co na nás po nahrání vybafe, je hlavní menu. Bohužel jsem majitelem české verze a nemohu tedy uvádět originální termíny, doufám však, že následující popis pomůže i majitelům verze španělské... Volbu 0 ponechme zatím stranou - stisknutím této klávesy se spouští vlastní hra (resp. se objeví další menu - brrr!). Možnost 1 je už užitečnější, neboť nám umožní navolit si klávesy. Dejte si tedy pozor na poněkud příliš citlivé čtení klávesnice a do toho! Nejprve si zvolte klávesu pro zatáčení **vpravo**, následuje směr **vlevo**, **brzdění**, **akcelerace**, **řazení**, **pauza** a **přerušení hry**. Předvoleny jsou klávesy P, O, A, Q, SPACE, M a Z. Majitelé Kempston joysticku to mají jednodušší - těm stačí v menu stisknout klávesu 2 a mají po starostech. Jen zastavovat a přerušovat hru musí stejně z klávesnice, jako my, co už nemáme Kempston kam pchnout. Poslední položka menu, 3, nám umožňuje zvolit si **obtížnost**. K dispozici jsou čtyři možnosti, 1. a 2. se vyznačují automatickou

převodovkou a brzy zjistíte, že to není příliš velká výhoda. Takže jsme se seznámili s prvním menu, stisknete 0 a popojedem.

A máme tu další menu. Po stisku 0 se ocitáte rázem v řadách motocyklistické elity, v **mistrovství světa**, ale z vlastní zkušenosti vám doporučuji ještě posečkat. Volba 1 vám umožní se zúčastnit závodu na **jednom zvoleném okruhu**, ale ani tam bych, být vámi, prozatím nelezl, abyste si neurázili kšandu (nehledě na to, že byste se mohli i zabít, nenaučíte-li se nejprve pořádně ovládat mašinu!). V žádném případě také při prvním spuštění hry nevolte možnost 2 - tato možnost umožňuje **vstoupit do MS v jeho průběhu**, ovšem s "malým" omezením: musíte znát heslo a to se zase dozvíte jen tehdy, propracujete-li se do zvoleného stadia soutěže (viz. dále). Jakmile vás počítač jednou požádá o heslo, už vás nepustí, dokud mu ho nedáte (pokud ho ovšem nepřelstíte RESETem), proto nabízím jedno pro záchranu (opisujte pečlivě a opatrně):

CEBLABBPANAKAJAAAAAG.

Jak vidíte, stojí to heslo opravdu za to. Toto konkrétně vás přenese na start druhého závodu MS (Jerez) s plným počtem bodů (první závod jste vyhráli). Tím, k čemu celou dobu směřujeme, je položka pod číslem 3 - **trénink**. Po jejím zvolení si můžete zvolit stiskem **nuly** požadovanou trať, **jednička** vás pustí na trať a **dvojka** vás vrátí do **hlavního** (prvního - ovládání) menu (stejně se

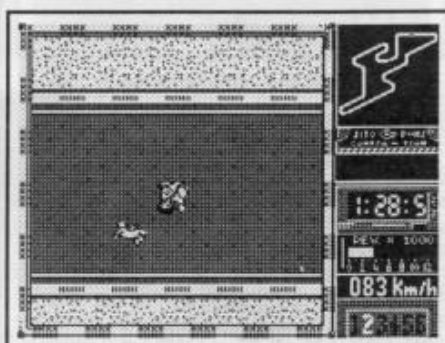


postupuje i v případě závodu na zvoleném okruhu a s malým omezením - trať je navolena a nejde ji změnit - také při MS. Zapamatujte si to, nebude se k tomu už vracet!). Ty úplně nejslabší povahy dají možná přednost volbě 4, což není nic jiného než **demo**, ale my, chlapi, už sedáme (resp. leháme) za říditka a túrujeme motor...

Zvolili jsme si tedy trénink, vybrali si trať a jdeme na to. Co vidíme? Vpravo nahoře **mapku zvoleného okruhu** a věřte mi, bude se hodit (je na ní vyznačena vaše poloha); vpravo dole pak **zařazený**

rychlostní stupeň (pokud jste zatím na nic nesahali, měli byste mít zařazený neutrální - nic vám tam nesvítí). Pak následují (směrem nahoru) **ukazatel okamžité rychlosti** (nejvíce se dá jet cca 322 km/h, nevíme ovšem kde), **sloupcový otáčkoměr**, **digitální display ukazující jak dlouho už jedete a nahoře, pod mapou, je okénko**, ve kterém se vám budou později vypisovat **různé zajímavé informace** (kdo havaroval apod.). Většinu obrazovky pak zabírá pohled na trať prostřednictvím jakoby kamery, středem jejíž pozornosti je osamělý motocyklista - vy!

Na první pohled vás jistě zaujme nádherná, byť nebarevná, grafika. To ovšem není všechno. Vidím, že se celí třesete touhou - prosím: přidejte plyn, zařaďte (zvolili-li jste obtížnost 3 nebo 4 - s "manuálem") a vyražte! Obraz se dal do pohybu. Jsem přesvědčen, že jste uchvácen



pohledem, který se vám naskytá (motocykl prudce vyrazí vpřed, zvedaje se na zadní) a nadšen realistickými zvuky, nebudu vás tedy zdržovat, snad malé upozornění: cedule s nápisy 150, 100 a 50 udávají vzdálenost k nejbližší zatáčce a jsou velmi užitečné. Až se dostatečně vyřádkáte, přerušte hru navolenou klávesou (ocitnete se v hlavním menu) a věnujte pozornost následujícímu odstavci...

Tak co, jaké to bylo? Paráda, ne? Bezcičné popojíždění vás už omrzelo a rádi byste změřili síly s někým zkušenějším? Snadná pomoc - zvolte si **závod na zvoleném okruhu** (viz. výše). Nejprve se ovšem musíte kvalifikovat, tak do toho! Malé upozornění: zatímco v tréninku jste po pádu zase nasel na motocykl a klidně pokračoval, při závodech (a kvalifikaci na ně) to není tak jednoduché! Pokud totiž při pádu jezdec (tedy vy) narazí do některé překážky, jichž je kolem trati početně, už se nezvedne a závod (ale i kvalifikace (?)) se dojde bez vás! Pokud se vám podaří ujet celé kolo (je jedno za jak dlouho), postoupil jste do vlastního závodu. Prohlédněte si pořadí jezdců na startu (Co to, vy tam nejste? Ale ano, jenomže v této hře už nejste nějakým Pepkem Novákem, ale slavným jezdcem Sito Pensem - je to ten poslední...) a až se nabažíte, zvolte si start závodu (stejně jako u kvalifikace). A zase je tu něco nového! Kromě vás je na startu dalších 8 jezdců (snadno je od vás rozpoznáte, jsou cef v černém) a než vyrazíte, musím vás opět varovat: jakákoliv, byť sebenepatnější, kolize s nimi má za následek váš neodvratný pád a

Whole New Ball Game

(C) 1988 Pete Cooke

Dnes si představíme další hru z dílny geniálního Pete Cooke (autor her ZOLYX, TAU CETI, ACADEMY, JUGGERNAUT, STUNT CAR RACER, MICRONAUT ONE, EARTHLIGHT a dalších). Tentokrát si pro nás pan Cooke připravil **úplně novou míčovou hru** - tak zní překlad názvu. Abych se nedopustil nějakých chyb, vezmu vše pěkně od začátku.

Hlavní menu

V hlavním menu máte možnost ovlivnit tyto položky:

PLAY GAME - spustí hru

START SCREEN \$\$ - o řádek níže je "odfajfkovaná" obrazovka, v níž se začne

DIFFICULTY LEVEL - tři stupně obtížnosti

SEE SCORES \$\$ - tabulka výsledků (zvlášť pro každou obtížnost)

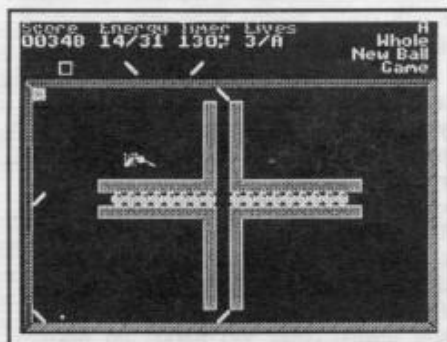
VIEW/ALTER KEYS \$ - možnost změny ovládání (S,X,O,P,SPACE)

RULES \$ - pravidla hry

GUIDE TO EFFECTORS \$ - rejstřík objektů

SCREEN DESIGNER - tvorba vlastních místností

V pravém dolním rohu je ještě jedna malá ikona - pod tou se skrývá malé překvapení, které ovšem s vlastní hrou



nemá nic společné. Jde o kreslení fraktálových stromů. Pokud vás tato matematicografická činnost zajímá, pohrajte si s tím.

tím i vaši diskvalifikaci! Během jízdy se dozvíte v okénku vpravo, kolikáté že jedete kolo, kdo je vedoucím jezdcem, jaká je vaše pozice a kdo vypadl (neradujte se, vaše jméno tam bude nejčastěji). Víte vše, do nich! Čekám na vás v následujícím odstavci...

Hotovo? Porazil jste je, všechny trati už znáte jako své boty? Výborně, přišel čas pro zlatý hřeb: **Mistovství světa!** V podstatě je to to samé jako závody na

Rules/Pravidla

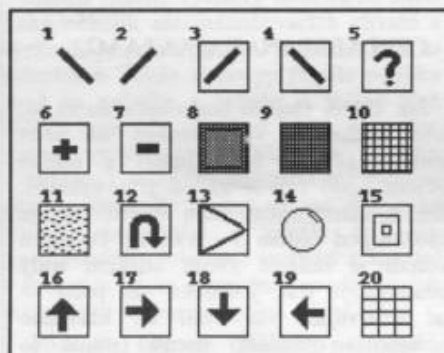
Cílem hry je sesbírání žlutých energotablet tím, že přes ně přejedete míčkem. Cestu míčku můžete řídit pokládáním a odebíráním odrazecích ploch. Každá obrazovka musí být vyčištěna od energotablet během stanoveného časového limitu. Pozor na ostatní objekty (zvané EFFECTORS), které budou měnit trajektorii a rychlost míčku.

A Guide to Effectors/ Průvodce objekty

Odrážecí plochy (1, 2). Odráží míč podle zákona o úhlu dopadu a odrazu. V průběhu hry je možné tyto plochy smazat.

Permanentní odrážecí plochy (3, 4). Není možné je smazat.

Randomizer (5). Odráží míč náhodným směrem.



Urychlovač (6) - zvýší rychlost míčku.

Zpomalovač (7) - sníží rychlost míčku.

Zeď (8). Odráží míček zpět.

Překážka 1 (9). Na jejím místě není možné stavět brány. Cestu míčku neovlivní.

Překážka 2 (10). Viz překážka 1, ale míček tuto překážku zlikviduje.

Sliz (11). Pomalu se roztéká a zabírá místo pro stavbu odrazecích ploch.

Rotátor (12). Změní směr míčku o 90 stupňů po směru hodinových ručiček.

Start (13) - z tohoto místa míček startuje.

Energotableta (14). Ty musíte sbírat.

Skoková jednotka (15). Přenese míč do jiné skokové jednotky na obrazovce.

Odrážecí šipky (16, 17, 18, 19). Odráží míček ve směru, kterým ukazují.

Past (20). Zničí při doteku Váš míček.

Během hry postupujete takto: kurzorem najedete nad hrací plochu a zvolíte si jeden ze tří objektů (dva typy odrazecích ploch nebo gumu). Tento objekt pak vhodně umístíte na hrací plochu. Prebytečné odrazecí plochy můžete "smazat" použitím gumy.

Hra umožňuje tvorbu vlastních místností (**SCREEN DESIGNER**).

Po zvolení této volby se ocitneme v módu tvorby místností. Stejně jako ve hře, i zde umísťujeme objekty na hrací plochu. Navíc jsou zde možnosti:

EDGE 1 a EDGE 2 - dva typy okrajů hrací plochy

SAVE - uložení místností na kazetu

LOAD - nahrávka místností z kazety

CLEAR - smazání hrací plochy

GET - použití některé již hotové obrazovky k úpravám

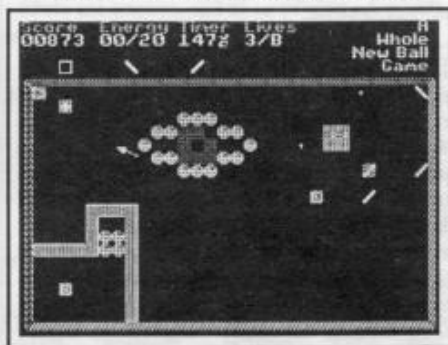
PUT - uložení vlastní obrazovky pod určité písmeno

EXIT - návrat do hlavního menu

Některé volby vyžadují potvrzení ikonou "OK", pokud volbu potvrdit nechceme, volíme "CANCEL".

Vše potřebné víte - můžete se dát do hraní. Hodně štěstí!!!

- Pouze pro ZX Magazin napsal SCALEX -



Sito Pons je hrou pro skutečné "fajnšmekry" a pokud snad ještě váháte, nebo vám hra připadá příliš těžká, nenechte se odradit - na shledanou na Phillip Islandu!

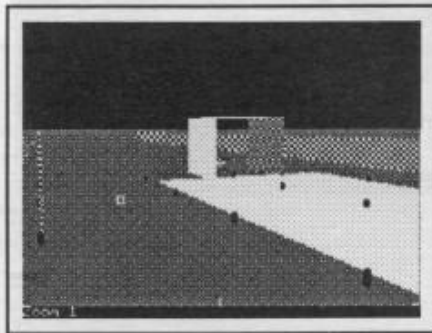
- MB&DG alias Crazy Egg -

ADVANCED FLIGHT TRAINER 128

Electronic arts 1988

I když je tato hra už trochu staršího data, k nám se dostala až po delší době a myslím si, že by se naši čtenáři s ním měli seznámit. Advanced Flight Trainer je simulátor akrobatického letu v letadle. Zde Vám přinášíme jeho stručný popis.

Po nahrání se objeví menu, které se



ovládá Q, A, O, P, SPACE (ss=submenu).

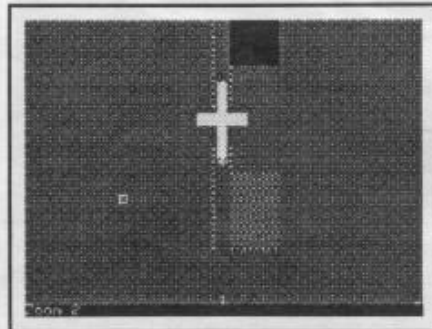
Intro Flight: Krátká ukázka akrobacie.

Test Flight: Zde si můžete otestovat Vaše akrobatické dovednosti. Po stisku SS se objeví další menu:

LOCATION - PLANE

Location - Obsahuje 16 pozic letu.

- 01 Hangar start z hangáru
- 02 Lined up přistávání na letišti z výšky 900 stop
- 03 3 000 ft přistávání z výšky 3 000 stop
- 04 10 000ft přistávání z výšky 10 000 stop
- 05 N 40 mi ocitáte se vždy na různých místech. (???)
- 06 E 40 mi "
- 07 S 40 mi "
- 08 W 40 mi "
- 09 logo lehký průlet mezi vysokou budovou a pyramidou
- 10 street let mezi řadou budov
- 11 slalom velmi těžký a krkolomný slalom mezi budovami
- 12 obstacle průlet nad překážkami



- 13 gates průlet mezi branami
- 14 S course jižním směrem k pyramidě
- 15 Square let ve čtverci x*x mil
- 16 Start up přistávání

Plane

Zde máte k dispozici 14 různých letadel: Camel, Cessna, Cherokee, F-16, F-18, P-51, Spad13, Spitfire, SR-71, X-1, X-3, XNL-16, XNL-12, XRH-4. Každé letadlo má svá specifika jako je citlivost na vychýlení výškovky, klapky, brzd, co je však zajímavé? Všechna letadla mají stejnou palubní desku! (při změně letadla se pouze změní číslování tachometru).

Airplane racing. Zde jde již o samotnou soutěž, kde soupeříte v 5 ti disciplínách se svým soupeřem (počítačem). Disciplíny:

Closed course - uzavřený okruh. Musíte proletět všechny brány. Let v F-18

Reno - Opět průlet branami s letadlem P-51

Slalom - Krkolomný slalom mezi překážkami. Spad 13

Straight - rovný, přímý let. P-51

2 mile box - Let ve čtverci 2 krát 2 míle...

Formation flying - Další velmi zajímavou částí je létání ve formaci. Před Vámi se objeví druhé letadlo a Vy ho musíte sledovat.

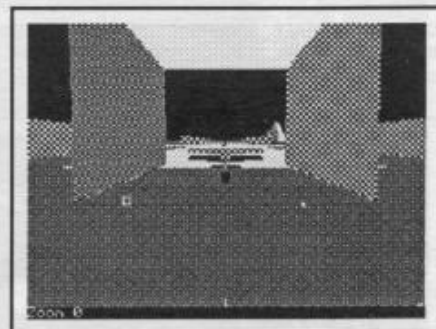
Flight instruction - V této části se z Vás stává žák pilotní školy. Na výběr máte opět 3 možnosti výcviků. Basic - Obsahuje 7 letových zkoušek z různých pozic nad letišťem, pyramidou. Při každé zkoušce se dole na obrazovce píše co máte dělat (pokyny instruktora). Advanced - středně těžká výuka akrobacie, která obsahuje celkem 8 zkoušek. Aerobatic - Posledních 6 velmi náročných zkoušek. V tomto menu máte možnost si pustit i demonstraci akrobatického letu. Vždy na spodu okna jsou dvě položky: OBSERVE a FLY. Observe znamená, že budete pouze pozorovat jak se má podle instruktora létat. FLY je samotná pilotáž.

Ovládání

- Q - nahoru
- A - dolů
- O - doleva
- P - doprava
- F - Flaps - klapky
- G - Gear - podvozek
- B - Break - brzdy
- 1-5 Thrust - plyn
- L - pauza
- W - pohled - západ
- E - pohled - východ
- S - pohled - jih
- D - pohled - sever
- X - pohled z pumovnice letadla
- U - pohled ze satelitu
- Z,C - Zoom - přibližování / oddalování letadla, cílů...
- V - View - celoplošný pohled z kokpitu

R - return to cockpit - návrat do pilotní kabiny

T - pohled na ocas letadla

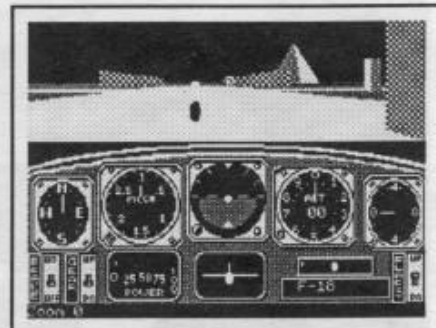


CS - pohled z letištní věže

SS - Mission menu - submenu

Y - First menu - první menu

Grafické provedení hry je v některých ohledech lepší než ve Fighter bomber (mnoho objektů na hrací ploše, přistávací dráhy jsou důkladně zpracovány) na druhé straně je však grafika mírně pomalá, letadla poznáte jen podle toho jestli je to tryskové letadlo nebo 1-2 plošňíkové, těžká ovladatelnost (spíše až moc věrná, ne?), skoro nulové zvukové efekty, bez



podrobného manuálu je skoro nemožné hrát hru bezproblémově, při letu směrem dolů se někdy stane, že vypadne motor a napíše se STALL (zastavení letadla) navzdory tomu, že otáčkoměr ukazuje 140 Mph a ukazatel přetížení se zblázní a napíše -3G (nic proti, ale letím střemhlav k zemi) a dokonce se mi podařilo snad rozbít pilotovi brýle a poškodit jeho zrak natolik, že nic neviděl a obrazovka byla až do havárie úplně černá.

Nicméně se jedná o verzi 1.0, takže je možné, že v pokročilejší verzi budou tyto "chyby" odstraněny. Tato hra byla vydána jen ve 128 verzi a nic nenásvědčuje tomu, že by byla i pro 48 (zobrazování je děláno přes 2 Vram!).

P.S. Je možné, že se u nás objevila již novější verze této hry a tento popis jí v něčem nevyhovuje. Ještě jednou upozorňuji, že se týkal verze 1.0.

- JSH -

NAVY SEALS 1, 2 128

Ocean 1990

A je to tady! Jedna z nejlepších her na Spectru od fy. Ocean je tu pro vás, abyste posoudili její kvalitu. V Anglii si tato hra získala mnoho příznivců už i proto, že je téměř shodná s Amigounskou verzí a i z té hrstky stovacetosmičkářů u nás jsou slyšet pochvalné ohlasy. Plným právem považují tuto hru za trháka, protože grafikou, animací postav, provedením a vysokou

A nyní ke hře.

Pět speciálně vycvičených bojovníků námořní pěchoty bylo vysazeno z ponorky kdesi v Libanonu a v přístavu začíná jejich první akce.

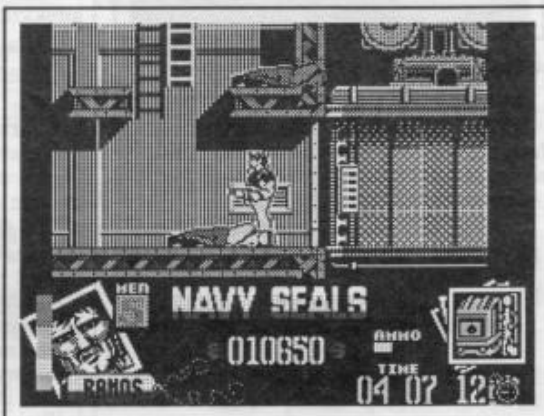
Přístav - vaším úkolem je rozmístit 9 časovaných bomb na červené bedny označené nápisem USA. Stiskem "W" přepínáme mezi čtyřmi druhy zbraní: samopalem, ostřelovací puškou, raketometem a plamenometem.

Na vaší cestě za vítězstvím likvidujte všechny nepřátele.

Stiskem W si navolte raketomet a střelte, Araba nad vámi zabijí střepiny. Jděte doprava a skočte. Šplhejte doprava k bedně s nápisem USA (bomba se sama aktivuje) a opět vystřelte z raketometu (třeba do zdi), výbuch zabil Araba nad vámi. Jděte doleva, přelezte bednu a střelte do ní raketometem. Arab na plošině nad vámi je mrtev. Po žebříku nahoru a aktivujte bombu, zpět po žebříku dolů, pak doprava, vylezte na bednu a skočte na řetěz. Po řetězu vyšplhejte nahoru. Bomba se už aktivovala (vždy se ozve zvuk) a stále doprava, skok a před námi je další bedna a zbývá nám aktivovat jen už 5 bomb. Vyskočte na nádrž s olejem, spadněte dolů a instalujte bombu. Doprava + skok a šplhejte doprava. Vidíte další bednu? Výborně, tak rychle k ní. Opět doprava až na poslední římso. Seskočte dolů a opatrně jděte dopředu. Přehoupněte se na římso nad vámi a zlikvidujte Araby, jež stráží bednu. Aktivace bedny, doufám, proběhne bez problémů. Seskočte o patro dolů, ještě jednou o patro dolů, pak po žebříku dolů, v polovině začnete šplhat doprava, seskočte na zelené barely doleva a použijte raketomet, jděte stále doleva. Nad vámi korzují dva Arabové, pomocí raketometu je zničte a jděte ještě doleva, pozor před vámi je zase Arab, zničte ho a instalujte bombu, pak jděte zpět doprava, po bednách nahoru, otočte se a skočte na plošinu.

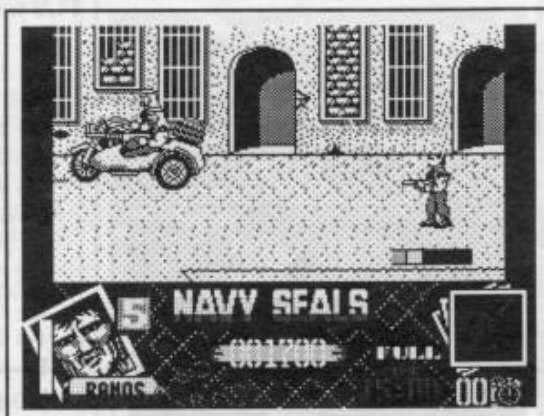


Pomalu nahoru po zelených barelech, pak dolů a zničte poslední nepřátele, aktivujte



poslední nálož a pak již tato akce končí. UFFFFFFFFF, byla to fuška!

Doufám, že už víte jak na to, a dostáváme se do druhé části této hry:

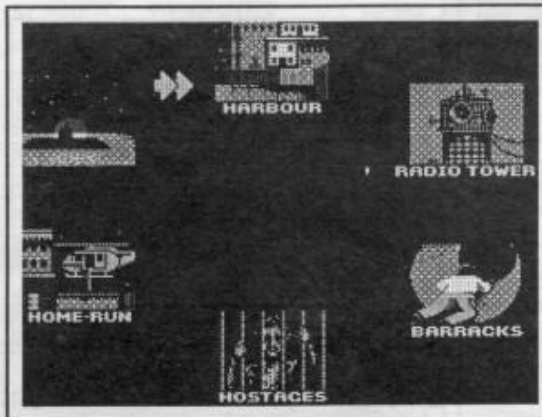


Radio Tower - musíte zničit nepřátelskou radarovou základnu, která je určena k likvidaci balistických raket. Ve věži musíte rozmístit už 12 (!) náloží a pobít co nejvíce nepřátele. Vzhledem k tomu, že tento level je velmi náročný, je tu k dispozici i perfektní výtah, který... (no snad vám nebudu vysvětlovat co dělá výtah?). Tento díl končí kdesi v podzemí.

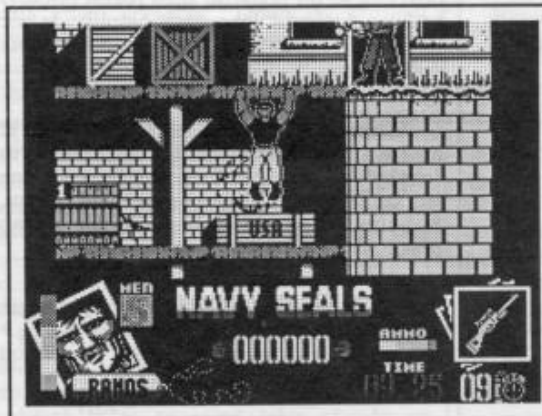


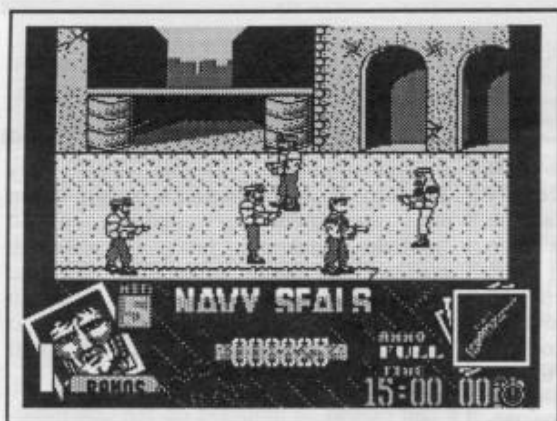
profesionalitou grafického ovladače předčí i novinky jako je Robocop 3 a Darkman.

Hudebně a zvukově na tom je Navy Seals velmi dobře. Mathew Cannon si s



hudbami pohrál a lze je označit za velmi slušné, rovněž ozvučení hry je na vysoké úrovni.

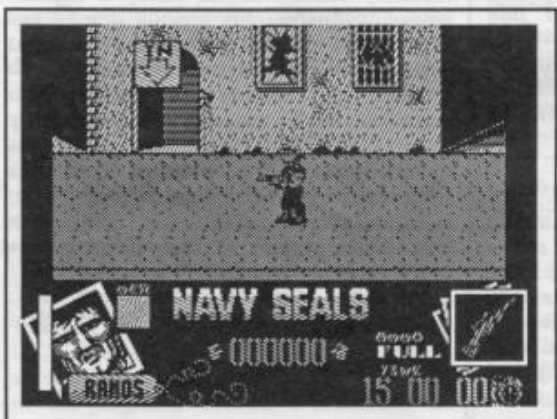




Barrack - očitáte se ve složitém komplexu budov a vaším cílem je rozmístit



7 bomb. Tato část zřetelně navazuje na konec z předcházející části, je asi nejtěžší, proto nezlomte váš joystick nebo nezničte



citlivou klávesnici vaší 128-ky nebo +2-ky!

Hostages - vaším cílem je osvobodit jedno rukojmí. Ocitáte se v obrovské, dobře vyzbrojené věznici a musíte zde proto umístit až 19 časovaných bomb. Na konci této části, kdy se nacházíte před celou s rukojmím vás čeká fantastické překvapení, které ve hrách podobného typu hned tak neuvidíte!

A je tu poslední část **HOMERUN** (útěk domů). Vaším cílem je probít se ke svým lidem, kteří na vás čekají úplně vpravo na konci základny. Podaří se vám pobít většinu nepřátel, jste už na dně svých sil když s hrůzou zjistíte, že očekávaný gazík tu není. Navíc ještě přijede auto plné Arabů a jen tak tak se z toho dostanete. Vaši vás však nenechali na pospas osudu a během chvilky přilétá vrtulník, vylezete po žebříku do kabiny, vrtulník odletí a to je konec Navy Seals I. Od Amerického námořnictva dostáváte mnoho řádů za hrdinství, ale "dáblové s ručníky na hlavě" opět zosnovali spiknutí. To však na vás čeká až v druhém díle.

Tak to byl hrubý popis Navy Seals I. Hry, která je výbornou akční střelnicí, která vyžaduje vaši rychlost a postřeh. Její hodnocení je prosté - 1*!

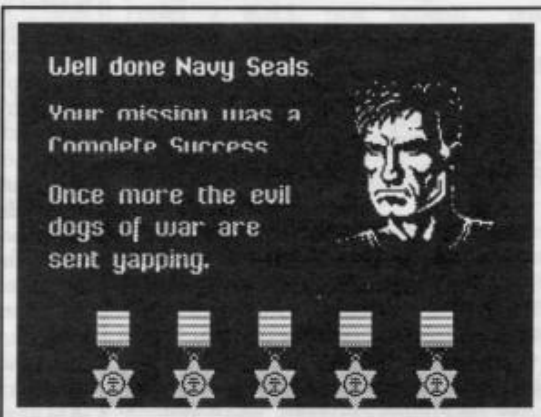
No a co nám přináší druhý díl?

Tentokrát je pohled na našeho bojovníka poněkud jiný než u první části, jakýsi trojrozměrný - nebo spíše pohled z boku.

Vylezte do dveří nad nimiž bliká nápis IN. Po chvíli se objeví perfektní obrázek zachycující váš rozhovor s Alim. Ali je tajný špión, který vám prozradí, co máte udělat. Stiskem EDIT+CAPS LOCK si zobrazíte mapu a její ovládání máte v ruce. Blikající křížek zobrazuje vaši pozici a další blikající "bod" je místo, kam musíte dojít. Tam na vás opět čeká Ali a dá vám další instrukce.

Let's go Navy team! A ještě jeden fantastický obrázek zachycující Aliho, jak vám dává poslední rady.

Na konci kdy zničíte poslední "sajdku" (= motorka + přívěs) se postavíte za roh jednoho domu a několika granáty zlikvidujete základnu nepřátel. To je jejich úplný konec. Objeví se americká vlajka (jak málo stačí k tomu, aby se vlnila?) - justice,



pravda a americká cesta opět zvítězily a to je konec Navy Seals II.

Máte-li možnost volby nekonečných životů, tak není rozumné na tuto otázku



odpovědět hned jednoznačně. Proč? Umíte a uvidíte!



To je více méně vše o těchto hrách. Druhý díl je poněkud slabší, ale řada digitalizovaných obrázků hru umocní a stává se tak velmi akční. Hudebně je bohužel druhý díl stejný a ozvučení se moc nezměnilo, ale to nic neubírá na faktu, že tato hra se i dnes řadí mezi trháky na 128.

- JSH -

Jak začínali... Tentokrát George K.

Jak začínali - nebo co já vim, jak se ten seriál jmenuje...

Minule tu vzpomínal na své spectrácké začátky UNIVERSUM a dneska se s Vámi o podobné zážitky podělím já.

První počítač, se kterým jsem kdy přišel do styku bylo **IQ 151**. Na první pohled se mi nelíbilo a navíc jsem nemohl pochopit, proč tak dlouho musí číst z kazety nějaký program a neustále jsem navrhoval, aby se ta doba zkrátila tím, že se kazeta přetočí... O pár týdnů později jsem poznal **PMD-85** a to mě uchvátilo, protože na rozdíl od "fkvéčka" umělo pracovat s jemnou grafikou a nevypadalo jak peříňák. Začal jsem navštěvovat pionýrský kroužek, kde jsem každý týden začínal dělat nějakou novou skvělou hru (třeba **Bacil Emil**, **Díl na penězokazů**, atd.) - sice jsem nikdy žádnou nedokončil, ale zato jsem se za pár měsíců naučil "programovat".

V roce 1986 jsem dostal k narozeninám **ZX Spectrum+** a od té doby na něj nenechám dopustit, protože je to určitě ten nejlepší domácí počítač, co byl kdy vyroben. Zpočátku jsem měl problémy s hardwarem - asi po měsíci mi odešla do věčných lovišť klávesnice, ale protože počítač byl v záruce, vyměnili mi ho za jiný, který - k mému rozčarování - nešel vůbec. Třetí exemplář **ZXS** už byl v pořádku a vlastním ho dodnes; nikdy jsem ho nebyl nucen dávat do opravy (pozn. posílám škodolibý pozdrav M. Bílkovi a jeho Didaktiku Gama), jen jsem asi 3x změnil klávesnici, což je u pluska i gumáka běžné. Další hardwarové problémy se objevily až po pěti letech, když jsem si koupil D40 a Melodika, jenže za to sir Clive Marles Sinclair nemohl, tak proč to sem tahat...

Vždycky jsem trpěl pocitem, že napíšu spoustu skvělých her - no, napsal jsem jich asi deset a žádná skvělá nebyla. Moje úplně první hra se jmenovala **Theodor & ostrovy** (1986) a silně připomínala hru **Žaby z PMD**. Zamýšlel jsem sice napsat pokračování **Theodor na lyžích** (pod vlivem "pecky" Horace Goes Skiing), ale nějak jsem si to časem rozmyslel, a rozhodl se udělat něco víc akčního, něco, co na Spectru chybělo: **KING KONG**a. Od roku 1986 do roku 1990 jsem to každý rok zkoušel asi dvakrát, a i když jsem se vždy dostal o kousek dál, hru jsem naštěstí nikdy nedokončil. Skutečným skvostem mezi mými "kongy" byl **Donkey Kong Junior** (hra jistě známá z pouličních automatů) - s maximálním vypětím jsem naprogramoval chůzi vlevo, vpravo, šplhání po jedné liáně, šplhání po dvou liánách, pády, skok vlevo a - na skok vpravo už mi v paměti nezbyl ani bajt. Místo toho, abych se program pokusil

napsat znovu a lépe, řekl jsem si, že basic pravděpodobně není to pravé, protože se v něm nedá vytvořit delší program, a začal se pít po strojovém kódu. K mé smůle nikdo z mých známých o něm nic neměl; krom jednoho človíčka, který mi přinesl nějaký oxeroxovaný papír se záhadnými slovními spojeními jako "přičtení s maskou" a "odečtení pŕlmasky" - dodnes jsem to sice nepochopil, ale ten papír mám ještě schovaný. Pátrání po něčem o strojáku pokračovalo, až jsem jednou náhodou zalistoval svojí příručku od Spectra (tvrdila, že strojový kód je velice rychlý a že psát v něm programy dokáže jen skutečný profesionál), kde mi padla do oka ASCII tabulka. Ta, jak známo, obsahuje na Spectru nejen klasickou abecedu a ostatní základní znaky, ale v druhé polovině jsou ukryty příkazy basicu - a tady se stala ta chyba: vůbec mi nedošlo, že basic má příkazy očíslované kvůli jednodušší orientaci a tisku, kratšímu uložení programu a buhvíčeho ještě, ale vydedukoval jsem, že jsou to čísla "příkazů strojového kódu". Třeba příkaz **CIRCLE** má číslo **216** (zkuste si **PRINT CHR\$ 216**), takže já jsem pěkně do paměti napoukával **216, 88, 128, 20** (běžná syntaxe příkazu **CIRCLE** v basicu), sputil to **RAND USR** a moc se divil, že to nic nedělá... Pak se na mě přece jen usmálo štěstí a získal jsem popis nejdříve instrukčního souboru procesoru **MHB8080** a potom i **Z80**. Můj první program ve strojáku přenášel obrázek odkudsi z paměti do videoram - psal jsem ho (ještě s kamarádem) asi týden, použil v něm dvě smyčky a zaplácal polovinu registrů... přitom stačil jeden **LDIR**.

Ale ještě zpátky do dávné minulosti... do roku 1987; v té době jsem ještě "naprogramoval" výkřiky jako **Bombardér** (ano, ten co ho k mému překvapení i zděšení uveřejnilo minulé číslo ZXM), **Ptáci** (vlastně kopie hry Ptáčník Harold z PMD) a **Telefony** (historii Telefonů jsem do vyčerpání rozepsal v manuálu KUD-3, můžete si ji tam přečíst). U Telefonů mě napadlo, že bych vlastně mohl využít softwarových vymožeností, kterým se říká kompilér, a celou hru jsem přeložil pomocí Floating Point Compileru (to byl můj nejoblíbenější, za kterým daleko zaostávali jinak populární HiSoft Colt Compiler, Mcoder III nebo ZX Compiler) - měla asi 25kB živého kódu (vida!) a nechodila na Didaktiku Gama.

Další "hra", kterou jsem udělal, bylo **UFO** (1988), alias Všechno za minutu (kdo pamatuje tuto televizní soutěž, musí mu to být jasné). UFO už jsem dělal s tím, že ho zkomplikuji a také jsem s ním tak naložil (ke své velké spokojenosti). Následoval **Souboj Titánů**, moje první textovka, která ovšem stála za starou bačkou. Snad jedině příkaz "*vezmi pokácený strom*" byl určitě

povšimnutelný... Trochu lepší byly **Telefony II**, které nemají s prvním dílem nic společného - jedná se o "graficko-logickou" hru s příšernou grafikou a ještě příšernější logikou; sympatické je jen to, že můžete chodit do kina, kde většinou dávají film o jedinečnosti počítače ZX Spectrum oproti rohožkám zn. Atari (fuuuu!).

Jednoho dne jsem se ve škole rozhodl, že udělám textovku v reálném čase - a opravdu - stalo se tak (skoro). Vznikl **Thriller** (1989, 1990) - nervák obsahem i formou, za který se ani moc nestydím - hru jsem zkompiloval pomocí **BLASTu** (pravděpodobně ten nejlepší kompilér basicu na ZXS), přilepil k ní hudbu na Orfeovi (Kaťuša), nasamploval smích Vinceta Price z Jacksonova Thrilleru a pustil mezi lid. Vzhledem k "ochraně", kterou jsem hru vybavil, opět nešlo Thriller hrát na Didaktiku Gama, takže jsem začal získávat pochybnosti o slově kompatibilita a uvažovat, jestli by se spíš nehodilo kompatibilita.

Ačkoliv strojový kód jsem uměl od roku 1988, byl jsem líný v něm dělat programy; neměl jsem totiž assembler a psát je v číslech z basicu mi moc nevyhovovalo. Zkusil jsem to několikrát, ale měl jsem k práci vyložené špatný přístup: psal jsem programy sekvenčně, tj. začal jsem smazáním obrazovky a nastavením barev - na víc mi obvykle nezbyly síly. Po čase jsem sehnal **GENS** (ovšem byl mi tak nesympatický, že jsem ho v životě nepoužil), **Zeus** (s tím jsem pro změnu nedokázal dělat) a **Editas** (ten se mi docela líbil), ale na moji strojákovou produkci to nemělo větší vliv. Někde jsem si přečetl, že nejlepší assembler pro ZXS je **Laser Genius**, a tak jsem ho dlouho sháněl - až v roce 1989 jsem se dočkal. Musím přiznat, že autor článku nelhal: **Laser Genius** nejlepší byl, je a bude. Bohužel zabírá trochu víc paměti, než je zdravé, a tak se na něm delší programy pší s obtížemi - to neplatí, používáte-li ZXS 128. Postupem času jsem naprogramoval spoustu rutin, které jsem ovšem nikdy nepospojoval v rozumný celek; veškeré snahy o napsání nějaké "jedinečné" hry končily stejně - pro nedostatek nadšení nedoděláno. Jedinou výjimku tvořil program **Soundbox**, do kterého jsem napsal asi 2kB rolujícího textu a 40kB nasamplované písničky **GHOSTBUSTERS** (cca 26 sekund), a vytvořil tak své první hudební demo, aniž bych vlastně tužil, co dělám. Rozdělal jsem i **Soundbox II**, ale ten už světlo světa nespátřil, a já zase zkoušel udělat bomba hru. Takhle jsem přežil většinu roku 1989 i 1990 a byl bych tak strávil i rok 1991, nebyť nápadu, který mi za to doděláním prostě stál: **Beerland**.

S odstupem času se mi sice Beerland moc nelíbí po stránce estetické (zvláště na Didaktiku M vypadá odpudivě), ale jinak jsem s hrou spokojený - asi proto, že vychází z reality, vystupují v ní skutečné postavy a koneckonců i já. Konečně jsem také použil trochu strojových rutin (i když je jinak Beerland v basicu) - vkládání znaků, proporční tisk a roll obrazu směrem vzhůru. Původní verze Beerlandu měla navíc páskový loader, který při nahrávání po obrazovce posunoval text (pravda, po znaku, ale já si tenkrát myslel, buhvíjak není dobrý); o tento loader hra přišla, když jsem ji nabídl jakési firmě **PROXIMA** k distribuci. Asi do prázdnin se nedělo nic, a pak se Beerland objevil na jednom z kompletů her jako premie, což já osobně považuji za podceňování textovek. Od dob Beerlandu pracuji pro **PROXIMU**, což má tu výhodu, že většinu věcí, které rozvrtám, nakonec dokončím (ač s odporem) - i tak mi (díky minulosti) zůstává poměr rozdělaných a dodělaných programů asi 10:1.

V létě 1991 jsem (stejně jako rok před tím, rok potom a dva roky potom) opustil myšlenku napsat pouliční mlátičku **The Terminator** (dnes už toho mám tolik, že by se snad dala poskládat i celkem provozuschopná hra... jenom nebýt tak líný) a začal dělat na **Jméno Růže**, které se do konce roku objevilo v distribuci. Že se jedná o hru celkem populární, dokazuje počet prodaných kopií a počet černých kopií (těch je samozřejmě daleko víc) - minulý měsíc jsem se snažil předělat ji i pro nemajitele disketových jednotek, jenže pak jsem narazil na spoustu nepříjemných problémů, takže i nadále - **Jméno Růže** pouze pro D40/D80. **Jméno Růže** je sice textovka - ovšem má nejen texty, ale i spoustu grafiky a dost špatně se srovnává s čímkoliv jiným na Spectru. Pojmout hru tímto způsobem vůbec nebyla moje originální myšlenka - napadlo mě to při sledování her od Lucasfilmu (Indiana Jones, Tajemství Opičích ostrova, Loom) na PC. Pokud budu dělat na ZXS ještě nějakou hru bude se pravděpodobně podobat (stylem) **Jméno Růže**, ale chci aby se ještě víc přiblížila tomu, co běhá na 16-ti bitech.

Koncem roku 1991 a začátkem 1992 jsme s **UNIVERSUMem** napsali **TOOLS 40**, který jsem postupem roku předělal na **TOOLS 80** (i zde má **UNIVERSUM** nemalý podíl) - jeho (zatím) poslední verzi (2.3) jsme udělali letos na jaře a oba doufáme, že už na ni nebudeme muset sáhnout, jenže znáte to... staré chyby mizí a nové se objevují, nové funkce se přidávají a program se pomalu nevejde do assembleru najednou. V roce 1992 také vznikla předělávka **Telefonů I**; program jsem psal 3 dny, takže narozdíl od **FIFA**, které o hře napsalo, že je geniálně debilní (nebo debilně geniální, nevím přesně), si nemyslím, že jsem svůj čas mohl strávit účelněji. Z **Telefonů** jsem nakonec udělal utilitu pro **DESKTOP** (je na **KUD-3**), a slíbil, že ji uživatelé budou moci najít v každém mém novém programu. Pravda, v každém to asi

nebude, ale ve dvou už je (jeden z nich jsou **Heroes**, ale ten druhý...?) a v jednom ještě určitě bude... sliby se mají plnit (občas).

Léto 1992 je ve znamení **HEROES '92**, nejlepší textovky, co jsem kdy napsal, a podle několika dopisů a pirátských katalogů i nejlepší českou textovkou vůbec (berte to klidně jako vytahování... heč!). Dal jsem si tu práci a vyrobil i **AY-demo HEROESdemo**, které snad viděl každý, kdo si v té době od **PROXIMY** něco koupil. Jinak ovšem musím říct, že mě soutěž "Heroes" silně zklamala, protože se jí zúčastnilo žalostně málo lidí (a to jste do dotazníků furt psali "chceme soutěžit...") a ještě méně lidí ji vyřešilo (devět - no to je fakt hausnumero!). Nemyslím si, že by hra byla **TAK** těžká, ale spíš hráči **TAK** líní - jak to nemá **POKE** nebo podváděcí mód, tak se to nedá hrát a šup s tím do šuplíku. Každopádně mě to vyléčilo natolik, že žádnou další soutěž už organizovat nehodlám, můžete být klidní.

Na vánoce 1992 jsem napsal diskové kopírky **SINGLEcopy** a **MASTERcopy**, které se letos na jaře dostaly do distribuce (pro informaci: poslední verze mají čísla 2.2



a 2.3), jenže to já už jsem dokončoval kompletaci **KUD-3** a měl téměř hotového **Tolstoje** - slovníkovou databázi, která je dělana přímo pro disketovou jednotku D40/D80 a pojme úctyhodné množství dat (až 1 mega). V dubnu jsem udělal své další **AY-demo DUCKMANIA!**, které nemá žádný hlubší smysl, krom toho, že je to strašná kachnovina. Pak jsem vytvořil nový unírún pro D40 s názvem **File Manager** (viz. minulý ZXM) - stejně jako **HEROESdemo** a **DUCKMANIA!** i tohle je shareware.

Jen tak z matematické zvědavosti jsem naprogramoval textový pressor **Tolkien v5.0** - a skutečně, výsledky předčí moje očekávání... pressor pakuje až na 50% původní délky; takto spakovaný text se dá výhodně použít v textovkách, protože je naprosto nečitelný oku pirátovu a navíc ho před použitím nemusíte rozbalovat - stačí použít asi 50-ti bajtový prográmeček, který ho dokáže přečíst i takhle zahuštěný.

V současnosti pracuji na **SQ-Trackeru**, programu na tvorbu **AY-hudby** (viz. článek od **Scalex** o hudebních editorech). Hudebníci, máte se na co těšit - **Soundtracker** je vykopávka, nejlepší esa na **AY-hudbu** používají **Scalex** - **Qjeta** -

Tracker (aspoň doufám). K **SQ-Trackeru** vníká i **SQ-demo**, které tvořím společně s novým členem **Depeche Code Exotic Fruits** (Tropické Ovoce) - i když je to "jenom" demo, dáváme si tentokrát záležet, protože už nám leze krkem poslouchat, že Poláci z **E.S.I.** jsou nejlepší... pokud dokážeme demo dodělat tak, jak jsme si ho naplánovali, bude to pro naše severní sousedy **SHOCK 2**.

Možná jste si všimli, že jsem nenapsal nic o hrách, které jsem na svém Spectru rád hrál (a občas ještě hraji) - tak tady to je: Moji oblíbenou hru **Manic Miner** jsem vlastnil dokonce dřív než počítač a vydržel jsem ji hrát do omrzení. **Jet-Set Willy** už se mi moc nelíbil, ale zato mě uchvátila **The Chip Factory**, která "maníka" dost připomínala. Moje další oblíbené byly **Tapper**, **Commando**, **Splitting Images**, **Knight Lore**, **FIST**, **Pyjamarama**, **Three Weeks In Paradise**, **Dan Dare**... a spousta dalších, vesměs staršího data výroby.

Z textovek jsem nejraději hrával **Indiány Džounse** od Franty Fuky a **Fidlerův Belegost**, ostatní české textovky mě odrazovaly dost velkým nevkusem a pravopisnými chybami.

Mezi anglickými textovkami považuji za dokonalé **Rigel's Revenge** a **Kayleth**, ačkoliv jsem měl při jejich hraní zpočátku trochu problémy s jazykovou bariérou. Z novějších her si rád zahraju **Dizzy**, hlavně **Dizzy 5**, který mi připadá z celé té série nejpromyšlenější. Takhle bych mohl jmenovat hry ještě nejméně dvě stránky, a tak radši napíšu ty, které rád nemám: jsou simulátory (s výjimkou těch od J. Flašky) a 3D hry (ve stylu **Incetive**), jinak se těšící značné popularitě. Protože na **Didaktiku** a **Spectra** přichází hodně nováčků a ti tyto hry neznají, uvažujeme o tom, že příští rok bude **ZXM** vycházet seriál o historii her na **ZXS**.

Zcela jinou kapitolu tvoří hry pro **ZX 128** - i když je to skoro ten samý počítač, jen díky větší paměti a hudebnímu čipu se na něm kvalita her blíží **Amize** (v rámci hardwarových možností, samozřejmě). Těžko to popisovat, od toho je tu **JSH**, aby Vám ve svých článcích popisoval, co si na své 48-čce zase nezahrajete... ale skutečně - kdo viděl **Dizzy 5 v128**, již stěží se přinutí hrát 48k verzi... a což teprve hráči, kteří uvidí něco jen 128k - třeba takovou bombu jako **Navy Seals** nebo **Darkman**. Jenže to už jsme se dostali z historie do současnosti a já nemám pomalu o čem psát... každý rok si říkám, jestli **Speccy** dožije konce roku a každý rok jsem překvapen tím, že ano. Tenhle malý počítač se pomalu stává nezávislým na čase a na možnostech, které mu byly původně vymezeny - kvalita programů neustále stoupá a přitom, **Speccy** zůstává stále stejný jako před pěti jedenácti lety... byla by věčná škoda, kdyby tomu někdy mělo být jinak, a proto se s Vámi rozloučím nostalgickou větou, kterou končí polské demo **Shock**:

Keep Speccy Alive!

Přehled klubů pro ZX Spectrum v ČR a SR

Vážení čtenáři,

přinášíme Vám stručný přehled dosud působících klubů uživatelů počítačů kompatibilních s počítači standardu ZX Spectrum, které vyvíjejí svoji činnost na území České a Slovenské republiky. Seznam a informace v něm jsou převzaty z letáků klubů a z informací, které nám poskytly vedení některých klubů - stav k 19.7.1993.

Sinclair Club Ostrov

Uživatelský klub s regionální působností. V současné době podstatně zužuje výběr nových členů. Má zájem především na spolupráci s aktivními programátory a hardwaristy. Někteří členové a spolupracovníci klubu jsou pravidelnými přispěvateli tohoto časopisu. Schůzky klubu jsou každých 5 týdnů v počítačové učebně 5. Základní školy v Ostrově (síl DG), vždy v sobotu odpoledne.

Kontaktní adresa: Sinclair Club
P.O.Box 132
363 01 Ostrov

Sinclair Club Šumperk

Regionální klub, zaměřený především na výměnu zkušeností a softwarovou i hardwarovou poradenskou činnost. Klub je zatím zdarma. Schůzky klubu každý lichý čtvrtek v centru ELTEC v domě mládeže v Šumperku. Do klubu si zájemci musí přinést s sebou vlastní počítač + magnetofon, nebo D40/D80. Televizory jsou k dispozici.

Kontaktní adresa: p.Josef OTTA
Finská 12
787 01 Šumperk

ComputerClub602 PRAHA

Nástupce počítačových klubů bývalého Klubu 602 v Praze. CC602 se mimo jiné aktivity zabývá také vysíláním teletextu na ČT1, kde můžete mj. na stránkách 640-649 získat informace o výpočetní technice a elektronice. Bližší informace o činnosti na kontaktní adrese.

Kontaktní adresa: ComputerClub602
p.Rudolf Mach
Žitná 4
120 00 Praha 2

Sinclair Club Hradec Králové

Regionální klub, jehož činnost není zaměřena na počítačové hry. Klub vydává vlastní časopis ZX CODE (10x ročně) a příležitostně speciální monografické číslo. Členský příspěvek je 50 Kč za půl roku. Klub se schází jednou za čtrnáct dní, v úterí od 17.00 hodin v budově YMCA v Šafaříkově ulici (křižovatka s Gočárovou třídou).

Kontaktní adresa: Ing. Petr Exner
Havlíčková 294
500 02 Hradec Králové I

Sinclair Club Ústí n. Orlicí

Regionální uživatelský klub. Bližší informace pro všechny zájemce z jeho blízkého okolí na kontaktní adrese.

Kontaktní adresa: VT centrum AMAVET
Letohradská 1359
562 06 Ústí nad Orlicí

Sinclair a Didaktik klub ČSE

Pražský uživatelský klub České společnosti elektroniků. Schází se v každý pátek v 17.00 hodin v pracovním klubu na Tolstého ulici č.16 v Praze 10, kromě pátku před sobotní schůzkou. Bližší informace na kontaktní adrese.

Kontaktní adresa: Klub 017 ČSE
pan A. Strnad
Tolstého 16
100 00 Praha 10

Sinclair Club Plzeň

Malý uživatelský regionální klub. Schůzky každé úterý od 17.00 hodin v autoškolě v ulici Na vinicích v Plzni (naproti benzínové pumpě u kruhového objezdu). Podrobnější informace přímo na místě.

ZX klub Uherské Hradiště

Uživatelský regionální klub. V současné době připravuje mj. např. seznam služeb a výrobků pro ZX Spectrum. Podrobnější informace na kontaktní adrese.

Kontaktní adresa: p. Martin Žamboch
Sadová 1036
686 00 Uherské Hradiště

Klub uživatelův BETADISKu

Nový korespondenční klub, určený všem majitelům tohoto poměrně rozšířeného řadiče. Klub si klade za cíl zkompletovat veškerou dostupnou literaturu a programy určené pro Betadisk. Klub vydává jednou za dva měsíce vlastní DiskMagazín. Klub založil také fond volně šířitelných programů.

Kontaktní adresa: BETAKLUB
pan. Imrich Konkoř
Leškova 9
811 04 Bratislava

Klub D40/D80

Korespondenční klub uživatelů této nejrozšířenější disketové jednotky v našich krajích. Podrobnější a novější informace nejsou k dispozici, protože klub přestal s námi udržovat styky. O bližší informace si tedy napište na adresu klubu.

Kontaktní adresa: Klub D40/D80
p.Jaroslav Kořínek
M. Gorkého 3
431 91 Vejprty

Post Spectrum Club

Největší korespondenční klub u nás. Nabízí širokou škálu služeb. Klub pracuje ve dvou sekcích - kazetová a disketová (D40). Všechny podrobnější informace na adrese klubu.

Kontaktní adresa: PENT Computer - PSC
P.O.Box 531
974 01 Banská Bystrica

Klub výpočetní techniky Karolinka

Jeden z nejstarších uživatelských klubů pro ZXS, který se v současné době se zaměřil především na vydávání návodů k programům a hrám. Informace na staré známé adrese klubu.

Kontaktní adresa: Petr Jochec
č.p. 280
765 05 Karolinka

Připravuje se:

Klub pro emulátory ZX Spectrum na PC komp.

V souvislosti s hromadným přechodem uživatelů našeho milého Spectra na počítače vyšší třídy hodláme založit konzultační klub uživatelů jeho emulátorů na počítačích PC a komp. Předmětem činnosti by měl být vývoj především vývoj jejich dalších možností, především různých konverzních programů atd. Klub má tedy zájem především na spolupráci s **programátory** z řad bývalých spectristů nebo programátory působícími na zároveň ZXS i PC. Zájemci se mohou ozvat na adrese Sinclair Clubu v Ostrově.

Náš seznam jistě nezachytí některé kluby, především ty menší, stejně jako nemáme žádné podrobnější informace např. o Brněnském Sinclair Clubu. Obracíme se tedy na vedení všech doposud činných klubů, aby se nám na adresu Ostrovského klubu ozvaly a informovaly čtenáře ZXM o svojí činnosti.

Na závěr jednu velice dobrou radu pro všechny, kteří se rozhodli do některého z klubů napsat si pro bližší informace: **přiložte do obálky také ofrankovanou obálku na odpověď!** Z vlastní zkušenosti Vám můžeme potvrdit, že tato základní slušnost při dopisování mnohým radychtivým čtenářům stále chybí.

Seznam klubů pro Vás sestavili *Jarda Krejčí* a *Martin Nový* (alias Studio 22) ze Sinclair Clubu v Ostrově.

POZOR VÝZVA!

Hledá se klub uživatelů stále rozšířenějšího diskového řadiče **DATAPUTER - ZX DISCFACE QUICK**. Pokud o takovém klubu víte, dejte vědět na adresu Sinclair Clubu v Ostrově nebo do ZX magazínu. Pokud ještě takový klub neexistuje, dost možná se mezi Vámi najde nějaký nadšenec, který by takový klub i se všemi věcmi okolo založil a organizoval.

George K.'s

AnimAce 2

Díl čtvrtý: Čtvrtý díl

Čtvrtý díl druhé AnimAce Vás vítá a nese Vám opět něco nového... i když ne tak docela. V minulém čísle jsem Vám ukázal, jak vypadá jedna z nejrychlejších rutin na vykreslování sprajtů (včetně malé řídicí nadstavby), a dneska se budeme věnovat kombinaci sprajtů a pozadí, což jsem nakoukl v závěru dílu. Ačkoliv jsem sliboval konkrétní příklad na obsluhu sprajtů, ještě od toho pro tentokrát upustím a vrátím se k tomu až v příštím díle, když už budeme umět sprajty jak kreslit, tak mazat.

Pozadí jsou dvojí: pohyblivá a statická. Se statickými nejsou vlastně žádné problémy, až na to, že se občas "perou" barvy pozadí a sprajtů. Navrhoval bych buď dělat grafiku takovou, aby co nejvíce vyplňovala prostor čtverečků, nebo sprajty nechat přebírat barvu pozadí (eventuelně dávat na výběr, jak to dělala hra 3 Weeks In Paradise).

Pozadí pohyblivá s sebou přinášejí problémy - některé programové a některé fyzikální (všimněte si, že v některých hrách při chůzi vpřed střely vůči Vám "stojí" nebo naopak při chůzi vzad stojí vzhledem k pozadí). Pohyblivé pozadí nemusí být v jedné rovině a s tím náročnost na jeho obsluhu stoupá. Geniálně je tento problém vyřešen ve hře "The Way Of The Tiger", která má asi 5 úrovní pozadí - v každé úrovni jsou objekty (většinou hroby nebo stromy), které se pohybují různou rychlostí (podle vzdálenosti od pozorovatele) jako ve skutečnosti. Autoři hry ale ničím nerolují... zkrátka vzali každý z objektů pozadí jako sprajt a tomu určují kde má být vykreslen. Nápad jednoduchý, účinek ohromný (podle mne nejlépe vyřešená otázka prostoru na Spectru).

Když jsem zkoušel psát rolující ulici a na ní pouliční bitvu, došel jsem k několika poznatkům (netvrdím, že jsou to ty nejlepší, ale vedou k uspokojivému řešení). Celou ulici (pohled ze strany a z nadhledu - viz. např. jízda Batmobilem v Batmanovi 3 nebo Renegade II, atd.) si rozdělíme do několika pomyslných částí:

a) dolní obrubník - jede nejrychleji a postavy ho nikdy nepřekrývají

b) vozovka - je "prázdná" (nic na ní neroluje) a postavičky ji vždy překrývají

c) horní obrubník - jede o něco pomaleji než dolní obrubník a postavičky ho mohou překrývat (podle toho, kde na vozovce stojí).

d) horní chodník - je "prázdný" a proto s ním není třeba posunovat (postavičky - viz. případ horního obrubníku)

e) přízemí domů - rolují nejpomaleji; postavy jako u horního obrubníku

f) patra domů - rolují rychle jako přízemí, ale postavičky už je překrývat nemohou

Z uvedeného rozdělení vyplývá, že části a) a f) můžete rolovat přímo ve videoram a dále si o tuto část obrazu nemusíte dělat žádné starosti.

Horší je to s částmi zbývajcími - na nich mohou ležet (a většinou také leží) sprajty, ke kterým se můžete zachovat dvěma známými způsoby: všechny je urychleně "zvednou" z obrazu (tj. vrátit zpět původní pozadí), šoupnout s ulicí vlevo nebo vpravo a sprajty zase položit dřív, než to všechno blikne (nestihli jste to...). Při větším počtu postavicek se jedná o úkol téměř nemožný.

Proto přistoupíme k druhému řešení, které je (podle mne) využíváno většinou zahraničních her. Všimněte si, že pozadí (např. v RoboCopovi) bývají celkem jednoduchá, poskládaná z motivů

(také nepříliš složitých) a občas povypňovaná nějakým vzorkem. To umožňuje dvě věci: připravení (resp. narotování) všech motivů v pozadí do čtyř pozic (tj. po dvou bitech, bohatě to stačí pro plynulý posun) a vykreslování nového pozadí (žádné sbírání sprajtů) do obrazovky pomocí sekvencí instrukcí PUSH, což je bezkonkurenčně nejrychlejší (tento způsob používá má oblíbená COBRA). Bohužel, nevím, jestli to byla moje vina nebo vina našich východních sousedů, ale při zkoušení mé rutiny (pracující právě na onom principu "pušování" do videoram) se na Didaktiku M vytvářel ve vedlejší třetině zajímavý mlžící efekt (na ZXS všechno fungovalo bez problémů), takže jsem tento rychlý a efektní způsob vykreslování zase opustil.

Vraťme se ale k naší ulici, přesněji k její části b) a d). Chodník i vozovka jsou prázdné, tedy nevyplněné nějakým složitým vzorkem, a proto je obsloužíme zvlášť, a to tak, že je co nejrychleji zmažeme. Mohlo by to vypadat takhle:

```
ld sp,VRAM
ld hl,VZOREK
push hl
push hl
push hl
push hl
push hl
push hl
...
```

Na samém začátku naplníme SP (vrchol zásobníku) tak, aby ukazoval na pravý konec pixlového řádku z chodníku či vozovky. Pravý konec proto, že instrukce PUSH posunuje vrchol zásobníku k nižším adresám, tj. ve videoram to bude směrem vlevo. Dále dáme do registru HL vzorek, kterým chceme ulici vyplnit - nevím, jestli "vzorek" není moc přehnaný výraz, protože to většinou bude 0 nebo 65535. Pokud byste chtěli ulici šedou (tj. bod - mezera, bod - mezera), museli byste čistit zvlášť sudé a zvlášť liché řádky, jinak by se Vám na místo jemné šachovnice udělali jemné proužky. Program byste upravili takhle:

```
ld sp,VRAM1 ;první pixlový řádek
ld a,%10101010 ;vzorek šachovnice 1
ld h,a ;kopíruj do HL
ld l,a
rca ;vytvoř vzorek šachovnice 2
ld d,a ;a kopíruj do DE
ld e,a
push hl
push hl
push hl
...
ld sp,VRAM2 ;druhý pixlový řádek
push de
push de
push de
...
```

Příkazy PUSH opakuje podle šířky ulice (pozor! najednou se ukládají dva bajty!!!). Protože na Didaktiku M se jeví PUSH jako nepoužitelný, můžete zkusit tohle:

```
ld hl,VRAM
ld a,VZOREK
ld (hl),a
dec l
```

```
ld      (hl),a
dec     l
...
```

Nemusíte už kreslit ulici "od zadu", ale můžete VRAM definovat jako levý začátek řádku a namísto DEC L dát logičtější INC L - na rychlosti to ovšem nic nezmění, ta zůstane oproti PUSHům nechtutně pomalá (zkuste si to sami spočítat v taktech, jaký by byl rozdíl např. u ulice vysoké 40 mikrořádků).

Tím jsme si tedy ukázali princip, pomocí jakého vyčistíte vozovku a chodník - předpokládám, že sestavit z toho jednoduchou smyčku by pro Vás měla být hračka. Zbývá nám poslední... a podle hesla "to nejlepší nakonec" jsou to části c) a e), které musíme odrolovat na požadovanou stranu. Opět máme dvě možnosti: mít pozadí narotováno do čtyřech fází po dvou bodech nebo poctivě rolovat.

První varianta je každopádně rychlejší, protože odpadá čas strávený rolováním a naším úkolem je pouze přenést odněkud z paměti správný kus ulice do obrazovky. Přenos nejrychleji realizujete takto

```
ldi
ldi
...
```

kde HL udává odkud, DE kam a BC můžete použít jako počítadlo. Instrukci LDI opět umístíte tolikrát, jak je široká ulice (většinou 26-30 znaků). Značnou nevýhodou tohoto řešení je nárok na paměť - chvíli si násobte a zažijete šok. Ovšem nic se neříká, jak se to uvaří: je čas vnést do bezduchého vykreslování myšlenku. Ulici můžete mít v paměti reprezentovanou znaky a podle jejich čísel vyhledávat grafické předlohy pro 8x8 nebo 16x16 bodů a vykreslovat si je do rezervní obrazovky (pro každou fázi potřebujete jednu) - s pamětí už je to sice trochu lepší, ale neustále ještě musíte mít každou předlohu uloženou 4x a chcete-li trochu bohatší pozadí, je to stejně zlé.

Přejdeme teď na chvíli k druhé metodě a začneme stejně jako o předchozí. V paměti máte uloženu celou ulici (graficky), kterou si postupně rotujete do vnitřní obrazovky a tu kopírujete do videoram. Je to skoro to samé, až na to, že potřebujete 4x méně prostoru na grafiku, což na druhou stranu kazí zjištění, že rolování trvá neskutečně dlouho.

Nejvýhodnější metoda, co mne zrovna napadá, je skloubit obě nabízené varianty v jednu...

Grafika je uložena ve formě grafických předloh o velikosti 8x8 bodů (tedy ve stejné formě jako znaková sada). V jednom levelu budete muset vystačit s max. 256 různými motivy. Ulice je uložena ve formě čísel grafických předloh (protože vždy se něco opakuje, je to velká úspora místa). K dispozici máte 4 vnitřní obrazovky, jejichž rozměry jsou asi 6x28 znaků (1344 bajtů). Vykreslování probíhá asi takto: každá obrazovka je určena pro jednu pozici předloh (posunujeme stále po dvou bitech). Narotovat musíme předlohu do správné polohy pouze v okamžiku, kdy do vnitřní obrazovky vstupuje, pak už se vlastně nemění.

Postupně vykreslujeme do videoram 1., 2., 3. a 4. vnitřní obrazovku a tím získáváme celkem dobrý roll. Určitě namítnete, že bude zapotřebí posunovat řádky ve vnitřních obrazovkách - není to tak úplně pravda, šlo by to udělat i bez jakýchkoliv přesunů, ale obávám se, že než bych to nějak rozumně vysvětlil, popsal bych několik stran papíru, k čemuž nemám moc chuti, takže nato zkuste přijít sami.

Tímto se dneska rozloučíme - možná, že jste čekali něco konkrétnějšího, ale věřte mi, že konkretizovat moc nešlo, protože to vždy závisí na hře, na které děláte.

pokračování příště

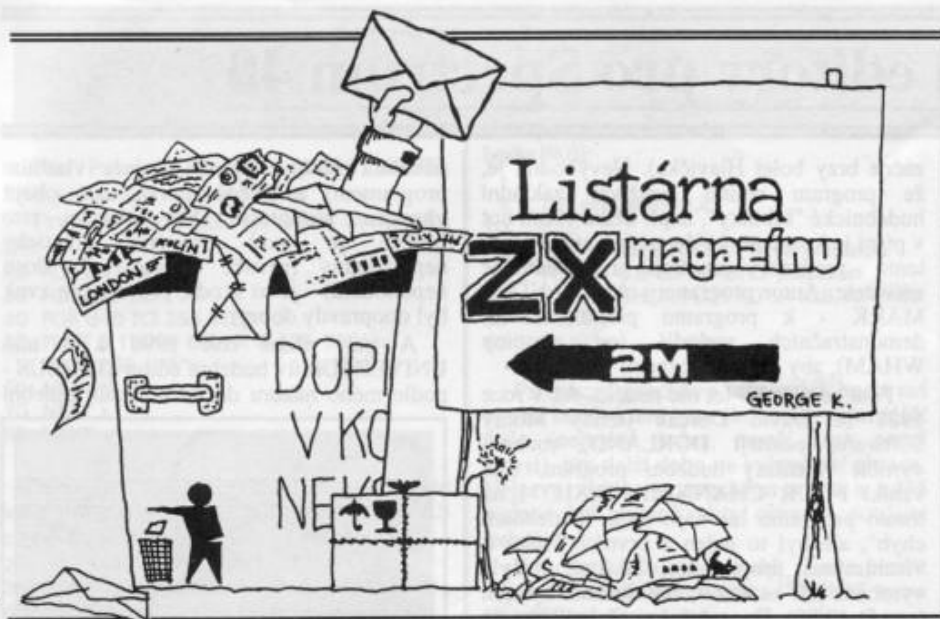
Animace v BASICu

Následující program je jednoduchá hra napsaná v jazyce BASIC (autor známý J. Flaška). Až ji budete opisovat, nahraďte podtržená písmena odpovídajícími grafickými znaky:

```
10 BORDER 7: PAPER 7: INK 0: FLASH 0: BRIGHT 0: OVER 0: INVERSE
0: CLS : RESTORE 6000: FOR l=0 TO 47: READ a: POKE USR
"a"+l,a: NEXT l
20 LET a$="PROXIMA PROUDLY PRESENTS": LET v=11: GO SUB 1000
30 FOR x=1 TO 120: PAUSE 1: NEXT x: GO SUB 2000: CLS : LET
v=10: LET b=2: LET a$="GRISLY DISPASSIONATE LETHAL
DESTROYER AND KILLER OF LOVELY MISERABLE
EARTHWORMS": GO SUB 1010
32 FOR x=1 TO 150: PAUSE 1: NEXT x: GO SUB 2000
35 CLS : PRINT AT 0,2: INK 1;a$: AT 11,0: INK 0:"PRESS -S- TO
START THE GAME !!!!";#0: AT 1,0: INK 2: BRIGHT 1:"PROGRAM
AND IDEA BY SUPERSOFT II"
37 PRINT AT 18,10:"PROTECTED BY"; AT 19,1:"THE RETURN OF
LIVING EARTHWORM"
40 LET a$=INKEY$: IF a$ <> "s" AND a$ <> "S" AND a$ <> " " NOT
THEN GO TO 40
42 RESTORE 3000: PRINT AT 21,0:"PLEASE WAIT I PLAYING MUSIC
!!!!"
45 READ a: IF a <> 100 THEN BEEP .1,a: GO TO 45
47 CLS : FOR x=1 TO 20: PRINT AT 11,7:"KILL EARTHWORMS II":
PAUSE 1: BEEP .01,0: PRINT AT 11,7,: PAUSE 1: BEEP .01,1: NEXT
x
50 REM "GAME"
60 LET v=17: LET score=0: LET u=17: LET l=1: LET skok=0
100 FOR s=27 TO 0 STEP -1: PRINT AT v,16:" "; AT v+1,16:"A": AT
v+2,16:"B": AT v+3,16:"C": AT v+4,16:" "; AT 20,s:"DEEF ";#0: AT
1,0:"SCORE: ";score: IF INKEY$ <> " " THEN LET skok=1
105 IF v=17 AND (s<17 AND s>11) THEN FOR x=1 TO 50: LET v= RND
*20-10: BORDER INT ABS (v/1.42): BEEP .01,v: NEXT x: BORDER
7: GO TO 5000
107 IF v=16 AND s<17 AND s>11 AND l=2 THEN FOR x=1 TO 10:
BEEP .005,x*3: NEXT x: CLS : LET score=score+1: GO TO 100
110 IF skok=1 THEN GO SUB 4000
120 BEEP .01,skok*5: NEXT s: PRINT AT 20,0:" "; GO TO 100
1000 LET b=(32-LEN a$)/2
1010 FOR x=7 TO 0 STEP -1: PRINT AT v,b: INK x;a$: FOR f=1 TO 3:
PAUSE 1: NEXT f: NEXT x: RETURN
2000 FOR x=0 TO 7: PRINT AT v,b: INK x;a$: FOR f=1 TO 3: PAUSE 1:
NEXT f: NEXT x: RETURN
3000 DATA 1, 2, 3, 10, 0, 8, -1, 6, -2, 0, 2, 4, 20, -10, 6, 1, 2, 1, 2, 1, 2,
1, 4, 1, 4, 1, 3, 1, 2, 1, 6, 2, 7, 2, 8, 1, 4, 1, 2, 2, 8, 2, 4, -10, 30,
25, 20, 15, 10, 5, 0, -5, -10, -15, -20
3005 DATA -5, -5, -5, -5, -5, -5, -5, 25, 25, 25, 25, 25
3100 DATA 100
4000 LET l= INT (score/10): IF l>15 THEN LET l=15
4005 IF l=1 THEN LET v=v-1: IF v=15-1 THEN LET l=2: RETURN
4010 IF l=2 THEN LET v=v+1: IF v=17 THEN LET skok=0: LET l=1:
RETURN
4020 RETURN
5000 LET a$="GRISLY DISPASSIONATE LETHAL DESTROYER AND KILLER
OF LOVELY MISERABLE EARTHWORMS"
5010 PRINT INK 2: AT 8,2;a$: AT 11,10:"IS DEAD !!!!";
5020 BEEP .003, RND *20: IF INKEY$ ="" THEN GO TO 5020
5030 GO TO 35
6000 DATA 0, 126, 219, 219, 255, 231, 126, 60, 126, 255, 189, 189,
189, 189, 189, 60, 36, 36, 36, 36, 231, 23, 1, 0, 0, 0, 0, 0, 0,
63, 255, 127, 0, 0, 0, 0, 0, 255, 255, 255, 0, 0, 0, 0, 0, 255, 255,
255
```

Na závěr ještě malý anglicko-český slovníček:

grisly - divoký, **dispassionate** - chladný, **lethal** - smrtelný, **destroyer** - ničitel, zhoubce, torpédoborec, **killer** - vrah, **lovely** - roztomilý, rozkošný, **miserable** - ubohý, bídný, **earthworms** - žížala, **dead** - mrtev, **is** - je, **and** - a, **proudly** - právě, **proxima** - vynikající softwarová firma, **present** - uvádět, **press** - stisknout, **to start** - odstartovat, **living** - žijící, **protect** - chránit, **return** - návrat, **please wait** - prosím čekej, **game** - hra, **play** - hrát, **music** - hudba, **program** - program, **idea** - idea, **kill** - zabít.



Nejprve sdělení Sinclair Klubu v Ostrově našim čtenářům:

Vážení čtenáři,

omlouváme se všem, kteří se na náš klub obraceli v období od února do května 1993 se svými dotazy a dostali odpověď na ně mnohdy se značným zpožděním. V této době jsme byly všichni velice pracovně vytíženi a nebylo možné na značné množství dotazů pružně odpovídat.

Při této příležitosti bychom rádi požádali všechny čtenáře, kteří se na náš klub obraceli se svými dotazy, o několik málo laskavostí. Prosíme, nezasílejte nám do klubu žádosti o pomoc s připojováním nejruznějších tiskáren a plottrů. Na dálku se takováto činnost dá jen velice neodpovědně vykonávat a navíc je pro nás velice časově náročná. Někteří tazatelé dokonce požadují informace o připojení zařízení, o němž mnohdy nemáme tušení, že existuje (jako různé tiskárny východní proveniencí apod.) a přičemž trpí vžitou představou, že jsme asi vševědoucí a vlastněme veškerou dokumentaci.

Další velkou skupinou dopisů, které dostáváme, jsou žádosti o zaslání seznamu nabízených programů, či přímo o nahrání nějakého software, schémat připojení a manuálů k programům. Nejsme na tuto činnost, obvyklou možná spíše pro poštovní kluby, zavedeni. A to mnohdy přitom po nás někteří z Vás požadují i pirátské nahrávky programů. Jediná věc, kterou lze na naší adrese objednat a kterou zasíláme, je naše příručka k tepelné tiskárně K 6304.

Často nám také chodí dopisy se žádostí o členství v našem klubu, mnohdy až z míst od nás na stovky kilometrů vzdálených. Velice nás tento zájem těší, ale jak již bylo napsáno, nejsme zavedeni jako poštovní klub a nemůžeme tedy přijímat do klubu členy, kteří se nemohou podílet na jeho aktivní činnosti a zúčastňovat se jeho schůzek. Pouze na schůzkách jsme totiž schopni poskytovat nějaké větší výhody našim členům. Nebudeme samozřejmě proti, pokud bude chtít s námi navázat spolupráci nějaký programátor nebo dobrý hardwarista.

Na závěr ještě jedno důležité upozornění. Vzhledem k množství dotazů, které k nám přicházejí, se vůbec nebudeme zabývat dopisy, ve kterých nebude přiložena ofrankovaná obálka, nebo alespoň 5 Kč ve známkách.

Děkujeme Vám za laskavé pochopení

-Studio 22-
(Sinclair Club Ostrov)

Po přečtení článku "CONSUL klávesnice pro ZXs" v čísle 5/92 Vašeho časopisu, mě zaujal jeden z mnoha inzerátů a lávesnici jsem za 180, tehdy ještě Kčs (nekupte to, když je to tak laciný) koupil. Podle pokynů v článku jsem odeslal disketu se žádostí o popis připojení panu Ručkovi, autorovi úpravy. Bylo to 22. 2. (slovy února) 1993. Od té doby jsem neviděl ani disketu, ani popis připojení. Pan Ručka nereaguje na urgence, kterých jsem mu již několik poslal. Je velice pravděpodobné, že stejně postiženo je více čtenářů Vašeho časopisu.

Proto Vás žádám, asi nejen za svou malíčkovost, o zveřejnění buďto celého popisu úpravy připojení nebo adresy člověka, který by v žádosti o zaslání tohoto popisu byl ochotnější a hlavně rychlejší než pan Ručka. Další možností řešení této situace by mohlo být třeba i zařazení této nahrávky do PUBLIC DOMAIN (pokud nenarazí na odpor autora).

Předem Vám děkuji za pomoc a doufám, že jinak překážející klávesnici se mi podaří připojit ještě letos.

Přeji Vám hodně úspěchů ve Vaší práci a méně, nebo ještě lépe, žádné přispěvatele podobného ražení jako pan Ručka.

Miroslav Hružá
Čechova 1231
676 02 Moravské Budějovice

Naše redakce popis připojení klávesnice CONSUL k ZXs nemá (autor ji neposlal) a nemůže jej tudíž zveřejnit, stejně tak neznáme nikoho (vyjma pana Ručky), kdo by tuto úpravu používal.

Jediné, co pro Vás můžeme udělat je, že otiskneme Váš dopis (což činíme) a vyzvat některého z našich čtenářů (který používá tuto úpravu), aby Vám (nebo nám do redakce) poslal chybějící informace:

Pokud používáte někdo tuto (nebo i jinou) úpravu klávesnice CONSUL a máte k ní popis připojení, pošlete ho, prosím panu Hružovi a také nám do redakce (bude-li to možné, uveřejníme ji).

A ještě krátký dovětek: redakce ZXM nemůže prověřovat nabídky v člancích, které otiskuje - mrzí nás to, ale těžko to můžeme změnit. Stane-li se, že narazíte na podobné potíže, napište nám, ať se o tom mohou dozvědět i další čtenáři, vytvoříme si takovou kolektivní sebeobranu.

Vážená firma PROXIMA,

pomozte mi v rozhodnutí, zda-li si mám pořídit staré ZXs 128 +2,3 či nového SAMA.

Nejasné jsou mi jejich parametry v hudební sféře. Například parametry MIDI (počet sequencerových kanálů, ovládaných nástrojů).

Nevím, zda je zvukový čip SAMA pouze vícekanálové AY (jednoduché hlasitostní křivky, jedna "barva" zvuku) nebo zda překonává AMIGU 500 (600) (6-sample kanálů).

Dalším problémem jsou sequencerové a hudební programy. U 128 určitě existují ale pro SAMA možná ne a nebo jsou příliš drahé.

Zajímalo by mne, jestli SAMácký simulátor ZX Spectra simuluje i 128+3.

PS: Opravdu velká část spectristů (i já) nikdy ZXs 128+2,3 a SAM Coupé neviděla a proto by bylo vhodné udělat v ZXM jejich přehled s vyobrazením a technickými parametry, recenzemi (Myslím všech ZX kompatibilních strojů).

Karel B., Dubí

Váš dotaz jsou poměrně komplexní a ihned Vám na ně odpovědět nemůžeme - tyto informace totiž nemáme.

Váš návrh, abychom se více zabývali ZX kompatibilními počítači je zajímavý a rádi se jej pokusíme uskutečnit.

A následuje výzva pro čtenáře:

Máte-li doma různé ZX kompatibilní počítače nebo nějaké informace o nich, pošlete nám je, anebo ještě lépe, sami napište nějaký článek, který by o tomto tématu pojednával. Největší zájem máme o informace o ZXs 128 +2, 3 nebo Timexu a o SAMovi.

Naše čtenáře by jistě zajímaly i informace o různých diskových systémech či jiných perifériích pro ZX.

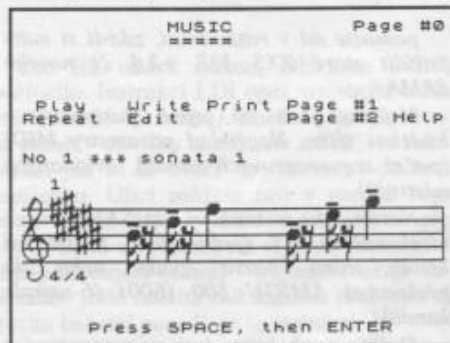
- Vaše redakce -

Hudební editory pro Spectrum 48

Tedy bez použití externích zvukových interface.

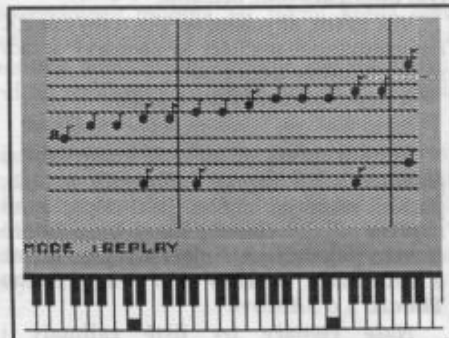
Už autoři ZX SPECTRA si uvědomovali, jak je důležitá přítomnost zvuku v programech a vybavili proto Speccy jednoduchým reproduktorem a příkazem BEEP. I zaplesala srdce stovek hudebníků Spectristů a ti pak dlouho do noci datlovali nekonečné řady BEEPů (i slavný SCALEX takto začínal (stovky BEEPů, aniž by použil FOR - TO - READ - BEEP - NEXT cyklus)). Bylo jasné, že i na Spectru bude brzy k mání hudební editor, jako tehdy byl na ATARI a C64.

Nejčilejší byla firma MUSIC SOFT se svým programem MUSIC TYPEWRITER, což byl na svou dobu (1983) velice slušný produkt. Umožňoval sice jen jednokanálový beepovaný zvuk, ale - budme upřímní - pokud si k počítači IBM PC nekoupíme externí zvukovou kartu, je na tom PCčko nachlup stejně, pouze trochu rychleji



(ostatně i ke Spectru se dá připojit kdecó (možná i Sound Blaster, zkusíte se zeptat pana Drexlera, on je na tyhle věci odborník)). MUSIC TYPEWRITER byl jistě vyroben někým, kdo hudbě rozuměl, protože jsou zde možnosti používat noty různých délek, pomlky, předznamenání a další. Po této stránce byl vlastně překonán až ORFEEM (viz dále).

Dalším editorem, který způsobil doslova revoluci v hudbě na Spectru, je WHAM THE MUSIC BOX 48, který vyrobila firma MARK TIME LTD. Hudba, vytvořená tímto editorem, zazněla v mnoha desítkách her (a některé firmy (zejména španělské) tento editor používají dodnes). WHAM dokáže

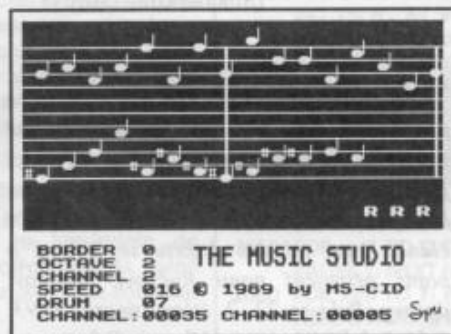


hrát dvoukanálovou hudbu s bicími (pokud zní bicí, nezni nic jiného). I s WHAMEM lze dosáhnout dobrých výsledků (viz Proximácký MUSIC LOGIC - tam máte tolik WHAMovaných melodií, že vás z nich

začne brzy bolet Hlavička). Nevýhodou je, že program neumí používat základní hudebnické "blinky", např. délka všech not v písni je konstantní. Ale možná právě proto tento program uspokojí i nenáročného uživatele. Autor programu - ALEXANDER MARK - k programu přidal několik demonstračních melodií (od skupiny WHAM), aby ukázal možnosti programu.

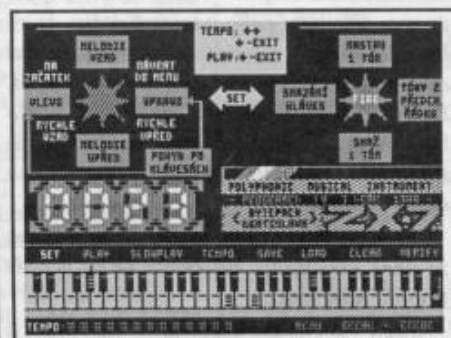
Poté se několik let nic nedělo. Až v roce 1988 se David Dorčák (tehdy Monty Software, později DORLAND) rozhodl vyrobit vícehlasy hudební program. Tak vznikl FOUR CHANNEL MUSIC. I na tomto programu lze najít řadu "hudebních chyb", ale byl to jeden z prvních kroků k vícehláskému programům u nás. Byla vyrobena jak kazetová, tak disketová verze (pro D-40/80). Dnes, když si přehráváme na Spectru samplované zvuky z Amigy, se nám tato rutina může zdát dosti nedokonalá, ale v době svého vzniku tu byl vlastně jen ten WHAM. Pokud nepoužijete příliš vysoké tony, můžete na Four Channel Musicu dosáhnout překvapivě dobrých výsledků.

O rok později se objevil další produkt na bázi WHAMU - firma MS-CID upravila jeden polský hudební program a vzniklo MUSIC STUDIO. Editor byl převzat z



WHAMU, zvuk byl oproti WHAMU vylepšen a zlepšila se i kvalita bicích. Přesto se tento program příliš neujal a čekalo se na něco nového.

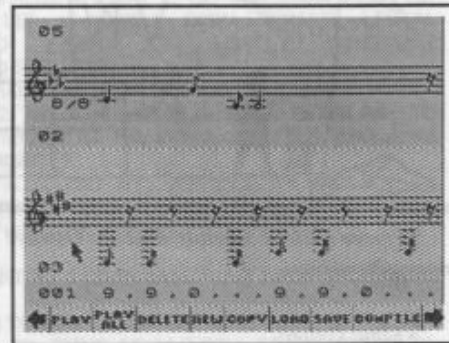
Něco nové přišlo v nabídce slovenské firmy ULTRASOFT pod názvem ZX-7. Program umožňoval tvorbu až osmihlase



hudby. Kvalita zvuku nebyla nejhorší, zásadní chybou ale bylo, že autor (J. Deák) neumožnil tuto hudbu použít mimo program ZX-7. Autor argumentoval tím, že hudba zabírá příliš místa v paměti. To je pravda, avšak k diskovým programům by to nevadilo (po odehrání hudby by se během

několika vteřin naplnila paměť vlastním programem) a leccos by se dalo obejít vhodnou kompresí. Díky absenci této funkce je tedy program prakticky nepoužitelný (rovněž editace byla dosti nepohodlná) - je to škoda, protože ten zvuk byl doopravdy dobrý.

A pak přišel rok 1990 a s ním UNIVERSUMův hudební editor ORFEUS - podle mého názoru dosud nejlepší hudební



editor, který se u nás objevil. PROXIMA vycitila slabá místa na softwarovém trhu a hbitě je zaplnila svými produkty. A právě ORFEUS se stal tím dlouho očekávaným programem - po hudební stránce nemá téměř žádné chyby (pouze je chybička v práci s rychlostmi - a i ta se projevuje pouze v případech, že jsme k jedné rutině připojili data několika melodií), editace je pohodlná (ale vyžaduje určité hudební znalosti) a výsledkem našeho snažení je relovatelná hudební rutina, hrající kvalitní dvouhlasou melodii (se ztišováním) spolu s bicími (nezávisle na melodii). Editačně tento program dosud nebyl překonán (na Spectru 48). Zvukově však (částečně) ano - viz další odstavec.

V roce 1992 vyvinula firma SKUSOFT program MUSIC MAKER 2, který umožňuje používání čtyřbitových samplů z AMIGY. Kvalitního výstupu na Spectru dosáhneme pouze přes AY, ale program hraje i v módu 48 (pochopitelně s redukcí kvality samplů na jednobit) - a pokud se vyhneme použití bicích, výsledek nás mile překvapí. Zejména flétny znějí i na pipátku velice dobře. Na tento program navázala firma CBM programem SAMPLE TRACKER, ale v době psaní toho článku ještě nebyl tento editor v distribuci, takže mi není známo, zda je určen i pro "čtyřicetosmíkáře".

V zahraničních programech se objevily hudby i s jinými hudebními rutinami. Editory těchto rutin se k nám však nedostaly, v některých případech dokonce ani žádné editory neexistovaly (písňe byly psány v datech). Z těchto rutin jmenujme např. rutiny používané Bennem Daglishem (DARK FUSION, SUPERSPORTS aj.), Davidem Whittakerem (LICENCE TO KILL, ATV SIMULATOR aj.), Jasonem C. Brookem (SAVAGE, MAD BALLS aj.) a především nejlepším hudebníkem, který kdy na ZX SPECTRU tvořil - Timem Follinem (AGENT-X, AGENT-X II, RAW RECRUIT, CHRONOS aj.). Je škoda, že se k nám tyto editory nedostaly. Naši

BASIC

Pan Pavel Bláha nám poslal několik krátkých programů v BASICu:

První program pokryje celou obrazovku sítí:

```
20 FOR q=1 TO 2
30 FOR a=0 TO 1
40 FOR d=0 TO 255 STEP 10
50 PLOT 0,175*a: DRAW d,175-350*a:
   PLOT 255,175*a
60 NEXT d
70 NEXT a
80 NEXT q
```

Pomocí druhého programu můžete využitím systémové proměnné CHARS získat zajímavý efekt:

```
20 FOR a=40 TO 0 STEP -1
30 POKE 23606,a
40 PRINT AT 10,10:"ZX Magazin"
50 BEEP .01, INT(RND*40)
60 NEXT a
```

Třetí program vám umožní vytvořit zajímavou čekatí smyčku - pruhy v borderu:

```
20 PRINT "Stiskněte libovolnou klávesu"
30 BORDER 0: BORDER 2: BORDER 4:
   BORDER 6: BORDER 4: BORDER 2:
   BORDER 0: PAUSE 1:
   IF INKEY$ = "" THEN GO TO 30
40 RANDOMIZE USR 3330: CLS: RUN
```

Čtvrtý program je malou modifikací toho předchozího:

```
20 PRINT "Stiskněte libovolnou klávesu"
30 BORDER 1: BORDER 3: BORDER 5:
   BORDER 7: PAUSE 1:
   IF INKEY$ = "" THEN GO TO 30
40 RANDOMIZE USR 3330: CLS: RUN
```

Pátý program namaluje zajímavý pohyblivý obrazec:

```
20 LET b=2: LET x=0: LET y=87
25 OVER 1
30 FOR m=1 TO 8: FOR n=x TO y STEP 8
40 PLOT 39+n,87: DRAW 87-n,-n:
   DRAW 87-n,n: DRAW n-87,n:
   DRAW n-87,-n: NEXT n
50 IF x=0 THEN LET x=87: LET y=0: LET b=b-1:
   GO TO 70
60 LET x=0: LET y=87: LET b=b
70 NEXT m
```

Šestý a poslední program nakreslí kružnici a vybarví ji:

```
20 LET x=127: LET y=87: LET r=80
30 CIRCLE x,y,r
40 FOR l=r TO 1 STEP -1: LET h=SQR(r*r-l*l)
50 PLOT x-h,y-l: DRAW h+h,0
60 PLOT x-h,y+l: DRAW h+h,0: NEXT l
70 PLOT x-r,y: DRAW r+r,0
```

Poznámka korektora: Vybarvení kružnice je založeno na analytickém vyjádření rovnice kružnice: $x^2 + y^2 = r^2$.

Když už jsme v tom analytickém vyjadřování různých křivek, můžeme si napsat obecný program, který nám bude zobrazovat různé analytické křivky v okolí bodu [0,0]:

```
5 LET precision=0.2
10 CLS
20 FOR x=-12.8 TO 12.7 STEP precision
30 FOR y=-8.8 TO 8.7 STEP precision
40 IF "???" THEN PLOT 10*x+128,10*y+88
50 NEXT y
60 NEXT x
```

Proměnná **precision** určuje, jak hustá má být síť bodů, které jsou zkoumány, čím nižší číslo (nejméně jedna, menší krok nemá smysl), tím delší doba na vykreslení obrazu. Nejprve zadávejte **precision** vyšší a když budete chtít získat kvalitní obrázek, použijte jedničku.

Protože BASIC je pomalý, zkuste použít nějaký vhodný kompilátor - TOBOS nebo Hisoft Basic Compiler.

A ještě nějaké ty rovnice, abyste vůbec mohli začít:

```
1) ABS ( x2 + y2 - 25 ) < 1
2) ABS ( x + y2 ) < 1
3) x2 + y2 < 25
4) x2/2 + y2 > 25
5) ABS ( x - y ) < 3
6) x < y2
7) COS x - COS y > 0.5
8) COS x/SIN y > 0.5
```

A můžete zkusit i další, dejte si jen pozor, abyste nedělili nulou nebo příliš malým číslem.

Dále si ukážeme podprogram, který bude simulovat jakousi **šipku** v BASICu, budeme s jeho pomocí moci ukázat a zvolit libovolný znak na obrazovce:

```
1 GO TO 1000
5 REM Podprogram "ŠIPKA"
10 LET r=0: LET s=0
20 LET a=r*32+s+22528
30 LET p=PEEK a
40 POKE a,150
50 IF INKEY$="" THEN GO TO 50
55 LET a$=INKEY$
60 POKE a,p
70 IF a$="p" THEN LET s=s+1: IF s>31 THEN
   LET s=0
80 IF a$="o" THEN LET s=s-1: IF s<0 THEN
   LET s=31
90 IF a$="a" THEN LET r=r+1: IF r>23 THEN
   LET r=0
100 IF a$="q" THEN LET r=r-1: IF r<0 THEN
   LET r=23
110 IF a$<>"m" THEN GO TO 50
120 RETURN
```

```
1000 GO SUB 10
1100 IF r<22 THEN PRINT AT r,s:""
1150 IF r>21 THEN BEEP 0.5,-30
```

1200 GO SUB 20: GO TO 1100

Podprogram se dá volat od řádku 10 (šipka se objeví na obrazovce v levém horním rohu) nebo od řádku 20 (šipka se objeví tam, kde byla naposledy).

Program můžete využít například pro volbu nějaké možnosti z menu, ovládání znakového generátoru nebo podobnou činnost.

Poslední prográček, který si dnes ukážeme bude velice jednoduchý hudební nástroj s pamětí a možností přehrávání melodie:

```
1 RETURN
32 BEEP .01,12: RETURN
98 BEEP .01,7: RETURN
99 BEEP .01,4: RETURN
100 BEEP .01,3: RETURN
103 BEEP .01,6: RETURN
104 BEEP .01,8: RETURN
106 BEEP .01,10: RETURN
109 BEEP .01,11: RETURN
110 BEEP .01,9: RETURN
115 BEEP .01,1: RETURN
118 BEEP .01,5: RETURN
120 BEEP .01,2: RETURN
122 BEEP .01,0: RETURN
900 CLEAR 29999: LET store=30000
1000 LET a= CODE INKEY$
1010 POKE store,a
1020 LET store=store+1
1030 GO SUB a
1100 GO TO 1000
2000 LET store=30000
2010 GO SUB PEEK store
2020 LET store=store+1
2030 GO TO 2010
```

Program se spouští od řádku 900 (RUN 900), můžete si zahrát, správně reagují klávesy Z X C V B N M SPACE (bílé klávesy na piánu) a S D G H J (černé klávesy). Program při hraní ukládá váš výtvar do paměti a může vám jej kdykoliv přehrát.

Až se nabažíte hraní a budete si chtít poslechnout to, co jste vytvořili. Stiskněte BREAK a zadejte příkaz GO TO 2000, pak už stačí jenom poslouchat.

Všimněte si, že čísla řádků s příkazy BEEP odpovídají přesně kódům jednotlivých kláves (při hraní nesmíte mít zapnutý CAPS LOCK).

Program samozřejmě můžete dále upravovat a vylepšovat (přidat nějakou klávesu, s jejíž pomocí se hra přeruší, možnost přehrávky nabízet programem, smazání staré nahrávky, uložení na kazetu (disketu) a načtení zpátky).

Program využívá paměť od 30000 až do konce pro ukládání dat, nedávejte si tam tedy žádná další data nebo programy, pokud to přeci jen potřebujete, musíte upravit práci s proměnnou **store**.

programátoři však tento nedostatek zčásti nahradili - UNIVERSUMův ORFEUS má zvuk dosti podobný Whittakerově rutině, DORLAND vyrobil konverzní program pro převod dat z WHAMU 128 do rutiny ze hry ZANTHRAX a zajisté budou vznikat další kvalitní zvukové programy, které se obejdou

bez AYčka (vždyť Follinovy hudby na 48 jsou tím nejlepším, co lze na Spectru slyšet - po hudební stránce jsou dokonce dokonalejší než některé AYčkové melodie).

Tolik tedy o čtyřicetiosmičkových hudebních editorech. Někdy přistě si povíme něco o editorech pro ZX8128 nebo

48 s AY 3-8910/12, o dalších programech na Spectru, které se týkají tvorby zvuků a o tom, co na ZX81 v oblasti zvukového software není a co tedy čeká třeba právě na vaše programátorské umění...

- Jen a jen pro ZX81 napsal S C A L E X -

TABULKOVÉ PROCESORY

Právě začínáme...

Určitě jste už mnohokrát slyšeli termín **Tabulkový procesor** nebo **Spread sheet** (což je totéž ale anglicky a používají to někteří lidé, když chtějí na ostatní udělat větší dojem). Nemusíte ale vědět, co se od tohoto typu programů dá očekávat - tak si o tom něco povíme:

Základem tabulkového procesoru je tabulka (poměrně překvapivé zjištění). Může mít libovolný (v jistých mezích) počet řádků a sloupců - záleží na vás, jakou tabulku právě potřebujete, takovou si vytvoříte.

Do každého políčka tabulky můžete zapisovat buď číslo, text nebo vzorec. Abychom mohli každé políčko v tabulce identifikovat, použijeme označení sloupců a řádků - každé políčko bude dáno jednoznačně sloupcem a řádkem, v jejichž průsečíku se nachází. Sloupce budeme označovat písmeny a řádky čísly, označení políčka bude vždy nejprve písmeno sloupce a hned za ním (bez mezery) číslo řádku, například první políčko v tabulce bude **A1**.

Zkusíme si pro názornost takovou tabulku nakreslit (bude mít tři sloupce a čtyři řádky):

	A	B	C
1	Cena bez DPH	DPH	Cena s DPH
2	1 000.00	23	1 230.00
3	400.00	5	420.00
4	3 500.00	23	4 305.00

Tabulka v prvním řádku obsahuje popisy sloupců (texty), v dalších řádcích pak vždy v prvních dvou sloupcích čísla a v posledním sloupci vzorec.

Pro ukázkou jsem zvolil velice jednoduchý příklad použití - výpočet ceny zboží s Daní z Přidané Hodnoty. V prvním sloupci jsou ceny bez DPH, ve druhém je sazba daně a ve třetím se ukazuje výsledek, který je vypočítán podle vzorce:

Cena bez DPH * (sazba DPH + 100)/100

V tabulce jsou ovšem vzorce napsány jinak, takto:

Na políčku C2 **A2 * (B2 + 100) / 100**
 Na políčku C3 **A3 * (B3 + 100) / 100**
 Na políčku C4 **A4 * (B4 + 100) / 100**

Možná se Vám zdá práce vynaložená na vytvoření tabulky a získání výsledků mnohem vyšší, než kdyby jste si to spočítali (ručně) pomocí kalkulatoru - vždyť by stačilo pouze vynásobit cenu zboží bez DPH číslem 1.23 nebo 1.05 - máte samozřejmě pravdu. Jenomže co když budete chtít počítat ceny u dalších druhů zboží, s kalkulačkou budete znovu násobit.

S tabulkovým procesorem stačí, když změníte čísla v prvním sloupci tabulky a případně sazbu daně - ostatní už zařídí tabulkový procesor sám, navíc máte možnost si výsledek vytisknout na tiskárně.

Tabulkový procesor ale zvládne i složitější problémy - co kdybychom ještě chtěli, aby v tabulce byla vypočtena také celková cena zboží a velikost daně z přidané hodnoty, tabulka by mohla vypadat nějak takto:

	A	B	C
1	Cena bez DPH	DPH	Cena s DPH
2	1 000.00	23	1 230.00
3	400.00	5	420.00
4	3 500.00	23	4 305.00
5	DPH celkem		1 055.00
6	Celková cena		5 955.00

Jak vidíte, teď už se z tabulky dozvíte mnohem víc informací a přitom musíte zadávat stále stejně vstupních dat (jenom ceny bez DPH a sazby DPH).

Asi přemýšlíte, jak vypadají vzorce na pátém a šestém řádku ve sloupci C, jsou opět poměrně jednoduché:

Políčko C6 .. **sum(C2:C4)**
 Políčko C5 .. **sum(C2:C4) - sum(B2:B4)**

Ve výrazu se objevila nová funkce, funkce **sum**. Její činnost je jednoduchá - jde o součet čísel na políčkách, které se nacházejí ve sloupci nebo v řádku mezi políčky, které jsou uvedeny v závorce (včetně). Všimněte si, že ve funkcích **sum** se vždycky shodují sloupcové souřadnice - obvykle není možné uvést současně různý řádkový a různý sloupcový index - jakési sčítání v oblasti).

Když si tuto tabulku podrobněji prohlédnete, může se Vám zdát, že tam nejsou v některých řádcích všechny sloupce, to je jen takový trik - chybí tam totiž pouze oddělovací čára, dvě políčka tam ve skutečnosti doopravdy jsou - vidíte, že text je napsán jenom ve sloupci A.

Možná vás napadne, že tabulka, do které se dají zapsat pouze tři různé položky je poněkud málo užitečná. Ale i zde si tabulkový procesor bez problému poradí, nikde není psáno, že mezi řádky 4 a 5 nemůže být ještě spousta dalších řádků. Asi vás napadne, že přece nebudete měnit součtové vzorce podle počtu položek v tabulce - to opravdu nebudete muset, všechny tabulkové procesory mají tu vlastnost, že hodnota prázdného (nevyplněného) políčka v tabulce je nula, tudíž součet se nezmění.

Když se nad příkladem zamyslíte, dojdete k názoru, že by neškodilo mít v

tabulce ještě další informace: nějaký popis zboží, počet kusů a případně jméno a adresu zákazníka, nebo také svoji adresu a jméno. Tabulka by tedy musela mít další sloupce, podobně by se musely upravit i výrazy v některých políčkách.

Na další možnosti použití tabulkového procesoru jistě přijdete sami, náš příklad ještě později trochu rozšíříme.

Poněkud obecněji...

To, co jsme si právě popsali byl jakýsi "obecný" tabulkový procesor. Ve skutečnosti se různé tabulkové procesory liší možnostmi a schopnostmi - něco si o nich povíme:

1) **Matematické schopnosti** - sfla (co všechno lze vyjádřit) matematických operací tabulkového procesoru se může lišit, základní aritmetické operace (sčítání, odčítání, dělení, násobení) však mají všechny. Navíc mohou mít speciální matematické funkce (sin, ln atd.). Další rozšíření mohou obsahovat různé funkce, které souvisí s tabulkou (sumace přes řádky nebo přes sloupce), případně podmíněné výrazy nebo i jednoduché programovací jazyky (každý programovací jazyk je v podstatě v jistém smyslu vzorec). Například naše funkce **sum**, která která provádí sčítání vybraných políček, může být rozšířena na jakousi funkci:

sum (I=1, 20, 2) (vzorec(I))

Jejíž smysl je ten, že dosazuje do I čísla od 1 do 20 s krokem 2 (tedy 1, 3, 5, 7, ...)

19) a pro každé I znovu počítá výraz **vzorec(I)**, v němž se vyskytuje. Jednotlivé výsledky sčítá a celkový součet je pak hodnota tohoto výrazu (pokud znáte program Apollon, můžete se tam s podobnou funkcí setkat).

2) **Formátovací možnosti** - formátem tu chápou to, jak vypadá tabulka, a to, jak vypadá zápis čísel v tabulce. U starších programů je velikost tabulky pevně daná, u programů novějších je volitelná. Nejjednodušší programy mají pevně danou šířku sloupce (např. 8 znaků), lepší programy umožňují šířku sloupců nastavit (podle jejich významu, například počet kusů asi nemá smysl zapisovat do 8-mi znaků, stejně tak popis zboží se do 8-mi znaků vejde jen těžko).

Další schopnosti, které programy nabízejí v různé míře, jsou možnosti formátovat obsah políček, zvláště těch, které jsou doplňovány vzorci - od jednoduchého zarovnání čísla v obecném tvaru doleva až po možnost volit si zarovnání (doprava, doprostřed), formát čísla (počet platných a desetinných míst).

3) **Grafické možnosti** - některé z tabulkových procesorů pracují v textovém režimu počítače, jiné využívají jeho grafické možnosti, část dokáže pracovat v

režimech obou - textový režim je používán pro svou rychlost, grafický pak pro lepší přehlednost tabulky. U lepších programů můžete volit, kde mají být v tabulce čáry, případně jejich tloušťku nebo barvu. U těch nejlepších můžete pro políčka definovat barvu podkladu, velikost, typ, tvar, velikost a barvu písma.

4) **Složitost dat** - jednoduché programy pracují vždy právě s jednou tabulkou, složitější dokáží pracovat se souborem tabulek - s jakousi kartotékou tabulek. Matematické výrazy se pak mohou odkazovat i do jiných tabulek, než je ta, na které se vyskytují. Možnosti tohoto rozšíření se ozřejmí, když je budeme aplikovat na náš příklad - v souboru dat budeme mít jednotlivé objednávky a jako poslední budeme mít kartu, která bude obsahovat vzorec pro sečtení celkové výše DPH, celkové výše cen (příjmů) a jejich rozdíl (celkové výše cen bez DPH) - získáte tak dokonale přehled o výsledcích prodeje (nákupu).

Se složitostí dat také souvisí maximální velikost datového souboru, který lze programem zpracovávat - u lepších programů je obvykle v operační paměti jenom část zpracovávaného souboru dat, zbytek je uložen na disku nebo disketě.

5) **Rozšiřující funkce** - u kvalitních programů máte možnost rozšířit výstup programu o kreslení grafů - obvykle je k dispozici velké množství různých grafů (sloupcový, čárový, kruhový, prostorový, ... a spousta dalších).

6) **Databázové funkce** - funkce, které přímo nesouvisí s hlavními úkoly tabulkového procesoru - třídění řádků (sloupců) podle vybraného políčka, třídění tabulek (v souboru tabulek) podle vybraného políčka, vyhledávání textu v rozsáhlé tabulce nebo vyhledávání tabulky se zadaným textem.

7) **Přítulnost programu** - sem bych zařadil jednoduchost obsluhy a množství nabízených editačních funkcí (např. blokové operace, možnost vyplnit políčka v určené oblasti tabulky vzorcem, který se bude "inteligentně" upravovat podle toho, do jakého políčka má být vložen, rychlost pohybu po tabulce, čitelnost, použitelné způsoby obsluhy - menu, horké klávesy, použití myši, atd.).

8) **Cena** - nejdražší nemusí být vždy nejlepší, ale naopak přílišná láce se může vymstít také.

9) **Nároky na hardware** - ne každý program poběží dostatečně rychle nebo bude vůbec použitelný na různých počítačích (hlavně u PC kompatibilních). Budeme-li se držet trochu při zemi a mluvit o ZX Spectru, může vás zajímat, jestli dokáže spolupracovat s disketovou jednotkou nebo ne, případně s jakým systémem.

10) **Možnosti výstupu na tiskárnu a exportu dat** - které tiskárny lze s programem využívat a jestli vůbec nějaké. V jakém formátu lze data z tabulkového procesoru přenášet do jiných programů (textový editor, databáze).

Zpátky ke Spectru...

Pro našeho miláčka bylo napsáno několik (tuším, že dva, ale možná i víc) tabulkových procesorů. První a nejstarší je:

VU-CALC

pochází z roku 1982, vyrobila jej firma PSION. Program umožňuje pracovat s jednou tabulkou poměrně značných rozměrů. Políčka tabulky jsou stejně velká - 7 znaků.

Pro ty, co mají tento program zahrabaný někde doma na kazetě a chtěli by jej použít nebo si alespoň vyzkoušet, co se s tabulkovým procesorem dá dělat, vypíšu seznam příkazů (to se dozvíte z programu) s parametry (ty už se tam nedovíte) a stručný popis:

Blanc - smaže běžnou položku

Calculate - provede přepočítání tabulky, použijte při změně vzorce nebo dat v tabulce, provádí se zleva doprava a shora dolů

Edir - edit - umožňuje měnit vzorec na políčko označeném kurzorem

Format, c, f, j - určuje formát sloupce

c - číslo sloupce nebo "A" pro celou tabulku

f - formát čísla, můžete použít znaky I (integer), \$ (dvě desetinná čísla), G (obecný formát)

j - způsob zarovnání - L levý okraj, R - pravý okraj

Go, rc - přechod na dané políčko tabulky, rc znamená například A3

Load - uložení tabulky na pásku (je prováděno basicem, lze snadno přeměňovat na disketu)

Print - tisk obrazovky na tiskárnu, (pravděpodobně ZX Printer)

Quit - ukončení programu nebo smazání celé tabulky

Repeat, rc, f:l - opakuje obsah políčka rc do oblasti určené políčky f (levé horní) a l (pravé dolní). Pro opakování vzorců viz dále.

Save - načtení tabulky z pásky

T, r1, r2 nebo **T, c1, c2** - přesun obsahu z jednoho řádku (sloupce) do druhého, nelze přenášet obsah řádku do sloupce a naopak

Funkce stačí zadávat prvním písmenem, nezapomeňte na oddělovací čárky.

Vkládat do políček tabulky můžete následující:

čísla - přímo

vzorce - přímo

text - nejprve vložte úvozovku

příkaz - vložte nejprve znak "#", na horní řádce se objeví seznam možných příkazů

Vzorce:

Používat můžete čísla, obsahy políček (např. A1, D3, AA33 - pozor, řádky jsou písmenkované, sloupce číslovány - tedy opačně než je obvyklé) a jednoduché aritmetické operátory (+, -, /, *) a závorky.

Při použití funkce **Repeat** se odkazy na položky mění s posunem podél řádku nebo sloupce (vzorec A3*B4 na políčko A1 bude na políčko A2 přenesen jako A4*B5 a na políčko B1 jako B3*C4, odkazy jsou tedy chápány relativně. Pokud chcete, aby se vzorec mohl odkázat na pevnou položku, vložte před ní znak \$ (\$A1 znamená A1 při libovolném opakování).

Program má také obdobu funkce **sum** z úvodního příkladu - zapisuje se jako vzorec ve tvaru: **&f, l**, kde f je počáteční políčko a l je koncové políčko oblasti, sumaci nelze kombinovat s žádnými jinými aritmetickými operacemi - chcete-li použít někde ve výpočtu výsledek sumace, musíte ho uložit do nějakého políčka a pak se na něj odkázat (políčko použijete jako proměnnou - dejte si pozor na pořadí počítání vzorců při použití funkce **Calculate**).

Pokud dojde při výpočtu k chybě a program s vrátí do BASICu, vložte příkaz **GO TO 9000**.

OMNICALC

Omnicalc je modernější program než VU-CALC. Má mnohá vylepšení, tak tedy alespoň stručně:

- matematické výrazy mohou obsahovat veškeré matematické operátory a funkce, které může používat BASIC a to se stejnou syntaxí (dokonce i NOT, AND a OR).

- možnost vytvářet formuláře
- větší editační možnosti
- lepší ošetření chyb
- možnost vytvoření sloupcového grafu z hodnot obsažených v tabulce

CALCULUS

Až tenhle program dokončím, měl by strčit do kapsy oba dva předchozí. Jeho hlavní přednosti budou (na 90%):

- bude umět pracovat se soubory tabulek
- soubor na disku (velikost souboru bude omezena jen kapacitou diskety) - tato část je již hotova

- přehledný a dobře čitelný tisk na obrazovku, proporcionální tisk

- volitelná šířka sloupce
- volitelné barva a orámování políčka
- grafy
- plná podpora češtiny
- množství editačních funkcí
- výstup na tiskárnu

- export a import dat v textovém formátu pomocí sekvenčního souboru

- volitelná velikost tabulky
- volitelný formát výstupu čísel
- vytváření předloh
- databázové funkce: třídění, vyhledávání
- obsluha pomocí roletových menu nebo horkých kláves

Doufám, že ještě přidám něco navíc, jak to nakonec dopadne, se budete moci přesvědčit sami (a já doufám, že poměrně brzy).

- UNIVERSUM -

Jak na zvukový interface s obvodem AY (1)

Tento článek přináší některé zajímavé náměty k zamyšlení pro ty z vás, kteří používají zvukový interface s obvodem AY-3-8910(12) a pronikli tak trochu do tajemství strojového kódu.

Indikace výstupu zvuku obvodu - je-li připojen obvod AY na sběrnici počítače, je možné jeho přítomnost velice snadno testovat. Většina starších her, využívající AY, jej vlastně ani netestuje, ale kontroluje ROMku a paměťové stránky Spectra 128, ve kterém se AY poprvé objevil, což vede k zavádějícím závěrům. Tento způsob je pro nás nevyhovující. Přímé testování obvodu je přitom vcelku jednoduché. Testování, které si popíšeme používá třeba hra DOUBLEDASH. Je založeno na zápisu a zpětném čtení portů AY. Program nejprve pošle do registru obvodu určitou hodnotu potom ji načte z registru zpět. Když se hodnoty rovnají, lze říci, že je AY připojen (prostinké, že). Celý program je dlouhý 18 bytes a lze jej umístit kdekoliv v paměti.

Program TEST AY:

```
TEST_AY ld bc,65533 ;adresa portu AY
        ld a,7      ;registr řízení kanálů
        ld d,63     ;všude ticho
        out (c),a   ;odešl registr
        ld b,191    ;bc=49149
        out (c),d   ;odešl hodnotu AY
        ld b,255    ;bc=65533
        in a,(c)    ;načti registr 7 AY
        cp d        ;porovnej s původní
                ;platí z=AY připojeno
```

Indikace výstupu signálu na AY - v některých DEMO programech (např. FX SOUND 2) jste si jistě všimli, že hodnota výstupu na AY je signalizována na obrazovce pomocí sloupcového ukazatele. Podívejme se tedy blíže, jak toho lze dosáhnout. Podprogram, kterým si můžete vhodně doplnit vlastní hudební programy, používá druhý mód přerušení a je schopen zahrát tón a vykreslit ukazatele za dobu menší než odpovídá jednomu přerušení (méně než 1/50s). Takže lze pohodlně dosáhnout toho, že třeba při psaní programu v BASICu vám k tomu bude hrát AY hudba a zároveň se bude kreslit ukazatel výstupu na obrazovce. Jedno malé upozornění: protože tabulka pro přerušení je umístěna v ROM ZX81, pak pokud budete stránkovat do jiné ROM (např. pracovat s D40), musíte nastavit mód přerušení na IM1, nebo tabulku přerušení umístit na vhodné místo v do RAM (velikost tabulky je 256 krát číslo 255 = (FF)_H). Program je pro lepší pochopení komentovaný.

K jeho napsání mne vyprovokoval článek AY-PXT-METR ve FIFO č.16, ze kterého si ale příklad raději neberte, protože tisk ukazatele realizuje neprakticky přes RST 16. Takovýto program potom zbytečně zaměstnává procesor a není čas na další (tvorivou) činnost:

Program indikátor:

```
org 55000 ;překládej od 55000
jp INIT   ;skoč na inicializaci

START push hl ;uschovej registry
        push de
        push bc
        push af
        push ix
        push iy
        call "strm" ;zavolej hraní tónu
        ld a,10     ;přečti hlasitost
        call ABC    ;všechny kanály
        ld (65000),a ;ulož hodnoty pro
        ld a,9      ;pozdější použití
        call ABC    ;zapsují se pozpátku
        ld (65001),a ;protože se kreslí
        ld a,8      ;zleva
        call ABC
        ld (65002),a
        ld hl,22527 ;adresa levého dolního
        push hl     ;rohu (pixels)
        ld de,65000 ;první kanál
        call TISK    ;zobraz ukazatel
        pop hl
        dec hl      ;sniž adresu ukazatele
        push hl     ;uschovej adresu
        inc de      ;další kanál
        call TISK    ;zobraz
        pop hl      ;obnov adresu
        dec hl      ;sniž adresu ukazatele
        push hl
        inc de      ;další kanál
        call TISK    ;zobraz
        ****
        pop hl      ;obnov všechny
        pop iy      ;registry
        pop ix
        pop af
        pop bc
        pop de
        pop hl
        jp #38      ;vrať se přes Mask-Int

ABC ld bc,65533 ;port pracovního registru
    out (c),a   ;odešl hodnotu
    in a,(c)    ;přečti hlasitost
    ret        ;vrať se

TISK ld b,16    ;16 stupňů hlasitosti
    ld c,126    ;tvar "čárky"
    ld a,(de)   ;čti hlasitost
    cp 0        ;je nulová?
    jr z,SMYC2  ;ano-smaž čárku
    ld a,c      ;do A tvar
    ex de,hl    ;sniž ukazatel
    dec (hl)    ;hlasitost
    ex de,hl
    ld (hl),a   ;vytiskni/smaž čárku
    call UPHL   ;posuň se nahoru
    call UPHL   ;vynech mikrořádek
    djnz SMYC   ;opakuj 16x
```

```
ret ;vrať se

UPHL ld a,h     ;tento podprogram
    dec h       ;je z knihy Assembler
    and 7       ;a ZX spectrum
    ret nz      ;první díl
    ld a,l
    sub 32
    ld l,a
    ld a,h
    jr c,UPHL2
    add a,8
    ld h,a
    cp 64
    ret nc
    ld h,87
    ret

UPHL2

INIT ld hl,23295 ;pravý dolní roh barev
    di          ;nenech se rušit
    push hl
    call VYPLN  ;vybarví sloupec
    pop hl
    dec hl     ;posun na další sl.
    push hl
    call VYPLN ;vybarví
    pop hl
    dec hl     ;posuň
    call VYPLN ;vybarví
    ld a,58    ;ukazatel adresy
    ld l,a     ;nastav do ROM
    im 2       ;druhý mód přerušení
    ld a,24    ;hodnoty skoků
    ld (65535),a
    ld a,195   ;potřebné pro
    ld (65524),a
    ld hl,START
    ld (65525),hl
    ei         ;povol přerušení
    ret        ;vrať se

VYPLN ld a,4    ;poprvé zeleně
    ld (hl),a
    ld bc,32    ;další atribut je
    or a        ;o 32 bytes dále
    sbc hl,bc
    ld a,4      ;komu se nelení
    ld (hl),a   ;tému se zelení
    or a        ;vynuluj CARRY
    sbc hl,bc   ;odečti
    ld a,6      ;jednou žlutě
    ld (hl),a
    or a
    sbc hl,bc
    ld a,2      ;jednou červeně
    ld (hl),a
    ret        ;vrať se

KONEC
```

Přehrávač více melodií - může se stát, že bude v jednom bloku více melodií a Vy chcete, aby se měnily stiskem nějakých kláves (třeba 1-5). Pokud jsou melodie dvě nebo tři, stačí k tomu jednoduchý podprogram. Pokud je ale melodií více (RENEGADE 3 jich má devět), je takovýto podprogram žrout paměti! Takže nezapomeňte, že vše je třeba dělat s mírou. Následující prográmek je pokračováním

toho předchozího. Jistě jste si všimli hvězdiček na řádcích s instrukcemi POP HL a JP #38. Tuto část programu mezi hvězdičkami (obnovení registrů) smažte a vložte před návěstí ABC následující podprogram. Podprogram test kláves:

```
KEYSTEST ld a,255 ;hodnota pro úpravu
          ld (PRCTI+1),a ;číslo melodie
          ld a,247 ;přečti klávesy
          in a,(254) ;1-5
          call PRETV ;přetvoř
          jr z,DALSI ;kl. 1-5 nestisknuty,
          jp ZMEN ;pokračuj změnou
```

```
DALSI ld a,4 ;hodnota pro úpravu
       ld (PRCTI+1),a ;číslo melodie
       ld a,239 ;přečti klávesy
       in a,(254) ;6-0
       call PRETV ;přetvoř
       jr z,DALSIZ ;6-0 nejsou, vrať se
       ld b,0 ;klávesy 6-0 jsou
       rra ;čteny obráceně (0-6)
       rl b ;musí se prohodit
       rra
       rl b
       rra
```

```
rl b
rra
rl b
rra
rl b
ld a,b
jp ZMEN
```

```
DALSIZ pop hl ;obnov registry a
        pop ly ;vrať se přes
        pop lx ;Mask-Int
        pop af
        pop bc
        pop de
        pop hl
        jp #38
```

```
PRETV cpl ;prohod 1 a 0, ponech
       and %11111 ;jen pět kláves
       ret ;vrať se
```

```
ZMEN ld b,a ;přečti bitové číslo
      xor a ;na dekadické
```

```
ZMENZ rr b ;na systém přepočtu
       lnc a ;zkuste přijít sami
```

```
PRCTI add a,0 ;podle portu uprav
      cp 0 ;pokud je shodné s
      jr z,DALSIZ ;posl., neměň hudbu
```

```
cp 1 ;překročen poč. mel.?
      ;(max. číslo hudby+1)
      jr nc,DALSIZ ;ano-neměň
      ld (POSL+1),a ;ulož poslední hudbu
      call "změna melodie"
      jr DALSIZ ;skončí
```

O změnu melodie se už musí starat player (přehrávací program) třeba novou inicializací melodie ze SOUND TRACKERu. Co se týče přehrávačů (player je přece jen lepší výraz), jsou u různých her udělány různě. Třeba programy Fr. Fuky ukládají šest čísel, starty jednotlivých kanálů. Naopak zahraniční produkty mají (většinou) podprogram, kam se vstupuje s číslem melodie v nějakém registru. Pokud je to reg. A, je možné použít uvedený podprogram bez úprav. Všechny uvedené programky nejsou zdaleka nejdokonalejší, jistě si je nějak vylepšíte. Programování hudebních interface zdar. A zase někdy příště!

-TRITOLSOFT-

Ovládání D40

Ovládání D40(80) ze strojového kódu popsal celkem podrobně Universum ve svém článku v ZX magazínu 1/93. Jeho způsob je vhodný na úpravu programů, které pracují s kazetovou hlavičkou. Pokud ale chcete ukládat nebo číst soubory rovnou (bez ukládání kazetových hlaviček) stačí vám napsat mnohem kratší program.

Nejprve SAVE:

```
ULOZ di ;zakaž přerušení
      rst 0 ;přestráknij do ROM D40
      call 7311 ;přečti formát disku
      ld hl,JMENO2 ;adresa jména s příponou
      ld de,16010 ;adresa uložení jména v RAM D40
      ld bc,11 ;deset znaků jména a jeden přípona
      ld ir ;přenes
      ld de,6912 ;délka souboru
      ld hl,16384 ;začátek dat
      ld (15992),hl ;ulož do budoucí hlavičky
      ld (15988),hl ;ulož pro potřeby systému
      ld a,1 ;povol případně smazání
      ld (15970),a ;souboru se stejným jménem
      call 6656 ;ulož
      ld a,79 ;povol další přeštránkování
      ld (16119),a
      jp 737 ;zastav motor a přeštráknij zpět
JMENO2 defm "obrazek" ;deset znaků jména
        defb 0,0,0 ;doplnit nulami do 10 zn.
        defm "B" ;přípona
```

Při prvním volání je nutné použít podprogram ze ZXM 1/93, nebo zadat z BASICu POKE #247,79. Další "POKE" si již program zajistí sám. Program NETESTUJE přítomnost souboru se stejným jménem (prostě jej přepíše!), taktéž není ošetřena chyba "Drive is not ready" (pokud jsou přepsány systémové proměnné, program se zhroutí).

A teď LOAD:

```
LOAD di ;zakaž přerušení
      rst 0 ;přeštráknij
      call 7311 ;přečti formát a adresář
      ld hl,JMENO ;adresa uložení jména
      ld de,16010 ;adresa uložení jména v RAM D40
      ld bc,11 ;deset znaků jména a jeden přípona
      ld ir ;přenes
      call 8491 ;je hledaný soubor na disku
      jr nz,ERROR ;není-chyba
      ld (15986),hl ;v HL je adresa hlavičky, ulož ji pro sys'ému
      push hl ;přenes adresu do IX
      pop ix
      ld e,(ix+11) ;do DE dej délku souboru
      ld d,(ix+12)
      ld ix,16384 ;adresa začátku ukládání dat do paměti
      call 6574 ;načti data
      ld a,79 ;hodnota pro další přeštránkování
      ld (16119),a ;ulož ji
      jp 737 ;zastav motory a přeštráknij do ROM spectra
ERROR call 9526 ;zastav motory
ERROR2 ld a,127 ;čekej na stisk "SPACE"
       in a,(254)
       rra
       jr c,ERROR2
       jr POKRAC ;pokud se číst znovu
JMENO defm "obrazek" ;deset znaků jména
       defb 0,0,0 ;doplnit nulami
       defm "B" ;přípona souboru
```

Podmínky pro tento podprogram jsou stejné jako v předchozím případě (první volání a systémové proměnné). Pokud na disku hledaný soubor není, motory jsou zastaveny a čeká se na stisk "SPACE" po kterém následuje opakování čtení.

Oba podprogramy jsou psány tak, aby zabíraly co nejméně místa v paměti. Podprogram LOAD jde ještě zkrátit (délku souboru zadávat přímo a odstranit test přítomnosti souboru-nedoporučuje se). Naopak, přidání některých testů by jistě neškodilo (snad jen obsazení paměti), ale to už ponechávám na Vaší tvůrčí invenci.

-TRITOLSOFT-

Máte rádi vlaky? (aneb hledání nejkratší cesty)

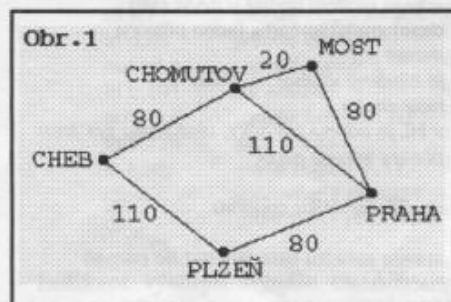
Úvodem

Jestli jste někdy zavítali na Masarykovo nádraží (tam jsem to viděl já, ale může to být i jinde), mohli jste si všimnout malého informačního automatu - zadali jste výchozí a cílovou stanici a čas odjezdu a dostali jste čas příjezdu do cílové stanice, také jste mohli zadat kdy tam chcete dojet a dozvěděli jste se, kdy musíte vyrazit, abyste to stihli.

Jestli jste uvažovali o tom, jak to ten automat vlastně dělá, zajímalo vás přesně to, co je obsahem tohoto článku. Povíme si něco o tom, jak asi takový automat pracuje, ukážeme si postup na příkladu a napíšeme i jednoduchý demonstrační program v BASICu.

Příklad pro začátek

Nejprve si problém hledání nejlepšího spojení trochu upravíme a budeme hledat nejkratší cestu z jednoho místa do druhého (asi nemusím zdůrazňovat, že mezi časovou náročností cesty a její délkou je přímá úměra). Na obrázku 1 vidíte jednoduchý problém - Potřebujeme najít nejkratší cestu z Prahy do Chebu (po kolejích) a můžeme používat v obrázku nakreslené tratě:



Když si obrázek dobře prohlédnete, zjistíte, že nejkratší cesta vede přes Most, jenže jak na to přijde počítač, vždyť obrázek nevidí?

Grafy

Nyní si zkusíme to, co je na obrázku nějakým způsobem reprezentovat tak, aby s tím mohl pracovat počítač.

Nejprve ale něco o tom, co je vlastně nakresleno na obrázku 1. Matematici takovému obrázku říkají graf, je to skupina (množina) bodů, a skupina spojníc (hran) mezi některými z nich - tyto dvě množiny (uspořádaná dvojice množin) jednoznačně určuje graf.

Vrcholy grafu jsou nějak označeny - v našem případě jsou označeny jmény měst. Množina vrcholů grafu tedy vypadá takto:

{Cheb, Plzeň, Chomutov, Most, Praha}

Hrany grafu musíme také nějak popsat, nejjednodušší způsob je každou hranu popsat jmény vrcholů, které spojuje. V našem případě jsou tedy hrany grafu tyto:

{Cheb - Chomutov, Chomutov - Most, Chomutov - Praha, Plzeň - Praha, Most - Praha a Cheb - Plzeň}

Když se ještě podíváte na obrázek, zjistíte, že jednu důležitou skutečnost jsme ještě nijak nezaznamenali - každá hrana má přiřazeno nějaké číslo (vzdálenost, délku hrany). Takovému grafu se říká *ohodnocený graf*. Můžeme si pro každou hranu tuto délku poznamenat (ve stejném pořadí, jako máme zaznamenány hrany):

{80, 20, 110, 80, 80, 110};

Tyto tři množiny nám popisují náš obrázek jednoznačně. S jejich pomocí můžeme graf znovu kdykoliv nakreslit jednoduchým postupem:

1) nakreslíme si libovolným způsobem všechny vrcholy grafu (podle množiny vrcholů)

2) doplníme do grafu příslušné hrany (podle množiny hran), vůbec to nemusí být přímky a mohou se klidně protínat (jen musíme odlišit vrcholy od průsečíků)

3) nakreslené hrany doplníme o jejich ohodnocení (podle množiny ohodnocení, můžeme to provádět současně s kreslením hran podle bodu 2)

Možná vás napadlo, že sice nakreslíme obrázek, ale body nebudou na stejných místech jako na obrázku 1 - máte pravdu. Musíme si uvědomit, že tato informace pro nás není vůbec důležitá - graf je abstraktní vyjádření určitých informací - říká nám o vybraných místech to, která jsou propojena a jak dlouhá je spojnice mezi nimi.

V počítači si můžeme takový graf uložit různými způsoby, my si zvolíme matici vzdáleností - budeme současně ukládat informaci o délce hrany i o její existenci. Použijeme datovou strukturu pro uložení matice (dvojměrné pole):

	CHO	MO	PR	PL	CHB
CHO	0	20	110	0	80
MO	20	0	80	0	0
PR	110	80	0	80	0
PL	0	0	80	0	110
CHB	80	0	0	110	0

Jak vidíte, v tabulce je vždy na políčku [I,J] číslo nula, když mezi městy I a J neexistuje přímé spojení (nebo jsou totožná), jinak je tam vzdálenost mezi městy.

Odbočka (čtete až později)

Další zajímavost, které si můžete všimnout, je, že cesta z I do J a z J do I je stejně dlouhá (což také nemusí být vždy splněno na 100 %). Kdybychom chtěli šetřit, mohli bychom z naší tabulky uložit jen trojúhelníkovou část nad (nebo pod) hlavní diagonálou (to jsou políčka, kde I=J), v tabulce jsou označena barevně - jak se to udělá? Je to jednoduché - políčka si uložíme v jednorozměrném poli za sebou podle řádků a při vybírání a ukládání

přepočítáváme dva indexy na jeden, jak, na to zkuste přijít sami (napíšte si pro naši tabulku indexy vedle sebe - vždy obě souřadnice a vedle toho odpovídající novou souřadnici, pak zkuste přijít na nějaký vztah mezi nimi).

Konec odbočky

V další tabulce (poli řetězců) budeme mít uložena ještě jména měst - spojovacím prvkem mezi jmény a vzdálenostmi budou indexy: CHO=1, MO=2, PR=3, PL=4 a jako poslední CHB=5.

Algoritmus hledání

Pro hledání použijeme postup, který se nazývá Nejkratší cesta v grafu - algoritmus vypadá takto:

Inicializace: výchozímu vrcholu v grafu přiřadíme číslo 0, všem ostatním pak nějaké velké číslo, které bude reprezentovat nekonečno, přiřazenému číslu budeme říkat ohodnocení vrcholu.

Krok: vybereme vrchol, který má přiřazeno nejmenší číslo (s nejmenším ohodnocením) a není označen, označíme ho. Nyní projdeme všechny sousedící vrcholy (vrcholy, do nichž existuje hrana) a každému upravíme přiřazené číslo tak, že jej nahradíme součtem ohodnocení vybraného vrcholu a délky hrany do příslušného souseda v případě, že je menší než ohodnocení zpracovávaného souseda (to znamená, že bude-li například zvolena Praha a vybraný soused bude třeba Plzeň, změníme ji ohodnocení v případě, že bude větší než je součet délky hrany Praha-Plzeň a ohodnocení Prahy - viz příklad). Tento krok opakujeme tak dlouho, dokud zbyvají neoznačené vrcholy.

Raději si to ještě ukážeme na našem příkladu:

Inicializace - přiřadíme Praze 0 a všem ostatním městům (vrcholům grafu) E, což pro nás bude znamenat něco jako nekonečno, vzpomeňte si, jak je hlášeno přetečení u kalkulátorů - vytvoříme si tabulku a bude vypadat asi takto:

Chom.	Most	Praha	Plzeň	Cheb
E	E	0	E	E

1. krok - vybereme si vrchol s nejmenším ohodnocením, v tomto případě je to samozřejmě Praha, označíme ji a projdeme všechny sousedy, kterým případně změníme ohodnocení: Praha sousedí s Mostem, Most má ohodnocení E, délka hrany Praha-Most je 80, ohodnocení Prahy je 0, součet je tedy 80, což je méně než E, takže nové ohodnocení Mostu je 80. Podobně naložíme s Chomutovem (nově 110) a Plzeň (80), tabulka bude po prvním kroku vypadat následovně:

Chom.	Most	Praha	Plzeň	Cheb
110	80	0	80	E

Pokud jste se zamysleli, co jsme vlastně udělali, tak víte, že jsme zjistili vzdálenosti všech měst od Prahy, pro ta města, do kterých se lze dostat po jedné hraně.

2. krok - nyní si opět vybereme neoznačené město s nejmenším ohodnocením, nyní to bude Most (pokud budou dvě ohodnocení dvou měst stejná, vezmeme libovolné z nich, třeba to první v pořadí), označíme ho. Opět provedeme přehodnocení sousedních měst - Praha (staré ohodnocení 0, nové 160), ponecháme staré, Chomutov (staré 110, nové 100), nahradíme ho. K tabulce přidáme další řádek:

Chom.	Most	Praha	Plzeň	Cheb
E	E	0	E	E
110	80	0	80	E
100	80	0	80	E

Vidíte, že do Chomutova jsme našli z Prahy kratší cestu než minule. Také bychom si měli říci, proč vlastně vždy město s nejmenším ohodnocením označíme a už nemusíme brát v úvahu - je to proto, že cesty z označených měst do nejmenšího neoznačeného už máme spočítány a cesty z neoznačených už nikdy nemohou být lepší (součet délky cesty a ohodnocení výchozího města je vždy větší nebo roven současnému ohodnocení).

3. krok - vybereme další neoznačené minimum - je to Plzeň. Přepočítáme sousedy (ty označené už nemusíme brát v úvahu, jak jsme si řekli, do popisu algoritmu můžeme přidat, že zkoumáme jen neoznačené sousedy). Výsledkem bude další řádek tabulky:

Chom.	Most	Praha	Plzeň	Cheb
E	E	0	E	E
110	80	0	80	E
100	80	0	80	E
100	80	0	80	190

Všimněte si, že se blížíme k cíli, větší část měst už je označena a zbývají tedy jen dvě města - více než dva kroky už nás tedy nečekají. Známe už nejkratší cesty z Prahy do Prahy (0) do Mostu (80) a do Plzně (80). Cestu do Chomutova už známe vlastně taky, protože v příštím kroku to bude právě on, který bude označen.

4. krok - opět vybereme neoznačené minimum, je to Chomutov, označíme ho a provedeme přepočítání ohodnocení neoznačených sousedů. Jediný soused je Cheb a jeho nové ohodnocení bude 100+80 = 180, máme tedy kratší cestu než přes Plzeň. Ještě poslední řádek tabulky:

Chom.	Most	Praha	Plzeň	Cheb
E	E	0	E	E
110	80	0	80	E
100	80	0	80	E
100	80	0	80	190
100	80	0	80	180

Našli jsme nejkratší cestu z Prahy do Chebu, je dlouhá 180 km. Správně bychom

měli provést ještě jeden krok, ale je to zbytečné, k žádným změnám ohodnocení už nedojde.

Jak vidíte, našli jsme nejen nejkratší cestu z Prahy do Chebu, ale také nejkratší cestu z Prahy do všech ostatních měst.

Podmínky ukončení

Je vidět, že algoritmus musí skončit, v každém kroku totiž vlastně vyřazujeme jedno město, po počtu kroků rovnajícím se počtu měst tedy musíme vyřadit všechna.

My ale můžeme hledání cesty přerušit dřív, než se označí všechna města, víte kdy? No přece v okamžiku, kdy označíme cílové město, další kroky algoritmu už pro nás nic nového nepřinesou.

Programové řešení

Zkusíme si napsat jednoduchý program v BASICu, který by uvedený algoritmus prováděl:

```

10 REM Nejkratší cesta
99 REM Inicializace měst
100 RESTORE 1000
110 LET pocet=0
120 READ a$: IF a$ <> "" THEN LET
    pocet=pocet+1: GO TO 120
130 DIM m$(pocet,20)
140 RESTORE 1000
150 FOR i=1 TO pocet
160 READ m$(i)
170 NEXT i
199 REM Inicializace vzdáleností
200 RESTORE 2000
210 DIM v(pocet,pocet)
220 READ i: IF i=0 THEN GO TO 300
230 READ j,vzdal
240 LET v(i,j)=vzdal
250 LET v(j,i)=vzdal
260 GO TO 220
299 REM Vlastní algoritmus
300 CLS: PRINT "Zvol si počáteční stanici":
    GO SUB 3000
310 LET pocet=x
320 CLS: PRINT "Zvol si cílovou stanici":
    GO SUB 3000
330 LET clst=x
340 DIM o(pocet)
350 FOR i=1 TO pocet
360 LET o(i)=100000
370 NEXT i
380 LET o(pocet)=0
399 REM Nalezení minima
400 CLS: PRINT "Hledám:"
405 FOR i=1 TO pocet: PRINT AT 1,i*6-6;
    m$(i,1 TO 5): NEXT i
407 LET index=pocet: GO TO 500
410 LET min=100000: LET index=0
420 FOR i=1 TO pocet
430 IF o(i)>0 THEN IF min>o(i) THEN LET
    min=o(i): LET index=i
440 NEXT i
450 IF min=100000 THEN GO TO 600
499 REM Přehodnocení vrcholu
500 LET o(index)=o(index)
510 IF index=clst THEN GO TO 600
520 FOR i=1 TO pocet
530 IF v(i,index)=0 THEN GO TO 560
540 IF o(i)<0 THEN GO TO 560
550 IF v(i,index)+ABS o(index)<o(i) THEN
    LET o(i)=v(i,index)+ABS o(index)
560 NEXT i
565 FOR i=1 TO pocet
570 IF o(i)=100000 THEN PRINT
    TAB 6*i-6;"E": GO TO 590
580 PRINT TAB 6*i-6;o(i):
590 NEXT i: PRINT: GO TO 410

```

```

600 STOP
999 REM Jména měst
1000 DATA "Chomutov","Most","Praha",
    "Plzeň","Cheb"
1010 DATA ""
1999 REM Data vzdáleností
2000 DATA 1,2,20
2010 DATA 1,3,110
2020 DATA 1,5,80
2030 DATA 2,3,80
2040 DATA 3,4,80
2050 DATA 4,5,110
2060 DATA 0
2999 REM Zadáání stanice
3000 FOR i=1 TO pocet
3010 PRINT AT i+1,10;
    CHR$(1+CODE "A"-1);";m$(i)
3020 NEXT i
3030 PAUSE 0
3040 LET x=CODE INKEY$-CODE "a"+1
3050 RETURN

```

Jména měst jsou uložena v poli **m\$**, které se skládá z 20-ti znakových řetězců. Vzdálenosti jsou uloženy v matici **v**, ohodnocení vrcholů je uloženo v poli **o**.

Jména měst jsou uložena mezi řádky 1000 a 2000, poslední jméno musí být znak hvězdička - program si počet jmen spočítá sám. Vzdálenosti mezi městy jsou uloženy od řádku 2000 do řádku 3000, jsou tam vždycky dva indexy měst a následuje vzdálenost. Data vzdáleností musí být ukončena nulou.

Program pracuje tak, jak má. Obsahuje ale jeden "špinavý" trik:

Označování měst se provádí tak, že se jejich ohodnocení změní znaménko, tento způsob ale nepracoval u nuly (ta se musí brát jako označená), tam je to řešeno tak, že pro počáteční město (kde jediné je ohodnocení nulové) se vyhledávání vynechá. Z toho vyplývá, že na některých místech s bere absolutní hodnota z ohodnocení.

Kdybyste přepisovali program jinam, dejte si pozor na inicializaci pole vzdáleností, program využívá skutečnosti, že Spectrum při vytvoření pole dosadí do všech prvků hodnotu 0.

Program ale přeci jen nedělá úplně všechno - nalezne délku nejkratší cesty, ale už neukáže, kudy cesta vede. K tomu, aby to dělal stačí malá úprava - tu už ale ponechám na vás (návod: když budete ohodnocovat nebo měnit ohodnocení u vrcholu, zapisujte si, od kterého vrcholu jste přišli, po skončení vyhledávání pak jednoduchým cyklem projdete zpátky od cíle k počátku pomocí této informace a vypíšete vrcholy (města) na trase).

Praktická modifikace

V úvodním příkladu se ale nevyskytovala vzdálenost, nýbrž čas. Jak bychom tedy mohli hledat nejlepší spojení?

Místo vzdáleností použijeme při ohodnocování čas, výchozí město bude mít čas odjezdu (ten přijde ze vstupu), a v jednotlivých krocích budeme ke každému vrcholu přisuzovat čas, v němž je možné k němu dojet z vybraného vrcholu, nakonec dostaneme pro cílový vrchol výsledný čas.

Poněkud složitější bude zjištění časové náročnosti přejezdu mezi jednotlivými

stanicemi - musíte totiž počítat nejen dobu jízdy ale také čas, který je potřeba na čekání na první vhodný spoj - jste-li například v Praze v 6:00, první vlak jede v 7:30 a v Plzni je v 10:00, bude časová náročnost cesty 4 hodiny, i když vlak ve skutečnosti jede jenom 2,5 hodiny.

Takhle se například může stát, že nejlepší spojení nebude přes Most a Chomutov (i když je to nejkratší trať a tudíž by vlak měl jet nejkratší dobu) ale přes Plzeň, protože nebudete muset tolik času počekat na nádraží.

Závěr

No a to je k hledání nejkratší cesty v grafu všechno - kde všude by se dal tento algoritmus aplikovat, na to jistě přijdete sami.

Ale o jedné zajímavé aplikaci bych se přeci jen zmínil - znáte určitě hry (z PC nebo Amigy), kde stačí jenom ukázat šipkou kam, a figurka tam sama dojde, ba co víc, dokonce "inteligentně" obejde překážky, které jí stojí v cestě. I tady se dá použít hledání nejkratší cesty - plochu lze

rozdělit na šachovnici - políčka, na která lze vstoupit jsou vrcholy, políčka, kde jsou překážky, vrcholy nejsou. Spojnice jsou vždy mezi sousedními políčky (jestli jenom po stranách, nebo i na diagonálách, to záleží od toho, jak se může figurka pohybovat).

Na samotný závěr ještě malá poznámka: **prosím nepište mi o tom, že skutečné vzdálenosti mezi uváděnými městy jsou jiné, vím o tom!**

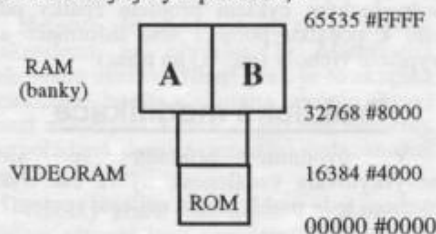
- Universum -

Softwarové využití druhé banky u Didaktiku GAMA

Neupravený počítač Didaktik Gama obsahuje proti klasickému ZX Spectru nebo Didaktiku M druhou paměťovou banku 32kB. V dnešní době tuto paměť využívá pouze několik programů vytvořených speciálně pro tento počítač. Mezi ně patří hlavně kopírovací programy (Gama Copy 80, Gama copy, SPD Copy), textové editory a viděl jsem i komfortní monitor a disassembler (popsaný ve FIFO č. 22). Mezi Gamisty koluje i mnoho her, které jsou upravené ze ZX 128 a ve 2. bance je uložena hudba nebo další dflly, které se později nemusí dohrávat z magnetofonu.

V tomto článku bych chtěl popsat několik způsobů, jak tuto druhou banku využít. Při tvorbě programů je potřeba si uvědomit dvě základní věci. První z nich je zásobník, který musí být umístěn tak, aby při přepínání banky nedošlo k poškození dat nebo ke zhroucení programu. Druhou věcí je to, že program v oblasti VIDEORAM (16384 až 32767) běží o něco pomaleji než v jiných částech paměti, takže zde nebude fungovat klasická neupravená load rutina ani různé programy, u kterých je třeba, aby byly co nejrychlejší.

Rozdělení paměti je na nákresu. K přepínání bank dochází pomocí příkazu **OUT (127),A**. Po tomto příkazu máte na adresách 32768 až 65535 zcela jinou paměť, takže si musíte dávat pozor, aby nedošlo ke zhroucení systému (pokud máte v druhé bance zásobník, nesmíte použít příkazy **RET**, **PUSH**, **CALL**, **POP** nebo mít povolené přerušení; nesmíte volat rutiny, které jsou v druhé bance, protože obsah obou bank je jiný a podobně).



První věc, kterou budeme potřebovat zjistit, je to, zda je počítač, na kterém je program spuštěn, je vůbec Didaktik s druhou bankou. To zajistí tento křafoučkový programček:

```
TEST  ld hl,65535 ;lib. adresa v oblasti bank
      xor a      ;A = 0
      out (127),a ;přepneme na banku A
```

```
ld (hl),255 ;zapišeme hodnotu
inc a       ;A = 1
out (127),a ;přepneme na banku B
ld (hl),0   ;do druhé banky nulu
xor a       ;A = 0
out (127),a ;zpátky na banku A
or (hl)     ;NZ znamená GAMU
           ;Z znamená něco jiného
```

Pokud bychom chtěli napsat nějaký monitor, který by sloužil k úpravám a prohlížení programů, budeme asi chtít, aby nezabíral mnoho místa a zároveň měl co nejvíce funkcí. Nejlepším řešením v tomto případě asi bude, když bude program uložen v bance B a do VIDEORAM umístíme křafoučkový rezident, který by mohl vypadat asi takto (oba programy se volají **call READ** nebo **call WRITE**):

```
READ  ex af,af ;adresa, odkud se bude
      ld a,126 ;číst je v registru HL
      jr WR2

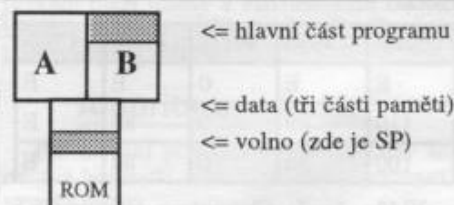
WRITE ex af,af ;adresa zápisu v HL
      ld a,119 ;nová hodnota v A
      ld ix,REZID ;REZID je jakákoli adresa
      ld (ix+3),a ;ve videoram, rezident
      xor a      ;se upravuje podle
      jp (ix)    ;funkce, spuštění
```

Nesmíte zapomenout, že podprogram **REZID** musí být umístěn v paměti VIDEORAM (mezi 16384 až 32767). Instrukce **nop** se podle potřeby nahrazuje instrukcí **ld (hl),a** nebo **ld a,(hl)**.

```
REZID out (127),a ;přepne na banku A
      ex af,af    ;původní hodnota A
      nop        ;modifikovaná část
      ex af,af    ;uschování registru A
      inc a       ;A = 1
      out (127),a ;přepne na banku B
      ex af,af    ;obnoví registr A
      ret        ;návrat na místo volání
```

Tento způsob nemusí vyhovovat tehdy, když chceme mít k dispozici daleko více paměti. Podle mě je velice pěkný způsob "přeořganizovat" si paměť pomocí malého programu. Paměť bude samozřejmě na svém místě, ale program adresu zkonvertuje, podle potřeby přepne banku a provede požadovanou funkci. Hlavní program může běžet v rychlé části paměti a nemusí se už o své fyzické uložení dat starat. Tento způsob je vhodný pro databanky, textové editory, slovníky a podobné programy.

Rozložení paměti je následující:



Tato část je uložena v bance A:

```
READ  ex af,af ;začátky obou programů
      ld a,126 ;vypadají podobně jako
      jr WR2   ;u předchozích programů
```

```
WRITE ex af,af
      ld a,119
      ld (KONV+3),a
KONV  ld a,h ;prvních 32 kB bude ve
      cp 128 ;VRAM a v bance A
      jr c,BANKA
BANKB ld a,1
      jp KONV2
```

```
BANKA ld de,32768-1024 ;pro banku A se
      add hl,de ;musí adresa konvertovat,
      xor a     ;pro hlavní prog. je 1 kB
      jp KONV2
```

Následující část musí být ve VideoRAM.

```
KONV2 out (127),a ;přepne banku
      ex af,af
      nop        ;upravovaná část
      ex af,af
      xor a
      out (127),a
      ex af,af
      ret
```

Pokud budete Vámi vytvořené programy odlaďovat, doporučuji vám použít některý z obrazkových nebo relokovatelných monitorů (DEVAST, DEVAST+). Monitor ale musí být celý umístěn na adresách 16384 až 32768!

Uvedené metody lze mírně urychlit, zkrátit a doporučit vám je přizpůsobit konkrétní situaci. Toto by mělo pro začátek stačit všem začínajícím programátorům, kteří by chtěli produkovat originální software pro Gamu nebo upravovat pro ni programy. Případné dotazy a připomínky mi zašlete na níže uvedenou adresu, osobní konzultaci je možno uskutečnit každý pátek od 15:00 v Havířově v KDLJ v učebně 45, kde sídlí Gama Klub.

- Martin Pustka -

(Zednická 6, Havířov, 736 01)

Nekonečné trápení s nekonečnými životy (2)

Zkuste vymazat počítač a napsat tento program:

```
5 LET A=1000
900 LET B=VAL "1000"
910 PRINT A,B
920 STOP
930 FOR J=23755 TO 6E4: PRINT J;" ";PEEK J; TAB 17;CHR$ PEEK J
AND PEEK J>32: NEXT J
```

Po spuštění vypíše počítač na první řádek obrazovky dvě čísla 1000. Nyní si vypíše paměť od adresy 23755 použitím cyklu na řádku 930. Měl by se vám objevit tento výpis (samozřejmě bez komentáře):

```
23755 0 .. číslo řádku = 0 x 256 + 5 = 5
23756 5 ..
23757 14 .. délka řádku = 14 + 0 x 256 = 14
23758 0 ..
23759 241 LET
23760 65 A
23761 61 =
23762 49 1 BASIC program v ASCII kódech, 23763 48 0 který se
vypisuje na obrazovku
23764 48 0 při listingu
23765 48 0
23766 14 .. bajt označující, že následujících pět
23767 0 .. bajtů tvoří číslo ve tvaru float. point
23768 0 ..
23769 232 .. A = 232 + 256 x 3 = 1000
23770 3 ..
23771 0 ..
23772 13 .. třináctka ukončuje řádek
23773 3 .. číslo řádku = 3 x 256 + 132 = 900
23774 132
23775 11 .. délka řádku = 11 + 256 x 0 = 11
23776 0 ..
23777 241 LET
23778 66 B
23779 61 =
23780 176 VAL
23781 34 " pokud pro zápis čísla použijeme
23782 49 1 VAL, počítač číslo nepřevádí do FP
23783 48 0 tvaru, čímž ušetříme tři bajty, ale
23784 48 0 program se tím zpomalí, protože 23785 48 0
počítač si musí číslo převádět
23786 34 "
23787 13 .. konec řádku
```

Nyní zkuste zadat POKE 23762;CODE "0" a vylistujte si program. Na řádku 5 se objeví příkaz LET A=0000, ale pokud program spustíte a vypíšete si proměnnou A, její hodnota bude stále 1000. To znamená, že počítač počítá místo se znakovým vyjádřením s FP, který jste nijak neměnili. Ale pozor, pokud byste řádek vyeditovali, počítač si samozřejmě FP vyjádření změnil podle znakového tvaru. Z toho vyplývají dvě věci. Za prvé, že si nikdy při rozkódování takového řádku nesmíte tento řádek vyeditovat (jinak se změní hodnota FP vyjádření na hodnotu, kterou vidíte), a za druhé, že i programátor, který takto kódovaný řádek vytvářel, si musel nejprve řádek napsat a potom pomocí například POKE změnit znakové vyjádření tak, jak jsme to zkusili i my.

Jistě se asi ptáte, proč je v programu řádek 900 (mimo toho, aby vám ukázal, jak se zapíše do paměti řádek 900). Na tomto řádku se do proměnné B ukládá také číslo 1000, ovšem pomocí příkazu VAL. Tento příkaz způsobí, že si počítač nevytváří FP vyjádření. To znamená, že pokud bychom použili například POKE 23782;CODE "0", spustili program a nechali si vypsát proměnnou B, bude obsahovat číslo 0. V tomto případě tedy nelze použít trik RANDOMIZE USR VAL "0", protože počítač si číslo do FP převádí až při běhu programu. Běh programu je potom sice pomalejší, ale číslo zabírá v paměti méně místa, což se někdy velice hodí. Velice často se také používá: místo nuly - NOT PI (ušetříte pět bajtů), místo jedničky - PI/PI (ušetříte čtyři bajty) atd.

Máme-li tedy program, kde je zřejmě použita tato finta, necháme si vypsát paměť BASICu (například pomocí cyklu na řádku 930 v našem programu) a najdeme řádek, který chceme rozkódovat. Za číslem ve znacích bude číslo 14 a za ním, jak již víme, následuje pět bajtů FP. Pokud je první, druhý a poslední bajt nulový, jednoduše hodnotu čísla vypočítáme, protože víme, že v tomto případě je třetí bajt bajt nižší a čtvrtý vyšší, a výsledné číslo se vždy vypočítá $256 \times \text{VYŠŠÍ BAJT} + \text{NÍŽŠÍ BAJT}$. Pokud však bude obsazená i pozice první, druhá nebo poslední, nezbyvá nám, než přijít na hodnotu čísla jinak, protože vypočítat ho, by bylo dosti náročné. Obsazení některé z dalších pozic je způsobeno tím, že programátor vám chtěl ztížit rozkódování a jako startovní adresu nepoužil číslo celé, ale desetinné. Počítači to nevadí, protože ten si číslo vždycky zaokrouhlí, ale FP se tím notně změní. Číslo pak zjistíme například tak, že si opíšeme všechny FP pozice tohoto čísla, vymažeme si počítač a napíšeme například řádek:

```
10 PRINT I
```

Vylistujeme si paměť BASICu a najdeme tento řádek (nebude to nic těžkého, protože pokud jste vymazali počítač, bude tento řádek hned první, čili bude začínat na adrese 23755). FP tvar jedničky na tomto řádku přepíšeme pomocí POKE hodnotami, které jsme si zapsali, a spustíme program. Počítač nám vypíše číslo, jenž po zaokrouhlení dává hledané číslo (počítač zaokrouhluje čísla tradičním způsobem, tzn., že například 23296.491 zaokrouhlí na 23296 a 23296.502 zaokrouhlí na 23297).

Upozornění:

Může se stát, že programátor, který vytvářel program, vás chce zmást a číslo nezmění na nulu, ale například na 50000. Vy si potom myslíte, že hra je spouštěná od adresy 50000, nefunguje vám to a jste v koncích, protože nevíte, jak dál. Neztrácejte proto v tomto případě hlavu a podívejte se, jestli znakový tvar odpovídá tvaru FP.

1.2.4.4. Nultý řádek

Řádek s číslem nula se v programech vyskytuje nadmíru často. Jeho velická výhoda je v tom, že ho nelze vyeditovat a nelze ho ani smazat. BASICový interpret nepočítá s tím, že by mohl takovýto řádek existovat (a oprávněně, protože pokud chcete řádek s číslem nula odeslat, počítač vypíše hlášení Nonsense in BASIC). V případě, že chceme takovýto řádek vytvořit, musíme změnit číslo řádku pomocí POKE (na které pozici se číslo řádku nachází, již víme). Dejme tomu, že máme program a chceme, aby první řádek tohoto programu měl číslo nula. Protože první řádek je v paměti uložen od adresy 23755, stačí použít POKE 23755,0:POKE 23756,0. Pokud bychom chtěli změnit číslo nějakého dalšího řádku, najdeme si jeho začátek v paměti a pomocí POKE číslo změníme. V případě, že budeme chtít řádek vyeditovat, budeme postupovat stejným způsobem, ovšem místo řádku nula vytvoříme například řádek jedna.

Možná vás napadá, co se asi stane, když změníte číslo řádku na 10000 a výše. Protože největší číslo řádku ve spectrovském BASICu může být 9999, počítač řádek vůbec nevypíše. V jeho paměti, pokud ho nepřemažete, je a jestliže zase POKEm vrátíte číslo řádku na nějaké normální, počítač řádek znovu vypíše. Tento trik můžeme ve hrách také často najít.

Příklad:

Vymažte počítač a opište tento řádek:

```
10 CLS:PRINT "PROGRAM BĚŽÍ":LOAD "
```

Nyní napište POKE 23755,100 a zkuste řádek vylistovat. Není tam, protože příkaz POKE změnil vyšší bajt čísla řádku a číslo řádku je teď $100 \times 256 + 10 = 25610$, což je více než 9999 a počítač takový řádek nevypíše. V případě, že chcete řádek spustit, musíte

ho spouštět přesně od jeho čísla, tj. v našem případě 25610. Pokud si tento program uložíte na kazetu třeba příkazem SAVE "ZKOUSKA" a potom ho zkusíte nahrát zpátky příkazem MERGE, dojde ke zhroucení systému (jeden z triků, kterým se kódují programy proti nahrávání příkazem MERGE). V případě, že chcete takovýto řádek vrátit zpět, stačí, když uložíte na adresu 23755 zpátky nulu, a řádek se opět objeví (pokud jste ho již ovšem nepřepsali jiným řádkem). Jak rozkódovat takto upravený program je více než jasné. Stačí si vylistovat paměť BASICu, najít řádek s číslem větším než 9999 a uložit místo tohoto čísla číslo ze správného intervalu.

1.2.4.5. Změny syst. proměnných

Protože v paměti ROM nelze měnit obsahy jejích adres, má interpret vyhrazenou část paměti RAM od adresy 23552 do 23750 pro systémové proměnné (viz. slovníček). Změnami hodnot těchto proměnných můžeme získat různé efekty, kterých jiným způsobem nelze dosáhnout. Seznam těchto proměnných včetně jejich ovlivnění se nachází v příloze A.

II. ČÁST

Zadání Poke do loaderu ve strojovém kódu

Tato část již vyžaduje více než dobrou znalost BASICu a ucházející znalost assembleru. Nejprve se popíšeme způsoby uložení loaderu ve strojovém kódu, s tím, kam lze poke uložit a jak, a nakonec se pomocí názorných příkladů seznámíme s nejčastějšími typy loaderů a kódováním.

Poznámka:

Všechny zdrojové texty mají adresy a hodnoty uvedeny v hexadecimálním tvaru, pokud se hexadecimální číslo nachází v textu, je za ním písmeno h.

2.1. Ukládání a spouštění loaderů z BASICu

Pokud chceme strojový program rozkódovat a vložit do něho pouk, musíme zjistit, na které adrese v paměti se nachází a od které se spouští. V hrách můžeme narazit na tři následující a nebo jim velice podobné metody:

Program ve strojovém kódu je uložen na některém z řádků BASICu, většinou za příkazem REM (není podmínkou) a spouští se buď přímo od adresy, na které se nachází (například RANDOMIZE USR 23760), nebo se hodně často využívá hodnoty v systémové proměnné PROG (například RANDOMIZE USR (PEEK 23635 + 256 * PEEK 23636) + 5).

Nežádá bývá strojový program uložen v oblasti proměnných tj. na adrese, kterou udává systémová proměnná VARS. Program se pak obvykle spouští příkazem RANDOMIZE USR (PEEK 23627+256*PEEK 23628).

Poslední možnost je uložení programu do samostatného bloku CODE, který se nahraje do paměti a potom spustí od příslušné adresy.

2.2. Vkládání POKE

Ještě než si popíšeme některé typy loaderů, vysvětlíme si, jakým způsobem je v assembleru nejlepší vkládat pouk (slovem pouk je zde myšlen sled instrukcí assembleru, který uloží na příslušné adresy příslušné hodnoty, čili v podstatě vykoná příkaz POKE ve strojovém kódu).

Jako v BASICu je nejlepší vložit pouk před instrukci spouštějící hru. To by však znamenalo, vytvořit si před ní prostor, který většinou zabírá důležitý program. Proto se používají metody, které operují pouze se skokovou instrukcí, plní stejnou funkci a daný program nijak nepoškodí.

Pokud se instrukce spouštějící hru (například JP 32768) nachází na posledním místě programu a za ní je místo, stačí ji přepsat

instrukcemi zajišťujícími pouk a na konec opět zařadit skok na startovní adresu (v našem případě JP 32768).

Jestliže se za příkazem spouštějícím hru ještě nachází nějaký důležitý program, můžeme to provést tak, že si vytvoříme vlastní program, který vloží dané pouky a potom spustí hru. Tento prográmk se uložíme někam do volné paměti (například do buferu tiskárny nebo do obrazovky) a příkaz, spouštějící hru, upravíme tak, aby skákal na jeho adresu.

2.3. Nejčastější příklady loaderů ve strojovém kódu

V této kapitole se naučíme porozumět tomu či onomu programu a případně ho rozkódovat do takového tvaru, aby bylo možné do něho vložit pouk. Pomocí příkladů si vysvětlíme některé typické loadery ve strojovém kódu, popíšeme si, jak fungují a kam lze vložit pouk. Příklady jsou vybrány tak, aby zahrnuly co největší počet metod, i když zde samozřejmě nějaký typ loaderu nebo kódování chybí. Po pochopení uvedených příkladů by vám neměly dělat velké problémy ani programy, se kterými se setkáte poprvé.

Je zřejmé, že všechny loadery musí mít část, která nahrává vlastní hru, a část, která ji spouští. Pokud má program pouze tyto části, nebývá velkým problémem do něho vložit pouk, ovšem to se stává málokdy, a proto je dobré mít již nějaké vědomosti a zkušenosti s assemblerem, protože čím budete mít větší znalosti, tím víc programů se vám podaří rozkódovat. Také je nutné vlastnit nějaký monitor (viz. slovníček) a to nejlépe takový, který se ukládá do obrazovky (není však podmínkou).

2.3.1. Loadery využívající nahrávací podprogram v ROM

Nejjednodušší loadery, se kterými se u her můžeme setkat, jsou ty, které využívají nahrávací podprogram z ROM (viz. příloha B). Tento podprogram se nachází na adrese 0556h, jmenuje se LD-BYTES a před jeho vyvoláním musí být v příslušných registrech uloženy tyto hodnoty:

registr IX první bajt nahrávaného bloku dat, tj. adresa, od které se bude blok dat nahrávat

registr DE délka (počet bajtů) bloku dat

registr A leader (úvodní, typový bajt)

indikátor CY pokud je nastaven na 1 - rutina nahrává, pokud je nastaven na 0 - rutina verifikuje

Někdy také loadery využívají podprogram na adrese 0802h, který skočí na LD-BYTES a po návratu zjišťuje, jestli nedošlo k chybě při nahrávání. Pokud ano, vypíše chybové hlášení:

```
0802 CALL 0556 Skok na LD-BYTES a
RET C návrat, pokud nedošlo k chybě
0806 RST 8 Výpis chybového hlášení číslo 1Ah
DEFB 1A R - Tape loading error.
```

Příklad:

Po rozkódování BASICu hry FIST II. jsme zjistili, že loader ve strojovém kódu se nachází na řádce 20 za příkazem REM a spouští se od adresy 5CE8h. Disassemblovaný výpis od této adresy vypadal takto (pozor na číselné soustavy - výpis je v šestnáctkové):

5CE8 DI Zákaz přerušení.

```
LD HL,4000 Toto je velmi často používaný
LD DE,4001 způsob mazání obrazovky.
LD (HL),0 Kratší by bylo LD (HL),L.
LD BC,1AFF
LDIR
```

```
LD HL,5D04 Přenos loaderu na 65000.
LD DE,FDE8 při spuštění na této adrese, by
LD BC,17 byl přehrán kódem hry, a proto
LDIR bude přenesen jina
```

JP FDE8 Spuštění loaderu.

```
FDE8 LD IX,4000 Do IX adresa pro nahrání
LD HL,FD62 Výpočet délky
```

LD DE,4000
OR A,CY je nastaveno na 0 (kvůli odečtení)
SBC HL,DE odečtena.
EX DE,HL Přenos délky do DE.

SCF CY je nastaveno na 1 - nahrávání,
LD A,FF typový bajt (leader) je 255 a
CALL 0556 skok na nahrání hry.

JP F10E Tato instrukce spouští hru.

Jestliže bychom chtěli do tohoto loaderu vložit například POKE 56002,23, stačilo by přepsat skok JP F10E třemi následujícími instrukcemi:

LD HL,DAC2
LD (HL),17
JP F10E

Je podivné, že program nemění hodnotu zásobníku (SP), ačkoli přehrává téměř celou paměť. Znamená to, že SP musí být nastaveno již z BASICu a to na hodnotu větší než 64872.

Abyste se naučili orientovat v takovýchto programech, připojují ještě dva loadery tohoto typu. Doporučuji, abyste nejprve sami zkusili pochopit, jak program pracuje, a potom se přesvědčili, zda-li jste měli pravdu. Můžete také vymyslet, kam lze vložit pouky.

První je trochu zjednodušený loader ze hry Target Renegade (v systémové proměnné PROG je adresa 5CCBh = 23755).

5CD0 LD HL,(5C53) HL=5CCBh (hodnota v systémové proměnné PROG).
LD D,H DE=HL.
LD E,L
LD BC,F7 Pomocí BC a adresy z PROG se
ADD HL,BC dopočítá určitá adresa v paměti.
LD C,(HL) Na dopočítané adrese (5DC2h)
INC HL se nachází hodnota,
LD B,(HL) která je přenesena
LD (5C80),BC na adresu 5C80h.
LD A,C3 Před adresu 5C80h je umístěn kód pro instrukci
LD (5C7F),A JP a na adrese 5C7Fh nám tak vznikne JP 0556h.
LD HL,0556 Kontrola, jestli je
AND A číslo v BC
SBC HL,BC opravdu 0556h.
JR Z,5D01 Pokud ano, odskok na nahrávání.
RST 8 Pokud ne,
DEFB 0C vypíše se chyba.

5D01 LD IX,61D8 V případě, že adresa v BC byla správná,
LD DE,9272 nahraje se první část hry.
LD A,FF
SCF
CALL 5C7F Na adrese 5C7Fh je JP 0556h.
CALL 61FE Skok na podprogram, který detekuje, zda-li
LD A,(61F9) není hra nahrávána na počítači
BIT 2,A ZX Spectrum 128.
JR NZ,5D35 Pokud ano, mohou se do paměti dohrát i další části.
JP 6201 V opačném případě je hra spuštěna.

5D35 LD B,4 Zde začíná loader pro ZX Spectrum 128.

.
.
.
JP 6201

Pouky bychom v tomto případě vložili (včetně skoku na adresu 6201 na konci) do volné paměti a upravili bychom startovní instrukci tak, aby skákala na náš program.

Druhý příklad je trochu rozsáhlejší. Patří ke hře, která má dvě části. Nejprve se nahraje a spustí první část, kde si zvolíte nějaké důležité údaje a poté se hra vrátí zpět do loaderu a nahraje se hlavní část hry.

Abychom správně pochopili, jak program funguje, je potřebné vědět, že při použití příkazu USR v BASICu se adresou, na kterou se skáče, naplňuje registr BC.

5D08 DI Zákaz přerušení.

LD HL,AF HL=Afh, BC=adresa spuštění (5D08h).
ADD HL,BC HL=Afh+5D08h=5DB7h=adresa textu.
EX DE,HL Výměna obsahu párových registrů DE a HL.
LD HL,9E
ADD HL,BC HL=9Eh+5D08h=5DA6h.
LD (HL),E Instrukce LD HL,Afh na adrese 5DA5h
INC HL se změní
LD (HL),D na LD HL,5DB7h.
LD HL,7B
ADD HL,BC HL=7Bh+5D08h=5D83h=začátek prg.na výpis chyby.
LD SP,5FE6 Zásobník=5FE6h.
PUSH HL Uložení adresy 5DB7h a přesun adresy zásobníku
LD (5C3D),SP do syst.pr. ERR SP způsobí, že při chybě
LD HL,4000 počítač skočí na adresu 5DB7 (adresa, na
LD DE,4001 kterou ukazuje ERR SP).
LD BC,1AFF
LD (HL),L
LDIR Vymaz obrazovky.
XOR A
OUT (FE),A Černý okraj.

LD IX,4000 Nyní se nahraje obrázek.
LD DE,1B00
LD A,FF
SCF
CALL 0556
LD IX,6000
LD DE,96F8
LD A,FF
SCF
CALL 0556 Nahrání první části.

LD BC,7200 První část měla být zřejmě původně spuštěna z
CALL 7200 BASICu.

LD IX,7200
LD DE,8E00
LD A,FF
SCF
CALL 0556 Nahrání hlavní části.
JP 8F00 Spuštění hlavní části.

5D83 LD HL,4000
LD DE,4001
LD (HL),L
LD BC,1800
LDIR Vymazání obrazovky (bez atribut).
LD HL,5800
LD DE,5801
LD (HL),0F 0Fh=1111 binárně - modrý papír, bílý inkoust.
LD A,1
OUT (FE),A Modrý okraj.
LD BC,02FF
LDIR Změna atribut.
LD A,2 Otevření druhého kanálu
CALL 1601 pro výpis na hlavní část obrazovky.

5DA5 LD HL,AF Zde je již LD HL,5DB7h, což je adresa textu.
LD IY,5C3A Nastavení správné hodnoty do IY.
5DAC LD A,(HL) Vezme se jeden znak textu do A a pokud to
CP FF není bajt s hodnotou FFh,
JR Z,5D83 tak se pokračuje ve výpisu textu.
PUSH HL Úschova ukazatele textu.
RST 10h Tisk znaku v registru A na obrazovku.
POP HL Obnovení původní hodnoty v HL.
INC HL Nastavení ukazatele na další znak a
JR 5DAC uzavření smyčky.

5DB7 DEFB "LOADING ERROR"
DEFB FF

Na konec této kapitoly bych vám chtěl ukázat, že loadery vůbec nemusí mít (a většinou nemají) strukturu jako v uvedených příkladech. Často se používají jednoduché triky (někdy ty nejtěžší) a člověk si musí dávat dobrý pozor, aby se nezamotal.

- Martin Dvořák -

(Pokračování příště)

STROJOVÝ KÓD Z-80

(trochu jinak, 7. díl)

Příklad přelož a pak v monsu na adresu #C355, kde je instrukce #C9 (RET) vlož bod zastavení (break point) stiskem klávesy ' W' . Nastav ukazatel a PC na #C350 a klávesou ' +' spusť rutinu až do bodu zastavení.

Přečti obsah STKEND (#5C65) a podívej se, že na udané adrese leží pětibytové vyjádření čísla, uloženého v A_registru řádkem 30.

```
1. 2. 3. 4. 5. BYTE
... 00 00 41 00 00 00 ... Pozn.: #41 = 65
  ^ ^
```

(STKBOT) (STKEND)

5.2 PRINT_FP CALL #2DE3

Podprogram vypíše na obrazovku číslo uložené na vrcholu ZK. Nesmí se používat, je-li ZK prázdný! Předchozí příklad rozšíříme o řádek 50:

```
50 CALL #2DE3 ;vypíše hodnotu 65, protože tato
;byla naposledy uložena na vrchol ZK
```

Upravený příklad znovu přeložíme, vrátíme se do basicu a spustíme příkazem: **RANDOMIZE USR 50000: PAUSE 0** Protože jsme neotevřeli kanál "S" je číslo vypsáno do editační zóny.

5.3 STACK_BC CALL #2D2B

Přeneseme hodnotu z BC_registru na vrchol ZK (obdoba STACK_A). Edituj řádky předchozího programu tak, abys sám vyzkoušel tuto rutinu!

5.4 SCANNING CALL #24FB

Podprogram vyhodnotí výraz a výsledek uloží na vrchol ZK. Adresa výrazu je uložena v systémové proměnné CH-ADD (umístěna na #5C5D).

Pro numerický výsledek uloží skutečné číslo typu FP, pro řetězec uloží na vrchol ZK 5 bytů takto: 1. byte není specifikován, 2.+3. byte obsahuje adresu začátku řetězce, 4.+5. byte je jeho délkaBit 6 systémové proměnné FLAGS (23611, #5C3B, IY+1) je roven 1 pro číselné hodnoty výsledku nebo 0 pro řetězec.

5.5 FP_TO_A CALL #2DD5

Přeneseme hodnotu z vrcholu zásobníku kalkulátoru do registru A.

5.6 STK_TO_A CALL #2314

Totéž co předchozí FP_TP_A, navíc je v C_reg #01, je-li hodnota v A_reg. kladná, #FF záporná

5.7 FP_TO_BC CALL #2DA2

Přeneseme hodnotu z vrcholu ZK do BC_registru a současně nižší byt do registru A.

5.8. STK_TO_BC CALL #2307

Totéž jako předchozí FP_TO_BC, navíc do DE_reg. uloží znaménko výsledku (#01 nebo #FF jako u STK_TO_A).

5.9 PŘÍKLADY POUŽITÍ

Před využitím kalkulátoru (instrukce RST #28) musíme nejprve naplnit ZK, a to tak, jak si to každá jeho operace vyžaduje. Rutiny

uvedené v předchozích kapitolách 5.1 až 5.8 slouží jak pro naplnění ZK před použitím kalkulátoru, tak pro převzetí výsledku z vrcholu ZK do některých z registrů procesoru.

Připomeňme si ještě, že po instrukci RST #28 následuje skupina bytů. Každý z těchto bytů definuje jednu operaci kalkulátoru - viz příloha. Posledním bytem je #38, který označuje konec práce kalkulátoru. Další byty jsou tedy strojovým kódem.

Pomocí jednoduchých příkladů se naučíme kalkulátor používat.

5.9.1 SEČÍTÁNÍ DEFB #0F

333 + 444 = 777 Bude to pravda?

```
10 ORG 50000
20 START
30 CALL #0D6B ;CLS
40 LD A,2
50 CALL #1601 ;KANÁL 'S'
60 LD BC,333 ;PRVNÍ ČÍSLO DO BC_REG
70 CALL #2D2B ;A NA VRCHOL ZK (333)
80 LD BC,444 ;DRUHÉ ČÍSLO DO BC_REG
90 CALL #2D2B ;A NA VRCHOL ZK (333 444)
100 RST #28 ;SPUSTÍ KALKULÁTOR
110 DEFB #0F ;SEČTE HODNOTY 333 a 444 A
; VÝSLEDEK ULOŽÍ NA VRCHOL
120 DEFB #31 ;DUP - ZDVOJÍ HODNOTU NA
;VRCHOLU ZÁSObNÍKU
130 DEFB #38 ;KONEC PRÁCE KALKULÁTORU
140 CALL #2DE3 ;VYPÍŠE VRCHOL ZK NA OBR.
150 CALL #2DA2 ;ZE ZK DO BC_REG.
160 RET
```

Program přeložte a spusťte z basicu. Při jeho správné funkci je výsledek výpočtu vypsán do levého horního okraje obrazovky.

5.9.2 NÁSOBENÍ DEFB #04

333 x 444 = 147852 Bude to pravda?

Protože jsme tvorové líní, upravíme předchozí program. Edituj řádek:

```
110 DEFB #04 ;PRO NÁSOBENÍ
```

Všimněte si, že výsledek je v obou příkladech v závěru výpočtu na vrcholu zásobníku kalkulátoru zdvojen, t.j. uložen dvakrát za sebou. To proto, že poprvé vybereme jednu hodnotu výsledku pro naplnění BC_registru, podruhé je výsledek z vrcholu ZK odebrán rutinou pro výpis na obrazovku.

5.9.3 ODEČÍTÁNÍ DEFB #03

Odečte hodnotu uloženou pod vrcholem ZK od hodnoty, uložené na vrcholu ZK:

333 - 444 = -111 Bude to pravda?

Tentokrát edituj jen řádek 110:

```
110 DEFB #03 ;PRO ODEČÍTÁNÍ
```

- pokračování příště -



ERNIE

NEBEZPEČNÝ
ŠÍLENEC

"A 15-TI MILIÓŇ BITOVÝ ZÁZRAK"

by Phil

JEDNOU ODPOLEDNE V BIFETU...



NAZDAR ERNIE!

NAZDAR SIDE!
CO TO MÁŠ?

ŽEROVNA JSEM SI KOUPIL TUHLE NOVOU 45-TI
MILIÓŇBITOVOU SUPER ULTRA TV-HRU



FAKT? A CO JE NA NÍ TAK
ULTRA SUPER DOBRÝHO?

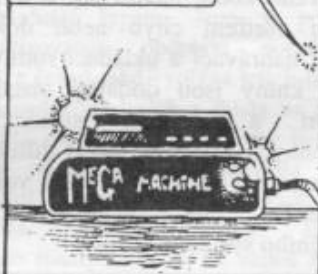


JEDINÉ TATO TV-HRA JE ÚPLNĚ DOKONALÁ!
NEJMODERNEJŠÍ VÝROBEK NAŠEHO VIDEOPRŮMYSLU!



SAMOLIBÉ STUPIDNÍ ŘEČI,
KTERÉ MÁ KAŽDÝ, KDO SI
KOUPÍ LEPŠÍ POČÍTAČ, NEŽ
MÁTE VY. NERUKEJTE, ŽE
BYSTE MU HO NEJRADEŠI
NEROZMLÁTILI
KLADIVEM A
JEŠTĚ SE U TOHO
SLADCE NEUSNAŽUJTE!

MÁ NEJLEPŠÍ GRAFIKU A SAMO-
ŘEJMĚ SAMPLOVANÝ ZVUK... V CENĚ
900 LIBER JE ZAHNUTO I NĚCO PŘES
2 BILIONY SUPER-HER...



KDYŽ ZMAČKNU TOHLE TLAČÍTKO, FUNKUJE TO TAKY
JAKO OTVÍRAK NA KONZERVY, PLYNOVÝ KAMNÁ, NŮŽKY,
KRABÍČKA POSLEDNÍ ZACHRANY, NOBELOVA CENA ZA MÍR
A JEŠTĚ AŽ 200 DALŠÍCH...



HMM... TO JE VŠEČKO MOC
HERKÝ, ALE CO TVOJE STARÝ
SPECTRUM?

SPECTRUM? ZBLÁŽNIL
SEŠ? TO JE ŠROT!!!



NECÍTÍM SE NĚJAK
DOBŘE, JINDRO

JŠŤ MOC RYCHLE,
TÍM JE TO!

NA SPECTRU UŽ NIKDO NEHRAJE!
SPECTRUM JE HISTORIE! PŘÍŠEL ČAS
ZMĚN!!!



NIKDY MĚ NEPOSLE-
CHNEŠ A POŘAD SE JENOM
PŘEJDAŠ
A PAK JE
TI BLBĚ...

TYHLE TELEVIZNÍ HRY BŘED DVLÁDNOUT
HERNÍ PRŮMYSL A PAK UŽ VŮBEC NIKDO
NEBUDE ČÍST ČASÁKY, JAKO JE TŘEBA TENHLE!



PROMIŇ, ALE JÁ DÁM RADŠÍ
PŘEDNOST SVÝMU SPECTRU

PROČ? MOJE TV-HRA
DOVÁŽE ÚPLNĚ VŠECHNO
JE NEJLEPŠÍ!



PARDON, NEVIDĚL JSTE TADY
MOJEDHO MALÝHO? JE AŽ TAK HLAVNĚ VELIKEL

SKVĚLÝ... ALE JE TU
JEDEN TAKOVÝ PROBLÉM...

JAKEJ PROBLÉM?

TO SE MI STANE VĚDŮCKY!



NEFUNKUJE TO!!!

BADBOOM!!

JÁ - MYSLÍM - ASI - SE -
VRÁTÍM - KE - SPECTRU...

HODNEJ KLUM



SW novinky

George K.'s SQ-Tracker

Na přání dvou vynikajících hudebníků **SCALEXE** a **QJETY** vznikl nový editor na tvorbu AY-hudby (Melodik, Best,...). Až doposud najuznávanější hudební program pro zvukový obvod AY Bzykův "Sound Tracker" (nemá nic společného se slovenským ST) začal pomalu zastarávat, a tak přišel čas pro něco nového - a pokud možno i lepšího...

Ano, přišel čas pro **SQ-Tracker**!

SQ-Tracker se skládá ze tří programů: **SQ-Editor** (v něm hudbu skládáte), **SQ-Compiler** (pro data z SQ-Editoru vytvoří rutinu, která hraje hudbu pod přerušením - můžete ji umístit do svých programů, her a dem) a **SQ-Linker** (dokáže pospojovat zkompilevané hudby, zároveň dochází podstatnému zkrácení dat - opakující se části jsou vypuštěny).

Hudebníkům, kteří dosud používají starý Sound Tracker, přinášíme stručný výčet hudebních možností nového **SQ-Trackeru**:

- větší počet samplů (nástrojů), ornamentů (akordů) i patternů (motivů)
- glissanda (klouzavé tóny) bez složitých definic vibračních obálek
- u ornamentů vlastní dozívání
- hardwarové obálky jsou kombinovatelné s definovanými
- patterny pro každý kanál zvlášť (ve skladbě můžete použít i patterny různých délek!)
- každou pozici lze přehrávat vlastní rychlostí, každý kanál na pozici jde zvlášť transponovat a určit jeho hlasitost
- po dohrání se skladba opakuje od libovolné pozice
- během hraní patternu můžete měnit hlasitost (každý kanál zvlášť) i rychlost

SQ-Tracker vám nabízí množství editačních funkcí (velice jednoduché kopírování patternů, samplů nebo ornamentů, transpozice, přehrávání skladby během editace, atd.), spolupráci

se všemi basicem ovládanými diskovými systémy; na ZX 128 je dostupný editor i kompilér zároveň.

Rutiny ROM D40/D80

Chcete, aby vaše programy spolupracovaly s disketovou jednotkou D40 nebo D80 (Kompakt)? Chcete si převést staré programy nebo dílové hry z kazet na diskety? Sháníte informace o operačním systému, který je ukryt v ROM vaší disketové jednotky?

Pokud se vás týká některý z výše uvedených problémů, pak neváhejte a objednejte si knihu **Rutiny ROM D40**, ve které se dozvíte vše potřebné.

V knize najdete kapitoly o základních příkazech MDOSu, popis, jak využívat služeb MDOSu ze strojového kódu, návod jak si vytvořit vlastní ošetření chyb nebo dokonce vlastní nahrávací a ukládací rutiny. Na konci knihy jsou dodatky obsahující seznam a význam systémových proměnných MDOSu, výčet důležitých podprogramů MDOSu a jejich volání a upozornění na některé chyby operačního systému.

Source D40/D80

Zdrojové texty ke knize Rutiny ROM D40/D80 ve formátu assembleru PROMETHEUS. Ušetříte si práci s přepisováním a následným hledáním takto vzniklých chyb.

Zdrojové texty dodáváme pouze na disketách.

Public 7

Public 7 se od předchozích souborů liší jedním detailem. Všechny programy jsou od jednoho autora - AXELSOFT.

Mr. CAT - program pro archivaci adresářů disket (s jeho pomocí si můžete na disketách udělat pořádek).

Hádej - hra - hádání slov a vět po písmenech - obdoba známé hry Šibenice.

Hra v 15 - známá 15-ka (přesunování čtverců s čísly).

Selector - program na výběr výřezů z obrázků.

Erotický test a **Test mládí** - otestujte se...



(c) 1993 George K.

ZX Magazín a robotika (7)

R.U.R.

(aneb nechte práci strojům)

Po několika dílech, věnovaných interfacingu, se konečně náš seriál přehoupnul do své druhé části, ve které si budeme konkrétně povídat o některých aplikacích dříve popsaných modulů. Podtitulek seriálu Vám již od začátku slibuje pohyblivou hračku řízenou počítačem. Myšlenka jistě přitažlivá, nicméně náročná. Pro ty, kteří se rozhodnou s námi si postavit robota, řízeného mikropočítačem, popíšeme nyní základní principy a rady, kterých je dobré se při takové konstrukci držet, aby jim robot dobře fungoval.

Takže je tu opět **zpětná vazba**. Tento pojem jsme si již objasnili ve čtvrtém pokračování našeho seriálu. Pokud si ještě vzpomenete, psali jsme o tom, že systém řízení čehokoli bez jeho zpětné kontroly, tzv. **řízení v otevřené smyčce**, může být při složitější aplikaci nespolehlivý, nebo alespoň značně neefektivní. Snažte se tedy ve svých zapojeních princip zpětné vazby (tj. **zapojení v uzavřené smyčce**) raději respektovat a ukončení, resp. průběžný stav každé činnosti potvrdit zpět řídicímu počítači. Můžete si tak ušetřit spoustu starostí s přesným taktováním jednotlivých, mnohdy na sebe navazujících, kroků. Jsou procesy, jako např. **regulace**, které se bez vyhodnocení akce zpětnou vazbou nedají realizovat.

Přes pojem zpětné vazby se dostáváme k dalšímu principu robotiky: **cyklovému řízení**. Tento pojem představuje programové řízení automaticky se opakující časové posloupnosti elementárních a ohraničených pohybů výkonných mechanismů robota nebo manipulátoru, kde zpravidla kontakt se záložkou dává signál pro zahájení další činnosti. Nemusí zde být přitom využita přímá zpětná vazba do počítače, ale při potvrzení dokončení nebo při určitém stavu činnosti lze přímo spustit činnost jinou, na první nezávislou.

Protože váš manipulátor budete ovládat pouze 8 bitovým počítačem, nepamenejte na to, že pokud spustíte naráz více procesů a kontrolu zpětné vazby provádíte v cyklu, případně v basicu nebo pod přerušením, je třeba myslet na to, že kontrolní signál z ovládaného zařízení by měl mít dostatečnou délku trvání, aby mohl být zaregistrován. Pokud nelze dostatečnou délku signálu dost dobře zajistit, je třeba, aby program kontroloval takovouto činnost

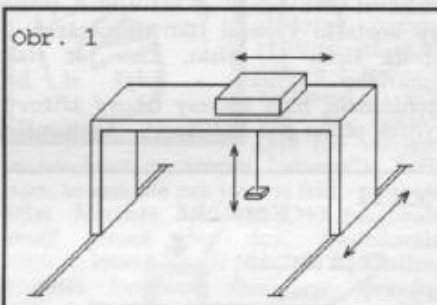
prioritně a byl pokud možno napsán ve strojovém kódu.

Neméně důležité je také vždy si uvědomit v jaké **časové souslednosti** mají činnosti probíhat, tzn. zda se např. pohne dříve to či ono rameno atd. Musíme přitom také myslet na to, aby robot zbytečně nevykonával pohyby navíc, tedy aby jeho pohyby byly **optimalizovány**.

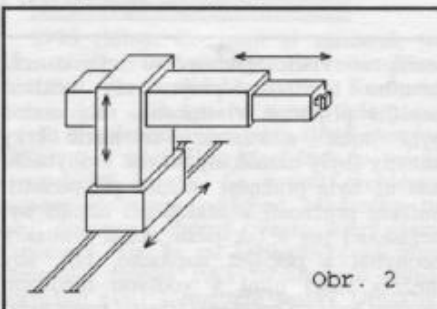
Tolik tedy k úplně základním zásadám a nyní se podíváme jak takový robot může vlastně vypadat.

Konstrukční typy robotů a manipulátorů:

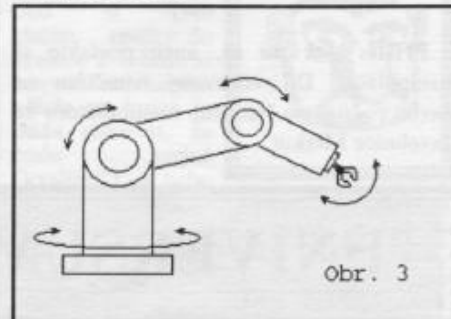
Portálové manipulátory (obr.1) jsou pro většinu z Vás konstrukčně nejjednodušší typem. Jsou v podstatě automatizovanou obdobou portálového jeřábu, z čehož také vyplývá jejich funkce: přemisťování předmětů z místa na místo. Hlavními mechanickými pohyby jsou zde posuny po kolejnici a zpravidla pouze vertikální pohyb výkonné hlavy, resp. **čapadla zvaného efektor**. Protože takovýto manipulátor se dá velice snadno postavit např. ze stavebnice Merkur, popíšeme si stručně náměty a rady na jeho stavbu v příštím pokračování seriálu.



Konstrukčně složitější jsou **teleskopické roboty** (obr.2), jež se dají také sestavit z nějaké stavebnice. Jejich ovládání je stejně snadné jako ovládání předchozího typu, ale mechanické části vyžadují odbornou konstrukci teleskopických ramen s nejčastěji ozubnicovým převodem.



Nejsložitějším typem jsou **roboty s kloubovým mechanismem** a pantografy (obr.3). Ty vyžadují nejen složitou mechanickou a odborně provedenou konstrukci, ale také jejich softwarová podpora je o stupeň náročnější. Do takové konstrukce by se neměl rozhodně pouštět žádný začátečník.



Existují samozřejmě rovněž **roboty s kombinovanými mechanismy**, např. s kombinací kloubů a teleskopických ramen.

Po výčtu všemožných typů se už tedy podíváme, co všechno budeme ke konstrukci vašeho robota, či spíše manipulátoru (robot je přeci jenom obecnější pojem) potřebovat:

- 1) Tak především budete potřebovat součástky, z nichž postavíme mechanickou kostru celého manipulátoru. V našich podmínkách se jako optimální jeví známá stavebnice Merkur. Je dostatečně variabilní, aby se z ní dal postavit celkem slušný výtvor. Majetnější z Vás si mohou pořídit specializované stavebnice firem Fishertechnik a LEGO. Ti zručnější, nejlépe zároveň majitelé vlastní dílny, se mohou pokusit o vlastní mechanické provedení podle fantazie.
- 2) Potom budete potřebovat alespoň trochu výkonné motorky k mechanickým posuvům. K ovládání otočných součástí se poohlédněte po motorcích krokových nebo servomotorech. Dále si obstarajte nějaké mikrosplnače na senzory a snímače. Nezapomeňte ani na kvalitní elektrické vedení, tedy dráty a šňůry, především pak na stínění, určené k vedení signálů v logice TTL. No a samozřejmě, že budete potřebovat mraky elektrosoučástek na stavbu modulů interface.

Na závěr jedna rada pro nedočkavce, kteří se pustí do nějaké konstrukce již po tomto dílu seriálu: než se ale pustíte do dalších experimentů, připomeňte si obecnou zásadu číslo jedna: **všechna**

postavená zařízení a moduly, stejně jako sestavené programky (především nastavení režimu 8255 v nich) po sobě vždy důkladně **PŘEKONTROLUJTE!** Předjete tím přinejmenším pochybnému fungování systému, v horším případě pak poškození počítače.

-Studio 22-

Příště: *Jak na to*, aneb postavte si manipulátor. Díl věnovaný námětům na stavbu počítačem řízeného manipulátoru ze stavebnice Merkur.

Pomůcka

pro čtenáře seriálu ZXM a robotika:
Přehled označení spojových míst ve schématech v seriálu:

Označení místa - popis

PX - vstupní/výstupní pin portu

OUTX - posílený výstup z portu (za indikátorem)

ZOX - posílený výstup z portu (za zesilovacím tranzistorem)

INX - chráněný vstup na port

INDX - připojení indikátoru pro vstup na port

(Port X = A, B, C bity 0..7)

Připomínky k tomuto seriálu zasílejte výhradně na adresu: *Sinclair Club, P.O.Box 132, 363 01 Ostrov.*

UNIVERSUMální klávesnice pro „Gumáka“

Úvod...

Vážení, už jste si asi zvykli, že vás neustále otravují nějakými články, které se týkají software, tentokrát, ač je to neuvěřitelné, vás budu otravovat také článkem, který se týká „hardware“.

Problém!

Jak jste si také mohli přečíst v minulém čísle, podařilo se mi zničit membránovou klávesnici u Spectra 48 (Gumák) asi za rok provozu počítače (hraní her si prostě vyžádalo svou daň). Ocitnul jsem se, jako mnoho vlastníků tohoto počítače přede mnou a jistě i mnoho vlastníků za mnou, před problémem: Co si počít?

Co s ním?

Shánět náhradní klávesnici se mi nechtělo, vlastně jsem ani nevěděl kde, a tak jsem se pustil do opravy svépomocí.

První pokusy:

Nejprve jsem se několikrát (přesně 3x) pokoušel o výrobu nové membrány, tedy přesněji řečeno, něčeho, co by se nové membráně alespoň trochu podobalo, a co by ji funkčně plně nahradilo.

Krize...

Výsledky se více či méně membráně opravdu podobaly, funkci také zpočátku zhruba plnily, ale většinou po velice krátkém čase došlo k trvalému spojení některých kontaktů. Vykoušel jsem různé dostupné materiály - papír, umělou hmotu, alobal - ale bezvýsledně.

Hledání cesty

Když selhaly pokusy o vyřešení problému standardní cestou, začal jsem uvažovat o tom, co bych mohl použít jiného. Nejprve mě napadla možnost využít nějakou klávesnici z vyřazeného počítače - jednu jsem dokonce sehnal, jenže byla poněkud „bezkontaktní“, tudíž bez hlubších znalostí elektroniky nepoužitelná. Navíc by to vyžadovalo dost velké úpravy „krabice“ počítače.

Nápad

Co kdybych zkusil vyrobit nějaké miniaturní (v porovnání s např. zvonkovým tlačítkem) kontakty, které by se umístily pod gumovou klávesnici a nahradily by membránu? A tak jsem si vzpomněl na nejoblíbenější materiál všech elektroamatérů a jim podobných lidí - cuprexit. Co kdybych zkusil vyrobit kontakty přímo na cuprexitu.

Prototypy

Základní myšlenka byla na světě - vezmu cuprexit (Nebo snad cupracart? Ono je to jedno) a vytvořím z něj základovou plochu pro kontakty, které připájím na vybraná místa tak, aby šly stisknout přes gumovou klávesnici. První typ kontaktu vypadal jako na obrázku 1 (tlustá šipka je místo, kam jde tlak gumového výlisek - když si ho prohlédnete, mají klávesy takový křížový výlisek přímo pod tlačítkem), ohnutí mělo

mohl jsem ho použít. Kontakty jsem vyrobil z měděných plíšků (dobře se pájí) ze starých relátků (kvůli potřebné pružnosti).

Sít kontaktů

Nejspíš víte, jak vypadá klávesnice z hlediska připojení tlačítek - je to matice 4 x 10 - čtyři řádky a deset tlačítek na každém. Na jednostranném cuprexitu nebylo možno vytvořit rozvod ke všem kontaktům (cesty by se křížily), musel jsem udělat mnoho přemostění - nejprve jsem použil slabý drát s izolací, ale ukázal se jako příliš tlustý a zbytečně moc zvedal gumový výlisek klávesnice. Nakonec jsem použil tenké drátky odmotané z vyhozeného transformátoru - mají totiž lakovou izolaci a přitom tloušťku desetiny milimetru. Aby se neupevňené drátky nepoškodily, zalil jsem je kanagomem.

Provozní zkušenosti

UNIVERSUMální klávesnici používám bez problémů už šestý rok. Za tu dobu se mi několik přiletovaných plíšků uvolnilo, ale stačilo vzít pájku a bylo po závadě.

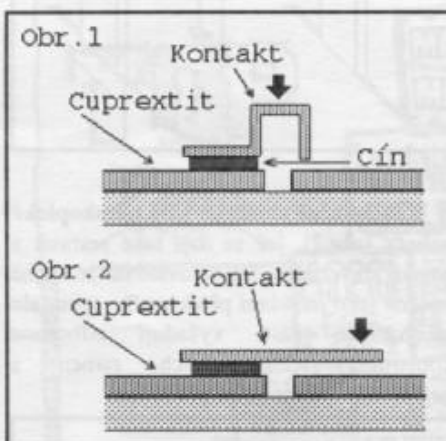
Po čase se mi „klávesnice“ zanesla jakýmsi zeleným „svinstvem“ - podle různých zdrojů je to nějaký silikonový olej, který pouští snad přímo gumový výlisek. Toto „svinstvo“ se dostává i mezi kontakty a cuprexit - čas od času je ho potřeba vyčistit (to když jde nějaká klávesa podezřele ztuha, stává se to tak jednou za čtvrt roku). Pokud si někdo z vás bude chtít něco podobného vytvořit, radím vám: vložte si mezi klávesnici a gumový výlisek nějakou tenkou folii, která zabráni znečištění klávesnice.

Přesné informace a rozměry si budete muset zjistit sami - záleží na tom, jaký materiál seženete a použijete, stejně tak rozdělení cuprexitu na plochy, umístění kontaktů a propojení.

Závěrem

Informace o tom, že ZXM přinese fotografii mé klávesnice na titulní straně se nezakládají na pravdě....

- Universum -



směřovat vstříc gumovému výlisek. U prvního typu se projevovaly některé nepříliš příjemné vlastnosti - klávesnice byla „tuhá“ a kontakty se navíc brzy unavily (byly namáhány hlavně v ohybech, kde už byla pružnost snížena při ohnutí). Snížení pružnosti kontaktů při ohnutí byl nežádoucí jev a tak jsem zkusil kontakty neohýbat a připájet naplocho, tak, aby mezera mezi nimi a vodivou měděnou vrstvou byla co nejmenší (tlustší papír nebo čtvrtka). Druhý prototyp pracoval dobře a

Jedna poněkud temná historie (děsivá, ale pravdivá)

A je to tady, už to v sobě nemohu dusit, a musím se s někým podělit o své zkušenosti. Až dosud jsem si myslel, že mám jen takovou nehoráznou smůlu, ale stále víc a víc se utvrzuji v názoru, že smůla asi nebude všechno. Proto bych se rád dozvěděl, jsem-li sám takhle postižený, a následující příběh je ve skutečnosti výzvou (a možná trochu varováním). Nevím, zda bude otištěn (pakliže ho čtete, tak možná ano) a kde - mně by přišla jako nejvhodnější rubrika Počítače a horror, jenomže ona se jmenuje poněkud jinak...



1988: Někdy těsně před Sivistrem zakoupil jsem (na dluh - půjčil jsem si od rodičů) Didaktika Gamu (6200,- Kčs). A učil jsem se s ním zacházet. Což nebylo tak snadné, neboť jsem narazil na spoustu překážek, neřkuli přímo vad. Velice kladně hodnotím fakt, že záruka byla prodloužena na tři roky. Díky, Skaličtí! Úplně mi také zůstal rozum stát, když mi opravili zadarmo i vady (utržená klávesnice), které jsem si způsobil sám. To zasluží zvláštní veřejné poděkování a když o své anabázi někde vyprávím, neopomenu tuto skutečnost zdůraznit. Co je to ovšem platné, když nejmenší tři z pěti oprav (přesný počet neviduji, ale bylo jich více než je zdrávo) jsem nezavinil, když závady byly způsobeny nekvalitou použitých komponentů! Nicméně - těsně před koncem záruky mi byl počítač uveden do skutečně funkčního stavu, a protože jsem až dosud nic neplatil (vyjma poštovného v několika případech), smířil jsem se se situací, a byl rád.

1992: Zachtělo se mi kvalitnějších zvuků, i objednal jsem si ve Skalici interface Melodik (690,- Kčs), který jsem také obdržel - leč nefunkční. Reklamoval jsem ho tedy a má reklamacce byla vyřízena skutečně obratem. To by bylo v pořádku, jenomže... poslali mi zpátky nejen toho samého Melodika, ale i v původním stavu! Na můj rozhořčený dopis reagovali zasláním už skutečně funkčního výrobku a omluvou, kterou jsem přijal - konec konců, jiní dopadli i hůře. Jenomže to mi nestačilo - nahrávání z kazety mi připadalo pomalé, a zatoužil jsem po disketové jednotce. Nakonec jsem peníze (3999,- Kčs) sehnal a D40 obdržel. Funkční. A se zárukou jednoho roku, což se mi vědlo jako prokleté málo - vezměte si můj příklad: Když jsem D40 obdržel, měl jsem právě počítač v (naštěstí poslední) opravě, pak jsem se připravoval na maturitu, pak byly prázdniny a já nebyl doma, nehledě na to že nějakou dobu trvá, než uživatel vůbec začne s D40 pořádně pracovat, sežene kontakty apod. ... Přesto, že jsem s počítačem v té době moc nedělal, zjišťoval

jsem, že jaksi není vše ideální - při porovnání s mechanikami známých byla ta moje pěkně hlučná a ne vždy po sobě přečetla vše, co zapsala. Pak přestala fungovat úplně, poslal jsem ji, kam jinam, do Skalice, jenomže "závada sa neprejavila". A opravdu, jakž takž od té doby šla. Jako reakci na můj dopis ohledně této situace jsem obdržel od p. Kondly nabídku, abych do Skalice zaslal celou sestavu, že se pokusí ji plně zfunkčnit. Odmítl jsem, bylo mi líto poštovného, a byl jsem rád, že jsem rád.

1993 (jaro): Rozpad Československa a s tím další komplikace, když si D40 opět postavila hlavu (stejně jako poprvé, tentokrát jsem ovšem čekal skoro týden, zda se to samo nespraví, abych se zase neblamoval). Ve Skalici radí Kompakt Servis ve Veselí nad Moravou. A další problém - přístroj už není v záruce, protože je odlepená nálepka (dříve to nevařilo...?) a opravu si pěkně zavcákám. A ať si dodám i počítač, zadarmo mi sladí činnost jeho a D40. A tak jsem opravu zavcákal, vyzvedl



jsem si na to z knížky, a byla pěkně "mastná" (cca 1000,- Kč), neboť mi pan Macháček (ředitel KS) tak dlouho slaďoval činnost přístrojů, až si za úpravu počítače naučtoval více než za opravu D40 (co všechno se neschová za termín "zdarma"). Zuřil jsem, to ano, ale pak jsem si řekl - co se dá dělat. Jenomže opravená D40 fungovala téměř přesně dva dny. Následovala stížnost, kromě Veselí putující i do Skalice, a další, bezplatná (tentokrát opravdu) oprava. Přístroje byly testovány po dobu dvou dní a fungují, píše pan Macháček. Potvrzuji jeho slova - mně doma fungovaly také dva dny. Telefonický dotaz a následuje další oprava. Závada byla v kabelu (stejně jako minule) a teď už je sestava opravdu funkční. Píše pan Macháček. Jeden den, druhý... třetí! Výborně, funguje to, radost nezná mezí! Ale zná - týden, a je vše při starém. Vypravím se za panem Macháčkem osobně a ať si mě nepřeje!

1993 (léto): Co jsem si zamluvl, to jsem i splnil - v polovině července jsem navštívil Veselí nad Moravou a Kompakt Servis (cca 250 km od místa, kde jsme trávili dovolenou). Tam jsem zjistil spoustu zajímavých věcí. Pan Macháček je velmi sympatický člověk. Se svými problémy nejsem sám, pro podobné zákazníky tu mají dokonce odborný termín, který jsem zapomněl. To, že mi D40 připojená přes Melodika zničila FAT tabulku, je úplně normální (!!) - majitelé sestav Didaktik Gama - Melodik jsou tu téměř denními hosty (pan Macháček dokonce vyjádřil

názor, že se měl možnost seznámit se všemi, kdo si k DG Melodika koupili...), leč není jim pomoci, je to prostě tak, a nechcete-li, aby Vám byly ničeny diskety, tak Melodika prostě nepřipojujte! A to nejdůležitější - u mé D40 odešla mechanika, čímž se přístroj stává hromádkou šrotu. Měl jsem na výběr ze dvou možností: buď si D40 strčím... zpátky do krabice a seženu si novou mechaniku někde na burze (a budu riskovat, že bude stejně "kvalitní"), a nebo mi tu namontují novou, spolehlivou, originál japonskou (zvláštní je, že na krabici stojí, že je "MADE IN CHINA"), a to za pouhých 1300,- Kč. Zvolil jsem druhou možnost a zatím nelituji, za a) funguje bezvadně (ovšem druhý den provozu ještě nekončí...) a za b) ty peníze nebyly moje (i když za c) je budu muset vrátit).



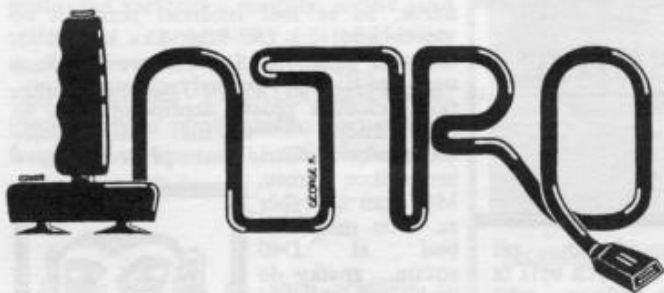
To je (prozatím) vše. Když si to tak shrneme, resp. sečteme, za ty peníze, co jsem do toho zatím vrazil, jsem už mohl klidně mít Amigu, a to nepočítám náklady na dopravu (za ty bych si mohl k té Amize koupit TV modulátor, ne-li rovnou monitor), z čehož rozhodně příliš velkou radost nemám.

Každého asi napadne, jestli bych to udělal znovu, jestli bych si zase koupil Didaktika. Odpověď zní, možná paradoxně: ano, koupil. Přesto, že jsem byl občas vzteky bez sebe, vytvořil jsem si ke svému miláčkově zvláštní vztah, který nepochopí ten, kdo nezažil, a jen hrozně nerad bych se s ním loučil. Je ovšem otázkou, jestli si ho koupí ti, co si přečtou tenhle blábol...

A tento příběh je pravdivý, neboť vám ho 25. 7. 1993 vyprávěl sám

- MB alias Crazy Egg -





ČERNÁ KRONIKA

* Jistý M.B. z M.B. (pouze shoda okolností) formátoval (zbytek si přečtěte v předchozím čísle, je naprosto stejný).

* Po zjištění, že na Slovensku jsou (podle jistého časopisu) velice populární hry domácí produkce (fakt, že jejich distributor a vydavatel zmíněné tiskoviny jsou jeden a tentýž, v žádném případě přece nemůže hrát nějakou podstatnější roli...), se tým severočeských programátorů rozhodl, nechat celou situaci zhodnotit Institutu pro výzkum veřejného mínění. Pracovníci ústavu došli k závěru, že oblíbenost činí asi 16.34% (ovšem nebyli dotazováni nekuřáci nad 72 let). Na otázku, která z oněch mnoha NAJ... her toto ohodnocení získala, jsme se dozvěděli, že všechny dohromady.

* **UNIVERSUM** si uspořádal již druhý ročník "Putování kolmo po vlastech českých" (tentokrát to byly čtyři etapy: Beroun - Slapy - Sázava - Ondřejov (120), Ondřejov - Kutná Hora - Pardubice - Třebechovice pod Orebem - Nové Město nad Metují - Jaroměř - Sadová (183), Sadová - Nový Bydžov - Chlumec nad Cidlinou - Mladá Boleslav (82) a poslední část Mladá Boleslav - Kokořínsko - Štětí - Roudnice nad Labem - Most (122)). V první etapě došlo pouze k drobnému připálení účastníka, druhá etapa bude znamenat přelom - těsně před **Kutnou Horou** přišel účastník na poměrně jednoduchý způsob, jak upravit **PROMETHEUS** pro 128-ku tak, aby zdrojový text mohl zaujmout 64KB, tatáž úprava by byla možná i pro **DESKTOP**. Třetí etapa přinesla hned několik pohrom - ztrátu mapy a šroubu přidržujícího nosič a zadní blatník, totální destrukce předního blatníku a jednoho drátu na předním kole, dále setkání s **MB&DG** (jako obvykle bez **DG**), naopak návštěvy u **SCALEX** a **J.&J. Flašků** se obešly bez trvalejších následků, závěrečná část se (až na jeden prasklý drát na zadním kole) obešla bez větších potíží, pouze silný protivítr a nepříjemně studené počasí zanechaly na zdraví účastníka dočasné stopy. Celková destrukce dopravního prostředku nepřekročila 15%, účastníka pak asi 20%. Největší dojem na účastníka učinil **Nový Bydžov**, kde po nápisu **Vítejte v Novém Bydžově** následovala značně hrbolatá dlažba.

* **George K.** dokončil **SQ-Tracker** a odjel s ním a se **Scalexem** navštívit **Brno**. Zatímco **Scalex** a **Qjeta** byli s editorem vcelku spokojeni (obzvláště jméno se jim líbilo), třetí tester (**Betabyte**) vznesl tolik připomínek, že by vydaly na zcela nový program (hlavní připomínka byla k názvu programu). Oproti starému **Soundtrackeru** má **SQ-Tracker** daleko dokonalejší editaci, poskytuje větší hudební možnosti a bez problémů spolupracuje se všemi normálními diskovými radiči.

* Za nejgeniálnější hru **Scalex** vyhlásil svoji "**Main Rebel DUCK**". Hra má až neuvěřitelně jednoduché ovládání a hladký průběh, takže ji mohou bez problémů hrát i úplní začátečníci. O distribuci se zatím neuvažuje.

CO SE PŘIPRAVUJE...

* **J. Flaška** vymyslel geniální využití víceúčelové tiskárny **BT-100** - z "tiskárny" se stal hudební nástroj - už nebudete potřebovat žádný zesilovač (program přineseme v některém z příštích kompletů **Public Domain**).

* **Depeche Code** opět v akci! **Exotic Fruits** a **George K.** pokročili ve výrobě svého nového **AY-dema**, ve kterém bude hrát hudba z **SQ-Trackeru**. Hlavní část dema se jmenuje **SHUCK** a paroduje přehnaně populární polský **SHOCK**. Že se jedná o skutečnou bombu, pochopí každý, kdo bude mít moci obě dema srovnat.

* **Universumův CALCULUS** dělá čest sloganu "software nové dimenze" - nejenže svými možnostmi daleko převyšuje všechny ostatní tabulkové procesory na **Spectru**, ale svým vzhledem a pojetím editace navíc silně připomíná prostředí písčkových **Windows** (jak jen to ve stísněných podmínkách na **ZX** jde).

KŘÍŽOVKA

Absence křížovky v minulém čísle se setkala u čtenářů **INTRO** s nevolí - dostali jsme několik desítek zklamanych dopisů, ve kterých se nás ptáte "kde byla" nebo "proč nebyla" a "jestli bude" - můžeme Vás uklidnit: **TADY JE**. V tajence najdete návod, jak křížovku jednoduše vyluštit.

TŘÍČTVRTĚMOZAIKOVÁ OBČAS STŘÍDAVÁ DEVÍTISMĚRKA SE SKRYTOU LEGENDOU

zleva vpravo: 1. - 2. - 3.
zprava vlevo: 1. - 2. - 3. - 4.
shora dolů: 1. - 3.
zezdola nahoru: 3.
zepředu: 4. - 3. - 2. - 1.
zezadu: 4. - 4. - 4. - 4.
z boku: 3. - 2.
dokola: **tajenka**
na místě: 2.

Pokud by se snad někomu křížovku podařilo vyluštit, zašlete řešení na adresu redakce, samotného autora by Váš postup zajímal.

NESOUTĚŽ!

Dříve než přejdeme k čtvrtému nesoutěžnímu programku, musíme vyřdit dvě věci: Jednak se omlouváme bratrům **Flaškům**, že se nám občas pletou jejich křestní jména, a zdůrazňujeme, že autorem všech uvedených zářků je **J. Flaška** a nikoliv **J. Flaška**, jak jsme mylně uvedli. Tou druhou věcí je **NESOUTĚŽ!** samotná - v prvním díle jsme psali, že do **NESOUTĚŽ!** se nemůžete nijak přihlásit a i přesto nám do redakce neustále chodí spousty dopisů s příspěvky. Žádáme Vás - žádné další už neposílejte, i tak jsme zavaleni. Díky!

Ale teď už k dnešní bombě: pro milovníky **Tomahawku**, **F-19**, **Comanche** či **Simulátoru jízdy na kole po vodorovné, nekonečně dlouhé silnici**, tu máme doopravdickou lahůdku - **Simulátor jízdy výtahem** (zjednodušená verze). Ti z Vás, kteří tuší, že autorem není nikdo jiný, než **J. Flaška** (tentokrát jsme jméno otiskli správně!), nejsou daleko od pravdy...

```
10 LET a=1
20 PRINT "Jsi v "; a ; ". patre."
30 PRINT "Do kterého patra chces jet? (1-6)": PRINT
40 PAUSE 0: LET i$=INKEY$
50 IF i$<"1" OR i$>"6" THEN GO TO 40
60 IF VAL i$=a THEN GO TO 40
70 FOR i=1 TO ABS (VAL i$-a)
80 BEEP .01,i: NEXT i
90 LET a=VAL i$
100 GO TO 20
```

Sám autor nám k programu nic bližšího neřekl, ale proslýchá se, že koncem roku by se na softwarovém trhu měla objevit senzace: **Simulátor jízdy rychlovýtahem v mrakodrapu**. Pokud se tak skutečně stane, **NESOUTĚŽ!** bude samozřejmě při tom.

- Intro připravil **George K.** -



Ve kterých prodejnách můžete zakoupit programy od PROXIMY?

- PROXIMA, obchodní dům Labe, Ústí nad Labem, tel. 047/24201 kl. 262
- 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, Štůrova 1, Košice
- BONO s. r. o., Hlavní 134, Prešov
- ELEKTROSERVIS Kocman, SNP 1443, Považská Bystrica
- WADOWSKI, 28 října 243, Ostrava - Mariánské hory
- CONSUL, Pálenická 28, Plzeň
- KOMPAKT servis, Masarykova 1192, Veselí nad Moravou
- JRC, Chaloupčického 1913, Praha 6
- CS COMP, Jindřišská 273, Pardubice
- ELEKTRON servis, U opatrovny 238/12, Liberec
- ELEKTRONIK Malina, Duchcovská 15, 415 01 Teplice
- G&H, Novomeského 7/15, 911 01 Trenčín
- PERPETUM, Ševčenkova 22, Bratislava.

Obchodní zastupci firmy PROXIMA:

- Petr Lapšínský, Písečná 22, Cheb, 350 02
- David Béréš, Demlova 26, Jihlava, 586 01
- Ing. Petr Juras, Otavská 1108, Písek, 397 01
- Vlastimil Holer, Alejní 2757, Teplice, 415 01
- Karel Zýbal, Dolina 644, Vnorovy, 696 61
- Jiří Schejbal, Ječná 504/7, Liberec, 460 05
- Martin Humpolík, Strakatého 11, Brno, 636 00
- Antonín Mastný, Sadová 10, Valašské Meziříčí 757 01
- Michal Čopík, Třeboňská 528, Borovany, 373 12
- David Viktora, Máchova 357, Stráž pod Ralskem, 471 87
- Pavel Bláha, Pionýrská 3, Bruntál, 792 01
- Michal Jehlár, Troubky 187, 751 02

Prodejcem těchto programů (dealerem) se může stát i Vaše firma po podepsání dealerské smlouvy. Dealerské slevy činí 10 až 30%, dle odběru. Podrobnosti získáte na požádání na naší adrese:

PROXIMA, box 24, 400 21 Ústí nad Labem.

ZX magazín, časopis pro uživatele počítačů ZX-SPECTRUM a kompatibilních. Vydává: PROXIMA-software v Ústí nad Labem. Povoleno pod číslem MK ČR 5293. MČ: 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 8.10.1993

Cena Kč 20,-



PROXIMA-software v. o. s.

post box 24

400 21 Ústí nad Labem

||| 000000