



602

SPECTRUM

3

88



A co INDEX ?

=====

Konečně dlouho očekávaná akce INDEX se dostává do závěrečné fáze příprav. A není divu, že její uvedení do provozu trvalo tak dlouho. Po úvahách, kdo bude moci být jejím účastníkem, jaké informace bude obsahovat, jakým způsobem bude zajištěn její provoz, evidence, finanční účast - to byly problémy, k jejichž řešení bylo nutno přistupovat tentokrát opravdu pečlivě.

Myšlenka INDEXu se objevila již před několika lety. Při vzniku MIKROBAZE se zkoušely různé varianty zpřístupnění informací. Během řešení se narazilo na problémy autorské, materiální, finanční i personální. Po dlouhých úvahách, pokusech, po jednáních s ústředními orgány Svazarmu i s dalšími zúčastněnými i nezúčastněnými organizacemi, po ověření zájmu, účasti, po - tento výčet by mohl pokračovat dosti dlouho. Některé problémy se nepodařilo z nejrůznějších důvodů vyřešit dosud. Ale o tom později.

Teprve loni došlo k otevření Střediska vědecko-technických informací Svazarmu, jehož provoz je plně zajištěn naší 602. základní organizací Svazarmu. Toto středisko je napojeno na mezinárodní informační systémy, a je škoda, že se dosud nepodařilo vyřešit plnou spoluúčast dalších organizací, jejichž problémy přesto pomáhá řešit. Jak je známo, služby jsou poskytovány pouze členům 602. základní organizace Svazarmu nebo členům Svazarmu v ní hostujících. Ale vzhledem k širokému výběru činnosti 602. ZO finanční náklady na členství nejsou velké. I tato otázka je dosud řešena a doufejme, že i tento problém se podaří odstranit.

Ale teď už konečně přistupme k vlastnímu INDEXu.

INDEX je rozsáhlý databázový systém informací o programovém i systémovém vybavení nejrůznějších výpočetních prostředků. Chce odstranit dosavadní roztržitost těchto informací. Jeho obsah je velice jednoduše přístupný všem zúčastněným. Způsob zařazování i výběru informací je řešen pomocí nejmodernější výpočetní techniky - počítačem typu PC-AT.

Služba INDEX bude poskytována všem členům Svazarmu, SSM a ČSVTS. Vzhledem k velkému zájmu budou informace poskytovány pouze písemně a to přednostně aktivně zúčastněným účastníkům, t.j. těm, kteří poskytnou své informace a tím aktivně pomohou ostatním zájemcům.

Přístup k základním informacím - t.j. název programu nebo hardware - bude zajištěn ve středisku VTI. Vlastní styk s INDEXem je zajištěn pomocí kartotékových lístků. Jejich formát je A5 a líc má tuto podobu:

```
.....  
: INDEX : :  
: : : :  
: : : :  
: : : :  
: 4006/465 : 602. ZO Svazarmu :  
: Novak Jindrich : :  
: Jistinova 1233 : Wintrova 6 :  
: Kolodeje : :  
: 113 47 : 160 41 Praha 6 :  
: : : :  
.....
```


Vlevo nad adresou člena je jeho členské číslo.

Druhá strana karty - rub má 15 řádků. Obsahují veškeré informace potřebné pro evidenci INDEXu. Uvádíme její příklad:

```

.....
.
. 1. Přihlášení programu
. 2. 570412/943
. 3. Kohout Miroslav Ing.
. 4. Václavova 15, Praha 5, 159 00
. 5. ZX Spectrum
. 6. 0.8
. 7. magnetofon
. 8. tiskárna Seikosha GP 500
. 9. stroj.kód
. 10. Připojení tiskárny Seikosha GP 500
. 11. program a zapojení interface pro připojení tis-
. 12. kárny. Program je uveden i ve výpisu.
. 13.
. 14.
. 15. ne
.
. Podpis a datum: Kohout 15.7.1988
.
.....

```

Každý řádek může mít maximálně 50 znaků. Je nutno důsledně dodržovat řádkování. Jinak by mohlo dojít ke zkreslenému načítání údajů do počítače.

1. řádek: návěst druhu služby. Smí obsahovat pouze jednu z těchto pěti variant:

- 1. Přihlášení účastníka
- 2. Změna osobních údajů
- 3. Přihlášení programu
- 4. Odhlášení programu
- 5. Požadavek na program

2. řádek: rodné číslo účastníka

3. řádek: příjmení, jméno a tituly účastníka

4. řádek: ulice, číslo, místo a PSC bydliště účastníka

5. řádek: typ počítače

6. řádek: rozsah programu v paměti - v kB

7. řádek: vnější paměťová media, na kterých je program nahrán

8. řádek: zvláštní hardware při použití programu

9. řádek: programovací jazyk, ve kterém je program napsán

10. řádek: název programu

11. až 14. řádek: stručný popis programu. Pokud se jedná o firmní program, je nutno název firmy uvést na konci popisu. Před názvem je nutno udělat hvězdičku.

15. řádek: adresa ano - na požádání sdělit.
adresa ne - majiteli programu bude zaslána adresa
žadatele.

Poslední řádek obsahuje vlastnoruční podpis účastníka a da-
tum odeslání nebo odevzdání lístku.

Veškerou korespondenci zasílejte na známou adresu:

602. ZO Svazarmu
Wintrova 8
160 41 Praha 6

Do levého horního rohu napište zřetelné označení INDEX .

Co na závěr ?

Služba INDEX chce odstranit dosavadní roztráštěnost informací
v různých organizacích a klubech. V současné době má INDEX za-
tříděny informace pro 32 typů nejrůznějších počítačů, ať osob-
ních či větších. Jeho snahou je ale poskytovat informace v co
nejširším rozsahu.

Doufáme, že účastníky INDEXu se stanou hlavně zástupci klubů
a kroužků, které aktivně pracují s počítači. Možnost komplexních
informací jim umožní efektivní práci s počítači a odstraní du-
plicitní řešení problémů.

Pracovníci podnikových výpočetních středisek zde budou moci
získat komplexnější informace o aktuální práci s výpočetní tech-
nikou.

602. ZO Svazarmu má ve výpočetní technice dlouholeté zkušeno-
sti. Ve své historii zaznamenala i neúspěchy. Některé byly zavi-
něny špatnou přípravou akcí, jiné akce neuspěly pro nerozhodnost
příslušných orgánů. Ale vždy šlo o dobrou snahu pomoci ostatním.

Dnes má naše 602. ZO dostatečnou podporu Svazarmu, SSM i pří-
slušných státních institucí. Služba VTI úspěšně přispívá k in-
formovanosti počítačových nadšenců 602. ZO. Službou INDEX chceme
pomoci i ostatním pracovníkům s počítači. Je na nich, jak bude
tato služba pomáhat. Naskýtá se konečně možnost soustředit dnes
rozptýlené a neúplné informace pod jednu "střechu".

Vzhledem k současným kapacitním a provozním možnostem je
služba INDEXu určena především aktivním účastníkům. Věříme ale,
že možnost získání aktuálních informací k efektivnější práci s
počítačem zaujme každého fanouška počítačové techniky.

=====
Pomozte si usnadnit práci s Vaším počítačem.
=====

INDEX.602 + STREDISKO VTI.602 + PROGRAMOVA NABIDKA MIKROBAZE
chtějí být Vašimi dobrými pomocníky ve Vaší práci.
=====

Počítačové jazyky na Spectru

=====

Jeden z největších půvabů počítače ZX-Spectrum spočívá v tom, že svému uživateli přinese "až na stůl" řadu počítačových jazyků, takže při výuce lze všechny příklady z učebnic napsat do počítače, vyzkoušet, měnit, pokusit se o vlastní programy. O didaktickém významu Spectra tady již ostatně jednou byla řeč - v souvislosti s Pascalem Mikrobáze. Ne všichni uživatelé Specter se však dostanou tak daleko. Je citelný nedostatek manuálů a hlavně dobrých učebnic, počítačových jazyků je mnoho (a stále vznikají nové, údajně dokonalejší), času je málo. Asi je to škoda, protože počítač, který u Vás zahálí na stole, nebo je používán na ničení podivných kosmických těles, chybí studentům a vážným zájemcům - a to se ve svých důsledcích někde nakonec projeví.

Opravdu stojí za to, povystrčit hlavu za obzor BASICu a seznámit se s tím, co nám zástupci hrozně chytrých pánů vymyslely pro komunikaci s chytrými stroji. Takový jednoduchý program, jakým je třeba přenášení "Hanojských věží", to zvládnete s trochou snahy ve všech jazycích, o kterých se zmíníme. Přitom ale oceníte různé výhody a dokážete alespoň trochu zasvěceně poukázat na to, co se v kterém jazyku nepovedlo.

Situaci máme ulehčenou tím, že Spectrum bylo vedle počítačů BBC zavedeno na školách ve Velké Británii, takže mnoho softwarových firem vyvíjelo na státní zakázku různé interpretery a compilery, které pak ovšem ještě zpeněžily na volném trhu.

BASIC

Jazykem číslo jedna na mikropočítačích je a asi ještě dlouho bude BASIC (navzdory rozrušeným pedagogům). Jednu z jeho verzí, ne tak úplně nepovedenou, ale trochu zvláštní, máte v paměti ROM. Protože BASIC můžeme používat v dialogu s počítačem a snadno jej editujeme, ubírala se celá řada pokusů o rozšíření možností počítače cestou rozšíření BASICu Spectra. Dá se říci, že hned od počátku se do čela těchto snah dostala firma Betasoft se svými prvními verzemi BetaBasicu (1.0 a 1.8). Například takový SigmaBasic 1.10T (1984 Sinsoft) není nic jiného, než BetaBasic 1.8 "obohacený" o možnost zrychleného nahrávání. Pro velký úspěch pak následovaly další verze BetaBasicu, oblíbené i u nás. Nejúspěšnější verze BetaBasic 3.1 (1985) je v literatuře označována za nejlepší BASIC na osmibitových počítačích vůbec. Má parametry volané hodnotou nebo odkazem, lokální proměnné a rekurzi, 128 různých okének na obrazovce (každé s vlastní velikostí písma a s vlastními atributy), grafickou úpravu listingu ve stylu Pascalu (postupující "schůdky"), řadu nových klíčových slov a funkcí, pohodlnou editaci, vestavěné hodiny, možnost psát po jednotlivých písmenech a mnoho jiného. Doporučujeme alespoň vyzkoušet, protože jen tak si můžete vytvořit v BASICu struktury se smyčkami DO-WHILE-UNTIL a EXIT IF, zkratka spoustu věcí, o kterých se jinak v učebnicích BASICu jen dočtete. Abychom jen nechválili, je nutné si připomenout, že to všechno je samozřejmě na úkor volné paměti. V návodu k programu PRO-DOS jsme si řekli, že se třemi a půl kilobajty paměti může PRO-DOS konkurovat výsledkům BetaBasicu, ovšem méně elegantními a méně výchovnými prostředky.

Podobný, méně dokonalý a kratší je Skyline Basic (1984 Ionis Software International).

Jiná cesta k rozšíření BASICu, vedla přes úvahu, zda by nešlo vylepšit chudičké a nepohodlné příkazy pro grafiku (PLOT, DRAW a CIRCLE). V tomto oboru si získal určitou popularitu Mega Basic (YS Software) ve svých několika verzích. Má sice i jiná vylepšení, např. procedury s parametry (méně dokonalé než BetaBasic 3.1 i 3.0), jeho hlavní síla je ale v práci s obrazovkou. Má několik druhů písma (a pěkného), dokáže scrollovat s okénky osmi směry, zkrátka je to BASIC, po kterém asi sáhnete, když budete stát před úkolem vytvořit třeba propagační program do výlohy.

Poněkud monstrózní verzi BASICu představila firma Oasis v podobě svého Laser Basicu (1986 Oasis/Ocean). Jde v podstatě o zdokonalení předchozích programů "White Lightning" a "Machine Lightning" a jejich "zabudování" do BASICu Spectra. Důmyslný systém těžko zapamatovatelných příkazů slouží téměř výlučně k tvorbě grafických her a k práci se sprajty. Na výsledné hře nepoznáte, že běží v BASICu a navíc ji můžete připojeným překladačem ještě asi dvakrát zrychlit.

Pro ty, kteří chtějí vyzkoušet programy z časopisů, psané pro Commodore 64, je určen emulátor CBM 64 BASIC firmy Softek. Imitace je věrná, včetně chybových hlášení.

PASCAL

Jazyk, o kterém se mluví. I do povědomí laiků pronikly ozvěny ofenzívy jeho zastánců proti BASICu, vedené ve sdělovacích prostředcích hlučně a bez zvláštních výsledků. Za jeho hlavní výhody považují čitelnost programů a rychlost běhu přeložených programů. Na rozdíl od jeho zastánců si myslím, že třeba pro výuku dětí v programování se vůbec nehodí. Nemá totiž ve své standartní podobě žádné příkazy pro grafiku a zvuk, je tedy "němý" právě v oblasti, která se u mikropočítačů rozvíjí nejrychleji (a která je pro děti nejpoutavější). Víím, že to všechno lze do Pascalu "doprogramovat", ale to asi po dětech chtít nemůžeme. Skalní fandové Pascalu necht' zkusí přeložit třeba jednoduchou řádku BASICu: PRINT AT 10,5; FLASH 1; "(nějaký symbol UDG)".

Dlouho byl jediným Pascallem pro Spectrum je překladač firmy Hisoft ve svých verzích HP4S a HP4TM1.6. Polský časopis Bajtek dokonce nazval svůj seriál o Pascalu pro Spectrum "Bez výběru". Má mnoho odchylek od normy, mizerný editor a nečekané výpadky. Je ale fantasticky rychlý, takže ve standartních testech (benchmarks) se díky němu Spectrum umísťuje mezi šestnáctibitovými počítači. Tuto rychlost snadno využijeme, protože překladač vytvoří blok strojového kódu, který pak z BASICu spustíme повеlem RANDOMIZE USR 24608. Několik našich autorů se pokusilo k překladači připojit vlastní editor. Pokud mohu soudit, nejoblíbenější verzí je HP 80 s volbou 32/64 znaků na řádek, s nečíslovanými řádky a řadou příjemných funkcí editoru.

Alternativou je DC Pascal, který pod názvem Mikrobáze Pascal byl předmětem naší recenze a následné polemiky na stránkách tohoto časopisu. Hlavní výhodou je soulad s normou, takže v rámci výuky můžete publikované programy prostě do počítače bez úprav opsat. Získané návyky pak použijete při práci na jiných počítačích s "normovaným" Pascallem. Má dobrý editor i průvodní dokumentaci. Hlavní nevýhodou pak je nemožnost vytvořit jednoduše spustitelný samostatný program ve strojovém kódu.

Dosud málo rozšířený je kvalitní překladač firmy MIRA, asi poslední Pascal pro Spectrum. Výsledkem překladu jsou samostatné programy, má poměrně málo odchylek od normy.

FORTH

Forth byl v módě na přelomu sedmdesátých a osmdesátých let, tedy zhruba ještě v době, kdy Spectrum vzniklo. To byl jeden z důvodů, proč byl hned druhým jazykem pro Spectrum (po BASICu). Druhým vážným důvodem je to, že je nenáročný na paměť, takže byly úspěšné pokusy nahradit jím BASIC v paměti ROM. Tyto ROM-FORTH se skutečně v zahraničí prodávaly, ovšem bez komerčního úspěchu. V oné době se prodával i první mikropočítač s jiným jazykem než BASICem, Jupiter Ace s Forthem - rovněž neúspěšně. Zmiňuji se o tom proto, že nekritičtí zastánci Pascalu dnes obviňují výrobce počítačů z jakési nevysvětlitelné posedlosti BASICem. Popis jazyka se samozřejmě vymyká možnostem tohoto časopisu. Snad nejvýstižněji si můžeme přirovnat přechod od BASICu k Forthu jako přechod od automobilu s automatickou převodovkou k ručnímu řazení. Je to pracné a nepohodlné, ale jste tak blíž ke stroji a docílíte vyššího výkonu. Výhodou je zvýšená efektivita práce při psaní delších programů a hlavně rychlost běhu programů (původně byl Forth vymyšlen pro řízení astronomických radioteleskopů, tedy pro náročné úlohy v reálném čase). To je docíleno hlavně důslednou prací se zásobníky (stacky), takže počítač nic nehledá, vše má vždy na vrcholu zásobníku - čísla, příkazy, adresy. Nevýhodou je zejména nesrozumitelnost zápisu, připomínající trochu programy pro programovatelné kalkulátory. Forth je totiž jediný jazyk, který si vymyslel programátor pro svou vlastní potřebu. Autor Ch. Moore těm potrhlym zkratkám asi rozumí.

Nejpopulárnější (a oprávněně) je fig-FORTH 1.1A firmy Abersoft z r.1983. To "fig" je zkratka "Forth Interested Group". Tato skupina fanatiků Forthu dozírá na to, aby verze Forthu na různých počítačích byly naprosto stejné, takže tím vzniká jakási mezinárodní norma. Důsledky jsou podobné jako u Pascalu; grafika i zvuk přijdou zkrátka. Navíc chybí výpočty s desetinnou čárkou.

Značné odchylky od tohoto standardu vykazuje ZX-FORTH (1982 Artic). Má také jen celočíselnou aritmetiku, ale převzal z BASICu příkazy pro grafiku, barvy a zvuk a podřídil je syntaxi Forthu. Hlavní nevýhodou je špatná průvodní dokumentace (kterou navíc mnoho uživatelů ani nemá).

Několik firem se pokusilo o Forth s plovoucí desetinnou čárkou, tyto pokusy však připomínají Forth jen svou "obrácenou polskou notací" (FP 50 apod.).

'C'

Jazyk C vznikl v Bellových laboratořích v USA jako pokračování jazyku B. Jeho vznik byl od počátku spjat s operačním systémem UNIX, takže na většině osmibitových počítačů ho vůbec nenajdeme. V zahraničí však je oblíben mezi profesionálními programátory na šestnáctibitových počítačích, zejména na IBM-PC a jeho klonech, protože existuje řada kvalitních překladačů. Syntaxí i filosofií vychází, podobně jako Pascal, z Algolu. Proti jednoduchému Pascalu má však téměř nepřehledné možnosti. Je to něco jako když jste zvyklí stavět z deseti typů kostek Lega a najednou před Váš někdo vyklopí celý

sortiment výrobce. K dispozici je třeba 11 různých operací přiřazení. Téměř vše můžete naprogramovat hned několika různými způsoby. Málokterý programátor dokáže v této situaci udržet přehlednost a koncepčnost programu.

Pro Spectrum (pokud vím) existuje pouze jeden překladač firmy Hisoft z r.1984. Jak je u této firmy zvykem, má špatný editor a v rychlosti trochu pokulhává za Pascalem téhož výrobce, ačkoli má pouze celočíselnou aritmetiku (tu si na úkor paměti můžete naprogramovat). Krásně ale pracuje s knihovnami podprogramů; Můžete mu pustit celou kazetu s hotovými podprogramy a překladač si z nich vybere jen ty, které v hlavním programu voláte (a připojí je k programu). Základní knihovna podprogramů je dodávána přímo s překladačem. Připojená dokumentace se v řadě případů dovolává referenčního manuálu autorů Kernighana a Ritchieho, takže si jej musíte opatřit také. Výhodou (které, pokud vím, využívá JZD Akrokombinát Slušovice) je přenositelnost na různé druhy počítačů.

LOGO

U nás (a v celém světě) je oprávněně známo jako jazyk pro komunikaci dětí s počítačem. Kreslicí želvička dokáže upoutat okamžitě dětskou pozornost. Jazyk svou přehledností, podrobnými chybovými hlášenými, možností připojení robota k počítači a dalšími vlastnostmi dokáže tuto pozornost udržet a svěst správným směrem. Není ale všemocný - v USA se Logo ve školách učí a výsledky ukazují, že přítomnost člověka-učitele je nutná, jinak děti začnou tvořit "jitrnicové", nestrukturované programy. Většina našich uživatelů zůstane u želviček, ale je to škoda, protože s Logem lze postupovat přes středoškolskou a vysokoškolskou matematiku až do oblasti umělé inteligence. Logo má např. částečně "vestavěn" jazyk LISP pro práci se seznamy.

Po starších pokusech napodobit Logo z BASICu přišel přímo od výrobce počítače kvalitní jazyk, Sinclair Logo v.1.6 (1984 SOLI/LCSI). Jako spoluautor je uvedena firma LCSI, která pro Logo vykonává podobnou službu, jako pro Forth skupina "fig" - snaží se o vytvoření programů pokud možno pro každý rozšířený počítač a zajišťuje jejich kompatibilitu. Ze standardu (DR LOGO firmy Digital Research) jsou vypuštěny funkce pro práci se sprajty a barevné možnosti jsou přizpůsobeny omezeným možnostem Spectra. Jinak máme před sebou plnohodnotné Logo s asi 130 klíčovými slovy, slušným editorem a volnou pamětí, která pro dětské programy nadlouho vystačí. Didaktické možnosti jsou neocenitelné, nevýhodou je pomalost.

Poněkud chudší slovník nalezneme u LOGO 1.4 firmy JiPoSoft.

LISP

Jazyk, který jako první nebyl určen pro numerické výpočty, ale hlavně pro práci s tzv. seznamy. Je kompromisem mezi imperativními jazyky (to jsou ty, kde počítači poručíme, jak má postupovat - tedy všechny dosud uvedené) a funkcionálními jazyky (to jsou takové, kde počítači vysvětlíme logické souvislosti a on jede dál podle zabudovaných pravidel aby dosáhl požadovaného výsledku). Je poměrně jednoduchý, má pět základních funkcí a zná jedinou datovou strukturu - seznam. V době svého vzniku (konec 50tých let) budil nadšení, protože daleko více než ostatní jazyky připomínal činnost lidského mozku. Mezi uváděnými jazyky

je dědečkem, ale dodnes se používá pro výzkumy v oblasti umělé inteligence (zejména ve své verzi Golden Common LISP). Je implementován na mnoha typech počítačů (i našich), bohužel ovšem bez mezinárodní normy, takže jednotlivé verze se od sebe často podstatně liší. Jednoduchost původního jazyka lákala různé upravitelky k jeho rozšiřování a vylepšování. I když se dočtete pravý opak, programy v něm psané nevynikají přehledností a (zastánci LISPu prominou) stejných výsledků lze docílit v Prologu s daleko menší námahou. Viděl jsem ovšem v LISPu program, který anglicky skládal japonská pětiverší Haiku a výsledek na mne hluboce zapůsobil.

Pro Spectrum znám pouze Spectrum LISP 1.3 (1983 Serious Software). Lze jej doporučit těm, kteří si chtějí s učebnicí v ruce LISP "osahat", ovšem lze očekávat nepříjemná překvapení vzniklá odchylkami od popisované verze. V literatuře se obvykle popisuje tzv. Pure Eval LISP nebo EVALQUOTE LISP, což jsou verze poněkud odlišné.

PROLOG

O Prologu toho bylo v poslední době napsáno mnoho. Existoval celkem bez povšimnutí, dokud si ho Japonci nevybrali pro své projekty umělé inteligence. Na rozdíl od ostatních jazyků už vůbec neříkáme počítači, jak má něco udělat, ale co víme a co od něj chceme. Tento uživatelský přístup je lákavý a může vést i k úvahám o brzkém soumraku "druhé gramotnosti", podaří-li se vylepšit komunikaci s počítačem a přimět jej, aby ještě více algoritmů uměl "sám od sebe". Prolog dokáže odhalovat nečekané vztahy a kromě inteligentních databank se tedy hodí zejména na nejrůznější expertní systémy.

U nás je dost rozšířen micro-PROLOG T 1.0 (1983 LPA, údajně původně maďarské provenience). Je vybaven dobrou dokumentací a knihovnou příkladů a funkcí. Do Prologu má však dost daleko, takže může uživateli poskytnout jen pocit práce s tímto jazykem.

Prolog 80 v. 3.8 (1985 PeNoSoft) je však standartní, tzv. Edinburghská verze tohoto jazyka a lze ji jen doporučit.

To je tak asi zhruba všechno, o čem se s Vámi mohu podělit. Víím, že to není vyčerpávající seznam. Chybí Mira Fortran, který nemám, chybí CAD BASIC, který je BASICem jen podle názvu, chybí DLAN (Display Language) a podobné exotické vymyšlenosti. Nezmínil jsem se ani o Karlovi, protože je mi zatěžko do této rodiny zařadit věc, jejímž jediným výsledkem je šmejdění panáka po dvorku. Není mi známo, že by pro Spectrum existoval Smalltalk či APL, o jiných jazycích nemluvě.

Určitě je ale spousta programů, o kterých nevím. Vaše doplnění jen uvítáme. Se smutkem však musíme konstatovat, že britské firmy se už asi do tvorby dalších programů tohoto typu pro Spectrum nepohrnou a že tento přehled je možná přehledem závěrečným.

Samozřejmě se budeme snažit, abyste popisované programy mohli (s výjimkou Mikrobáze Pascalu) získat na schůzkách klubu. Vy všichni můžete přispět překládáním manuálů a hlavně jejich přepsáním do Taswordu v těch případech (už je jich málo), kdy takto šířitelný manuál dosud neexistuje.

Co s hrou...

STRIKE FORCE HARRIER

=====

Strike Force Harrier je letecký simulátor firmy Mirrorsoft, který byl v recenzích i žebříčcích vysoce hodnocen a tomu odpovídá i jeho cena - na kazetě 9,95 `.

Simulován je AV-8B Harrier II, letadlo s krátkým nebo kolmým vzletem a přistáním (V/STOL). Má jeden turbínový motor Rolls-Royce Pegasus (tah 95,8 kN) se dvěma páry postranních trysek. Pilot je může nastavit do jakékoli polohy, od vodorovné až po svislou. V programu jsou ale jen tři polohy:

- vodorovná (pro horizontální let velkou rychlostí)
- svislá (pro vertikální let a přistání)
- šikmá (horizontální let malou rychlostí).

Indikátorem polohy je šipka na žlutém pozadí v levé polovině palubní desky. Trysky zdviháme tlačítkem E a spouštíme tlačítkem D.

Je jasné, že při vertikálním nastavení trysek nemůže fungovat kormidlo (J a K), protože není obtékáno potřebným proudem vzduchu. V programu je to přesto možné.

Na obrazovce dominuje HUD - Head Up Display, speciální sklo, na které se promítají nejdůležitější údaje, aniž byste museli spustit zrak z pohledu ven. Na levé straně jsou dva ukazatele: z levé strany stupnice je to VSI - ukazatel svislé rychlosti, tj. klesání nebo stoupání. Z pravé strany pak ASI - ukazatel dopředné rychlosti. Jeden, velký dílek stupnice představuje 100 mil/h. V horním levém rohu je ukazatel kursu (ve stupních), v horním pravém rohu výška ve stopách. V dolním pravém rohu je ukazatel podélného náklonu (ve stupních, znaménko + nebo - označuje stoupání nebo klesání). Uprostřed je mířidlo pro kanon, které je zároveň indikátorem příčného náklonu.

Pod HUD jsou dvě skupiny po čtyřech signálních světlech: indikátor nebezpečí a indikátor stupně poškození letounu (při poškození klesá postupně výkon motoru).

V levé polovině palubní desky je indikátor polohy brzd 'B' (červená = zabrzděno, zelená = odbrzděno) a již zmíněný indikátor polohy trysek. Pod nimi je multifunkční displej, zobrazující stav motoru, výzbroje a údaje palubního počítače. Z mnoha přístrojů pro kontrolu motoru zůstala pouze lineární stupnice pro plyn (THRUST) a palivoměr (FUEL). Tlačítkem 'Q' vyvoláme stav výzbroje. Dva rovnoběžné bílé pruhy označují stav munice do kanonu, žluté symboly pumy a červené symboly rakety vzduch-vzduch.

Největší přístroj je taktický displej. Vidíte na něm svou polohu, polohu základny a vrcholky hor - bíle. Zároveň je zobrazena poloha protivníka červeně.

V pravé polovině palubní desky jsou světelné indikátory polohy klapek (F) a podvozku (G), kde červená vždy značí vysunutí. Pod nimi je radar.

Start je typickou zvláštností tohoto letadla. Ke zkrácenému startu použijeme následující postup (skutečnému Harrieru stačí 500m dráhy):

- zabrzdit kola a zatáhnout klapy
- nastavit turbíny do šikmé polohy

- nastavit plný plyn
- odbrzdit
- po dosažení potřebné rychlosti lehce letadlo zvednout
- zatáhnout podvozek
- vysunout klapky (jste v malé výšce, blízko minimální rychlosti!)
- srovnat trysky do vodorovné polohy (v min. výšce 100 stop).

Hlavní riziko je v tom, že zatím nedokážete tímto způsobem zase přistát. Přistání mimo základnu končí (v dobrém případě) hlášením "unprepared ground", tj. "neupravený povrch". Toto riziko vyloučíme svislým startem a přistáním:

- nastavíte trysky kolmo
- dáte plný plyn
- ve výšce 900 stop lehce skloníte nos letadla; až letadlo dosáhne potřebné rychlosti směrem vpřed, skloníte trysky.

Při letu zjistíte, že Harrier II je velice stabilní letoun (hornoplošník s křídly do obráceného V a velmi nízkým těžištěm). Pro zatáčení není třeba ovládat směrové kormidlo. Rychlost stoupání je na simulátoru přehnaná - za 13,5 sec. dosáhnete výšku 10 000 stop. Ve skutečnosti je to asi 13 000 stop/min. Tím vychází program vstříc temperamentnějším hráčům, toužícím po boji.

Kolmo můžete přistát teoreticky kdekoli, ale pouze na základně bude letoun opraven a získáte palivo a munici. Navíc hrozí na nepřátelském území zajetí. Abyste našli základnu, stisknete 'T' a vyberete si základnu (tlačítka M,N,SPACE a SYMBOL SHIFT). Na HUDu se objeví kurs, kterým se vydáte. POZOR! Navigace funguje jen směrem dopředu, takže je-li základna za Vámi, ukazuje kurs rovně dopředu! V bloku základny pak nastavíte plný plyn a skloníte trysky na 45 stupňů, čímž snížíte rychlost. S takto sklopenými tryskami docílíte náklon -3 stupně bez klesání. Když základnu uvidíte, spustíte trysky do kolmé polohy (letadlo přitom ztratí výšku), nastavíte podélný náklon mezi -12 a -10 stupňů a tak se přibližujete středu základny. Vysunete podvozek. Nad základnou se ozve zvukový signál a zpráva na displeji. Jemně stáhnete podélný náklon na nulu a srovnáte případný příčný náklon (přejde-li hodnota podélného náklonu do plusu, začne letoun opět stoupat!). Motor dokáže zvednout Harrier kolmo do 2000 stop, což můžete využít i pro rychlé spouštění z velké výšky. Nakloníte-li trysky při rychlosti větší než 250 m/h, upozorní Vás palubní počítač na chybu. Při klesání pozorně sledujte VSI - max. povolená svislá rychlost při doteku se zemí je 25 stop/sec. Je-li výškoměr na nule, vypnete přistávací navigaci ('U') a stáhnete plyn. Zkontrolujete, zda bylo letadlo opraveno - když ne, nejste na základně. V tom případě musíte na plný plyn vyletět do výšky 900 stop a zde kombinací kniplu a kormidla dovést letadlo nad základnu.

Harrier, ve srovnání třeba s Tomahawkem, nespotřebovává tak rychle palivo, ale přesto s ním musíte zacházet ekonomicky.

Terén, nad kterým létáte, je rozdělen na "bloky". Název bloku je složen ze dvou písmen; první se mění ve směru východ - západ, druhé ve směru sever - jih. Jeden blok je právě tak velký, aby se vešel na taktický displej.

Nejvyšší vrcholky mají 9000 stop. Na taktickém displeji kromě nich vidíte rampy raket země-vzduch (SAM), skupinky tanků a nepřátelská letadla. Stiskem 'W' dostanete informaci o přelétávaném bloku. V programu je ale chyba (viz [3]) - někdy

počítač na 'W' nereaguje a je nutné hru přerušit a znovu spustit dvojitým stiskem tlačítka 'A'.

Máte čtyři základny: jedna je v bloku AA, kde hru začínáte (volá se tlačítkem 'N'), druhá v bloku AC ('M'), třetí v BC ('SPACE') a poslední v bloku BA ('SYMBOL SHIFT'). Po stisknutí 'T' si musíte jednu ze základen zvolit pro přistání. Je-li základna daleko, objeví se hlášení "fortrac weak".

AV-8B má pod křídly osm závěsů, na kterých může nést 4175 kg výbroje, což je polovina toho, co nesla ve II. svět. válce celá americká perut'. Součástí výbroje je automatický kanon ráže 25 mm s 250 náboji, který pálí krátkými dávkami. Nepřetržitá palba trvá asi 22 sec. Kanonem můžete ničit všechny druhy cílů. Proti letadlům máte dvě samonaváděcí rakety AIM-9L Sidewinder. Mají rychlost 2,5 M, takže ve středních výškách dostihnou všechny druhy letadel. Cíl napadají vždy z předního sektoru. Raketu zacílíme stiskem 'Y'. Je-li cíl v dosahu, upozorní nás palubní počítač zvukovým signálem a zprávou "target reguided". Střelíme stiskem CAPS SHIFT nebo FIRE na joysticku. Tlačítkem 'U' se vrátíme zpět ke střelbě kanonem.

Pro útok na pozemní cíle máme tři zápalné pumy po 500 kg. Před bombardováním musíme stisknout 'R'. Na HUDu se objeví vodorovná čárka v místě, kam bomba dopadne. Manévrováním překryjeme tuto rysku s cílem a stiskneme tlačítko pro střelbu. Proti radarem řízeným raketám můžeme vystřelit falešné cíle s vysokým odrazem (chaff) pomocí 'C', proti raketám s infračerveným naváděním pak světlice (flares) pomocí 'H'. Palubní počítač nemůže poskytnout informaci, jaké navádění raketa má, na raketu ovšem upozorní ("missile warning"). Letadla používají obou typů raket, rakety země - vzduch jsou vždy řízeny radarem.

Pohyblivé odpalovací rampy a tanky se brání lehkými protiletectkými kanony, které mohou zasáhnout letoun ve výškách 700 až 2000 stop. Nad 2000 stop jsou zase účinné rakety země - vzduch, takže pozemní cíle ničíme výhodně z výšky méně než 700 stop. Je-li Harrier zachycen nepřátelským radarem, objeví se hlášení "enemy radar" (sen všech řidičů). To je zpravidla předeheur k odpálení rakety země - vzduch.

Krajina je zobrazena technikou sprajtů, která se ve svém účinku blíží technice 3D, použité v Tomahawku. Letadla jsou podobná jako v simulátoru Fighter Pilot - MIG 23. Ve skutečnosti bychom proti těmto nadzvukovým letadlům neměli se dvěma Sidewindery mnoho šancí. Program nezná katapultování, ale snad to není velká škoda ve srovnání s A.C.E., který vesele a úspěšně katapultuje z desetimetrové výšky.

Všechny letecké simulátory na mikropočítačích mají jednu hlavní vadu - malé zorné pole. Proto jsou vybaveny radary neskutečného výkonu. Na obrazovce radaru o záběru 360 stupňů a dosahu 5 mil vidíme polohu letadel i raket a navíc relativní výšku nejbližšího letadla (P) i nejbližší rakety (M). Pozemní cíle nemůžeme radarem sledovat, podobně jako ve skutečném letadle.

Minimální rychlost pro přechod do kolmého stoupání je 450 m/h. V této rychlosti tedy můžete provést výkřut i další prvky letecké akrobacie. Harrier se zde prezentuje jako velice obratný letoun.

Přehled hlášení na displeji:

fortrac weak	- příliš velká vzdálenost pro navigaci
select GS	- vyber pozemní základnu
over GS	- jsi přesně nad základnou
high ground	- jsi nízko nad povrchem
collision predicted	- nezměníš-li směr nebo výšku, narazíš
target reguided	- raketa zaměřena na cíl
enemy radar	- jsi zachycen nepřátelským radarem
missile warning	- míří na tebe raketa
chaff x	- zbylo x falešných cílů
flares x	- zbylo x světlic
block xx	- jsi nad blokem xx
too slow	- letíš příliš pomalu
too fast	- rychlost je vyšší než 800 m/h nebo - podvozek je vysunut při rychlosti větší než 250 m/h nebo - klapky jsou vysunuty při rychlosti větší než 650 m/h nebo - trysky jsou nakloněny při rychlosti větší než 250 m/h.

Ovládání:

=====

nahoru.....	L (nebo joystick)
dolu.....	P (nebo joystick)
kormidlo vlevo.....	J
kormidlo vpravo.....	K
směrovky vlevo.....	Z (nebo joystick)
směrovky vpravo.....	X (nebo joystick)
přidat plyn.....	I
ubrat plyn.....	O
zvedni trysky.....	E
nakloň trysky.....	D
brzdy na podvozku..	B
klapky na křídlech.	F
podvozek.....	G
vyber bomby.....	R
rakety.....	Y
kanon.....	U
radarové terče.....	C
světlice.....	H
střelba.....	CAPS SHIFT (n. joystick)
stav munice.....	Q
pozice.....	W
údaje o motoru.....	V
navigace k zákl.....	T (vypneme U)
základna AA.....	N
AC.....	M
BA.....	SYMBOL SHIFT
BC.....	SPACE
pauza zap/vyp.....	A
zvuk zap/vyp.....	S

Strike Force Harrier není nejdokonalejším leteckým simulátorem na Spectru, tím je asi stále Gunship firmy Microprose. Jeho vadou je složitost ovládání. Proti ostatním simulátorům (i tzv. simulátorům) má však výhodu puntičkářského realismu (až na zmíněné výjimky) a slušnou grafiku.

-mm-

-
- Literatura: [1] Mirrorsoft: Strike Force Harrier (manuál k programu)
 [2] Gaines: Sea Harrier School
 [3] Moj Mikro č. 5/87

Pro začátečníky:

První kroky ve strojovém kódu (pokračování)

Příbuzná instrukce je LDDR. Tady uložíme do HL a DE poslední adresy bloků. Při provádění se zkopíruje obsah bajtu, jehož adresa je v HL do bajtu, jehož adresa je v DE. Potud je to tedy stejné, jako u LDIR, ale pak jsou všechny tři registrové páry zmenšeny o jedničku.

Podobně LDD je verze přenosu jednoho bajtu bez automatického opakování. Většinou je lhostejné, zda použijete LDIR nebo LDDR. Výjimkou jsou případy, kdy dochází k překrývání obou bloků paměti, zdrojového a cílového. Z uvedených údajů jistě sami dovodíte, kterou instrukci je nutné použít, překrývají-li se bloky na začátku zdrojového bloku a kterou při překrývání na konci. Operační kódy instrukcí byly uvedeny v tabulce.

Příklad:

```
10 CLEAR 24999
20 FOR I = 25000 TO 25023
30 READ X: POKE I,X
40 NEXT I
50 DATA 33,0,64,17,112,98
60 DATA 1,0,27,237,176,201
70 DATA 33,112,98,17,0,64
80 DATA 1,0,27,237,176,201
90 CLS
100 FOR I=1 TO 20
110 PRINT INK I/3; PAPER 8-I/3; "111122223333444455556666777788"
120 NEXT I
130 RANDOMIZE USR 25000
140 CLS
150 PRINT AT 12,0; "STISKNI JAKEKOLI TLACITKO"
160 PAUSE 0
170 RANDOMIZE USR 25012
```

Od adresy 25000 se uloží tento strojový kód:

LD HL, 16384	210040	Začátek obrazovky
LD DE, 25200	117062	Začátek oblasti uchování v paměti
LD BC, 6912	01001B	Počet přesunovaných bajtů
LDIR	EDB0	Přesun
RET	C9	Zpět do BASICu

a od adresy 25012 tento kód:

LD HL, 25200	217062	Začátek oblasti uchování v paměti
LD DE, 16384	110040	Začátek obrazovky
LD BC, 6912	01001B	Počet přesunovaných bajtů
LDIR	EDB0	Přesun
RET	C9	Návrat do BASICu

Možná jste se podivili, jak počítač ví, kam se po ukončení podprogramu má vrátit. To si teď vysvětlíme a zároveň uvidíme, jak nejmenší jednotka informace, bit, může být použit k vytváření zajímavých programů ve strojovém kódu. Ale nejprve k zásobníku:

Zásobník

Dosud jsme se zabývali instrukcemi, které slouží k přenosu bajtů z jednoho místa na druhé - hlavně z jednoho registru do druhého a mezi pamětí a registry. V paměti je vyhrazeno zvláštní "skladiště" bajtů, kterému říkáme zásobník. K jeho obsluze slouží zvláštní sada instrukcí. Nejlepší na celé záležitosti je to, že se nemusíme vůbec starat o to, kde je vlastně zásobník v paměti RAM umístěn. To udělá počítač za nás. Jestli opravdu chcete, můžete sice pozici zásobníku v paměti měnit pomocí zvláštních instrukcí, ale obvykle jej ponecháme tam, kam si ho umístí sám procesor po zapnutí proudu do počítače, tj. pod RAMTOP.

Co je vlastně "zásobník"? Je to "štos" bajtů, který, jak jsme si už řekli, začíná pod RAMTOPem. Postupem doby se zvětšuje směrem dolů. Jeho účelem je mít dočasné skladiště, kam můžeme číslo "uklidit" jedinou jednoduchou instrukcí a odkud jej zase můžeme právě tak jednoduše vyzvednout. Funguje to asi takhle:

V mikroprocesoru je zvláštní dvoubajtový registr SP (stack pointer) - ukazatel zásobníku. V něm je uložena první prázdná adresa v zásobníku. Chcete-li nějakou informaci dočasně uchovat, použijete instrukci PUSH k uložení této hodnoty na zásobník. Ukládaná hodnota se však předtím musí nacházet v některém registrovém páru, takže známe instrukce PUSH HL, PUSH DE, PUSH BC a PUSH AF (kombinace registru 'A' a registru příznaků). Při uložení čísla na zásobník je hodnota registru SP zmenšena o 2, takže ihned obsahuje čerstvou informaci o tom, jaká je nová adresa prvního volného bajtu. Jak jsme si již řekli, zásobník se takto zvětšuje směrem dolů.

K vyzvednutí čísla ze zásobníku použijeme instrukci POP. Vyzvednutou hodnotu pak uložíme do jednoho z registrových párů. V rámci provádění instrukce POP je rovněž ihned upravena hodnota registru SP. Operační kódy instrukcí (jedná se o jednobajtové instrukce) uvádíme v tabulce:

Registry	HL	DE	BC	AF
PUSH	E5	D5	C5	F5
POP	E1	D1	C1	F1

Zásobník slouží k usnadnění práce programátora. Potřebujete-li například uschovat obsah nějakého registru po dobu, kdy s registrem provádíte něco jiného, použijete jednoduše PUSH a následně POP. K přesunu čísla z jednoho registrového páru do druhého použijete rovněž nejprve PUSH k uložení čísla na zásobník a pak POP k jeho přesunu do druhého registrového páru.

Operace se zásobníkem však vyžadují určitou opatrnost. Jako první můžete ze zásobníku vytáhnout pouze číslo naposledy vložené - podobně jako papíry ze "štosu". Musíte proto dávat dobrý pozor na pořadí čísel. Zásobník je totiž používán mikroprocesorem i bez Vaší kontroly - to je odpověď na otázku o návratech z podprogramů.

Voláte-li podprogram, ať už v BASICu nebo ve strojovém kódu, návratová adresa je uložena na zásobník. Voláte-li postupně několik podprogramů, jsou jejich návratové adresy

uloženy postupně na zásobník ve správném pořadí, takže se při každé instrukci RETURN vrátíte přesně tam, kam máte. Proto je nezbytné, aby v rámci každého podprogramu byl stejný počet instrukcí PUSH jako počet instrukcí POP (a abyste nepoužili instrukci POP dříve než instrukci PUSH). Jinak poškodíte na zásobníku návratovou adresu a program se zhroutí!

Operace s bity

Mnohem více operací, než pro práci s celými bajty, známe pro práci s jednotlivými bity. Můžeme je však všechny zařadit do několika kategorií. Než se do toho ale pustíme, zkusme si zopakovat, co takový bit vlastně je.

Každý bajt paměti i každý registr mikroprocesoru je složen z osmi bitů. Jeden bit si můžeme představit jako vypínač, který má pouze dva možné stavy: vypnuto a zapnuto. Zapisujeme to jako 0 a 1. Protože ve skupině osmi bitů je 256 možných kombinací nul a jedniček, může jeden bajt obsahovat pouze čísla od 0 do 255.

Víme-li, že hodnota jednoho bitu může být buď 0 nebo 1, nepřekvapí nás, že nejjednodušší operace s bity umožňují nastavení hodnoty určitého bitu na nulu nebo jedničku a zkoumání, zda je hodnota bitu 0 nebo 1. Nastavování na jedničku provádí instrukce SET, mazání na nulu instrukce RESET, zkoumání hodnoty instrukce BIT. Tyto operace lze provádět se kterýmkoli bitem v registrech A, B, C, D, E, H, L a se kterýmkoli bitem v bajtu paměti, jehož adresa je uložena v HL. Když si spočítáte všechny možné kombinace, vyjde Vám 64 možností pro SET, 64 pro RESET a dalších 64 pro BIT. Instrukce SET, RESET a BIT mají délku 2 bajty, přičemž první bajt má vždy hexadecimální hodnotu CB.

Instrukce assembleru pro tyto operace musí mít dva argumenty. Prvním je číslo bitu uvnitř bajtu (0 až 7). Druhým pak je označení registru, nebo sdělení, že se jedná o bajt paměti, jehož adresa je v HL. Tímto sdělením jsou písmena HL v kulaté závorce. Tak např. SET 5,E umístí hodnotu 1 do pátého bitu v registru 'E' a RESET 1,(HL) nastaví na nulu první bit v bajtu paměti RAM, jehož adresa je právě uložena v registrovém páru 'HL'.

Instrukce BIT zkoumá hodnotu určeného bitu a výsledek uloží do příznaku nuly (zero flag). Je-li např. hodnota bitu 1, pak je tato jednička uložena do příznaku Z. Nastavení příznaku Z pak můžeme dále testovat, jak jsme si popsali v minulých lekcích.

Tak třeba BIT 3, (HL) testuje hodnotu třetího bitu v bajtu paměti, jehož adresa je uložena v HL.

V dalším pokračování uvedeme přehlednou tabulku operačních kódů těchto instrukcí.

(Pokračování příště)

Očekáváme Vaše písemné příspěvky. Uveřejněné články honorujeme.

Sinclair 602, technický zpravodaj pro mikroelektroniku a výpočetní techniku. Vydává 602. ZO Svazarmu pro potřeby vlastního aktivu. Odpovědný redaktor Michael Mánek. Adresa redakce: 602.ZO Svazarmu, Wintrova 8, Praha 6, 160 41. Telefon: 32-85-63. Povoleno UVTEI pod evidenčním číslem 87 006. Cena 6,- Kčs dle čcú č. 1030/202/86. Náklad 800 výtisků. Praha, září 1988.