

MIKRO

1989

10



BÁŮZE

technický
zpravodaj
pro zájemce o
mikropočítače

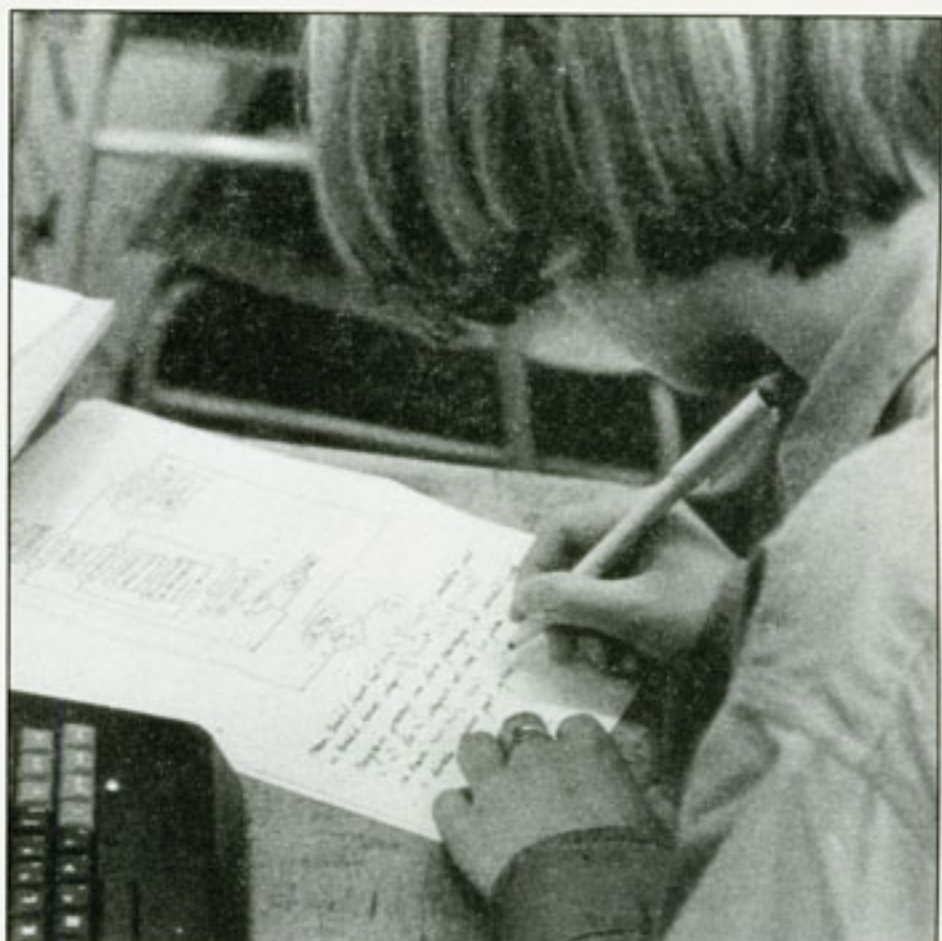
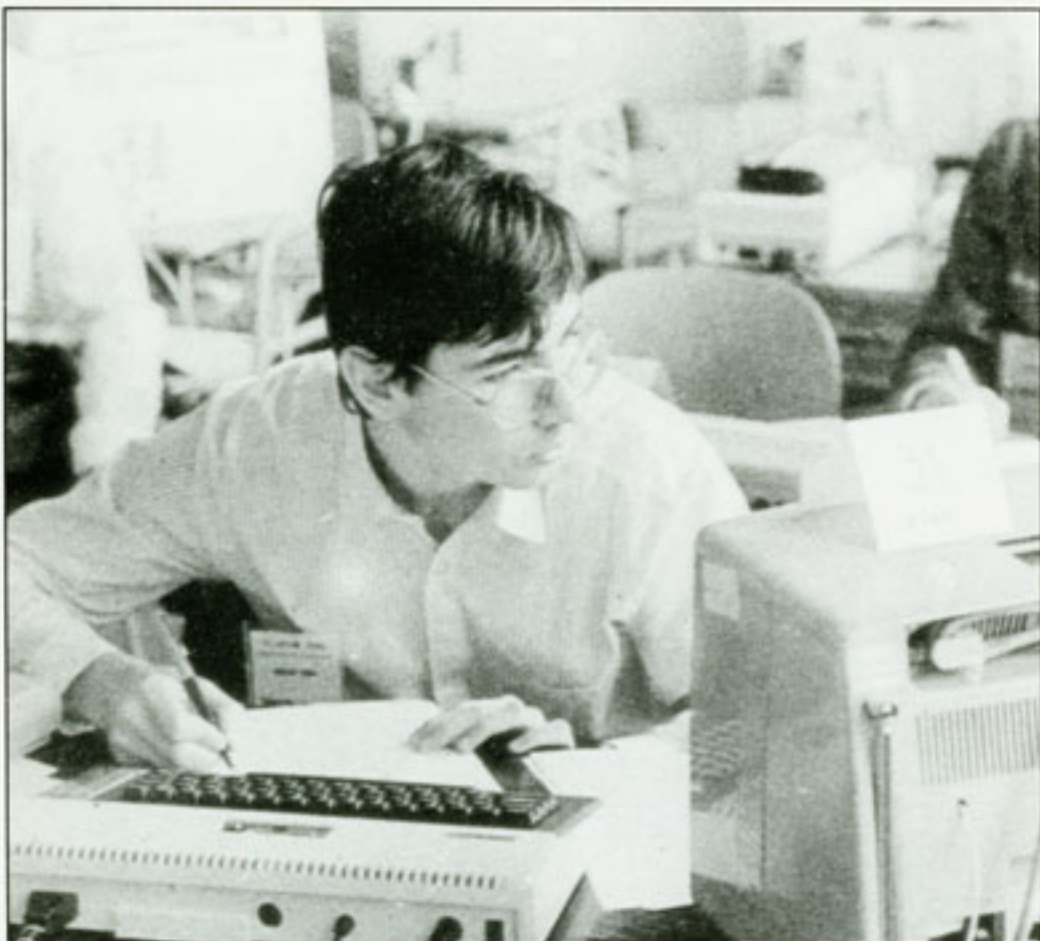
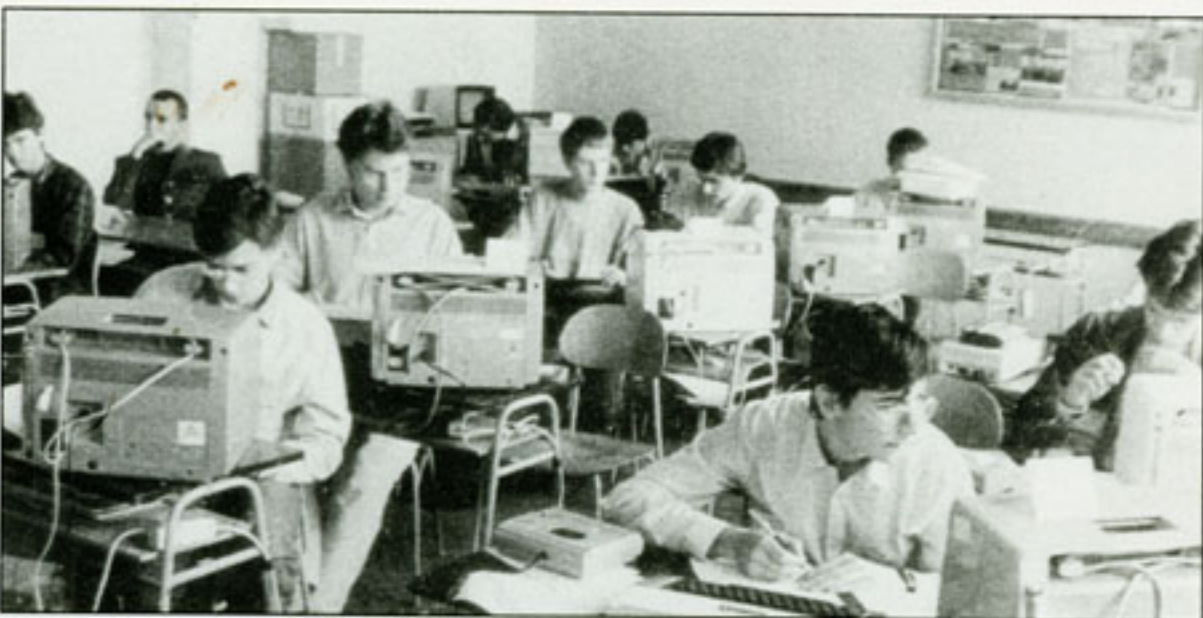
Cena 12 Kčs





CELOSTÁTNÍ SOUTĚŽ PROG-89

TÁBOR 28.-29.X.1989



MIKRO BÁZE

OBSAH

1989/10

Na rozloučenou	1
Dřu, dřeš, dřeme ... Céčko (10)	2
Závada na plotteru-printeru SHARP MZ-116	3
Z88 v roce 1989	4
Personal Computer Show 1989	5
Nepájivá kontaktní pole	5
Postavte si s námi diskový řadič (6)	6
O kohoutkovi a slepičce	9
RSX - rozšíření Basicu Amstrad/Schneider	10
Dopisy, ohlasy	11
Kopirovací programy pro ZX Spectrum	12
Programové vybavení od Robotronu	14
Uživatelské funkce mikropočítače ZX Spectrum	15
Využití přerušeni u počítače Sharp MZ-800	16
Čeští programátoři dobývají svět? ...	18
Seznam známých POKE pro Amstrad CPC	19
Program tvoří program	20
Terabasic Macrosoft	22
Disketová jednotka pro IQ 151 a CP/M	25
Program pro kontrolu českého pravopisu	26
Pomůcka pro návrh plošných spojů	29
Monitor informací	30
Programová nabídka	32

Milí čtenáři,

je mou smutnou povinností Vám oznámit, že máte v rukou poslední číslo Mikrobáze. Další už nebudou. Přestože, jak se obávám, se Vám toto číslo dostane do rukou až někdy v červnu (po volbách), jedná se teprve o číslo 10/89. Proč - to se dozvíte v jiném článku. Pro chronické potíže s vydáváním a s termíny tisku se náš vydavatel rozhodl s Mikrobází už dál nepokračovat.

Náhradou by měl být P+C magazín (pokud se jeho název do té doby nezmění), který by měl vycházet snad od července. Měla by to být náhrada kvalitní, i když, vzhledem k posunu obsahové náplně výrazně směrem k šestnácti- a vicebitovým osobním počítačům, ne zcela rovnocenná. Někdo tyto změny uvítá, někdo bude zklamán. Je pravda, že ve světě už jsou osmibitové počítače spíš kuriozitou, ale u nás jich je, především mezi amatéry, více než dost. (Pozor! Nezaměňovat amatéra s diletantem.) Vždyť i s osmibitovým počítačem lze dosáhnout obdivuhodných výsledků. Vzpomeňme jen na Jiřího Lamače a jeho Lecsoft, Pavla Trollera se skupinou Sinsoft, na konstruktéry teletextových adaptérů a řadu dalších, o nichž se Vás snažila Mikrobáze informovat. Ale život už je takový - vývoj jde nezadržitelně vpřed.

U příležitosti rozloučení s Mikrobází bych jen rád připomenul jména některých spolupracovníků, kteří se v průběhu let v Mikrobází vystřídali a kteří svou pilí a nadšením umožnili v tehdejší složité situaci její vznik. Jsou to především:

Ing. Alek Myslík - jeden z duchovních otců Mikrobáze, první zodpovědný redaktor a dlouholetý spolupracovník redakce.

Ladislav Zajiček - další z duchovních otců celého projektu, který od prvního čísla v roce 1985, s malými přestávkami sestavoval všechna čísla Mikrobáze až do poloviny ročníku 1989.

Ing. Jan Klabal - po řadu let zodpovědný redaktor Mikrobáze, který umožňoval propojení s redakcí AR a tím vůbec existenci Mikrobáze.

Josef Kroupa - vedoucí hospodářské činnosti 602. ZO Svazarmu, dříve redaktor kdysi oblíbeného časopisu Hudba a zvuk, který od samého začátku pomáhal svou zkušeností a postaral se o hospodářské podmínky pro vznik Mikrobáze. Postoupil v průběhu vydávání Mikrobáze nejeden boj s byrokratickými orgány.

To byli ti trvalí spolupracovníci Mikrobáze. Jinak však spolupracovala na projektu Mikrobáze v průběhu let řada lidí, které si zde ani netroufám vyjmenovat, abych snad na někoho nezapomněl. Za všechny připomenu jen paní *Grimmovou* a později paní *Kotěšovskou*, které vyřizovaly členskou agendu a předplatné a které trpělivě vyřizovaly vaše urgencye i reklamace.

Přátelé, dovoluji abych zde i Vaším jménem poděkoval všem jmenovaným i nejmenovaným spolupracovníkům Mikrobáze za všechno, co pro naši společnou věc udělali. Ono to nebylo vždy zrovna jednoduché. Mikrobáze totiž byla typickým samizdatovým časopisem se všemi problémy, které to přináší. Vždyť vlastně po celou tu dobu vycházela jen jako klubový zpravodaj. Proto také bylo podmínkou jejího odběru členství, nebo alespoň hostování v 602. ZO.

Závěrem bych jménem redakce rád poděkoval i Vám - čtenářům - za Vaši přízeň, kterou jste nás zahrnovali a za trpělivost, kterou jste s námi měli. Sám se znovu osobně omlouvám všem, kterým jsem pro nával práce v poslední době nestačil odpovědět na dopis a slibuji, že se budu snažit postupně na Vaše dopisy odpovědět, pokud to ještě bude mít význam. Vy, kterým je také smutno po časopisu typu Mikrobáze, napište mi. Třeba společně něco vymyslíme.

Mikrobáze je mrtva! - Ať žije P+C magazín!

Váš Daniel Meca

Technický zpravodaj Svazarmu pro zájemce o mikropočítače. Vydává 602. ZO Svazarmu ve spolupráci s redakcí časopisu Amatérské radio. Povoleno ÚVTEI pod evidenčním číslem 87 007. Sestavil vedoucí redaktor Daniel Meca. Obálka ak. grafik Jiří Blažek a Daniel Meca. Grafická úprava textu Daniel Meca. Sekretářka redakce Zdeňka Válková. Redakční rada: Petr Horský, ing. Jan Klabal, ing. Petr Kratochvíl, Josef Kroupa, Rudolf Mach, Daniel Meca, ing. Alois Myslík, ing. Josef Truxa. Za původnost a správnost příspěvků ručí autoři. Ročně vyjde 10 čísel. Cena výtisku 12 Kčs podle ČCÚ a SCÚ č. 1030/202/86. Objednávky přijímá a zpravodaj rozšiřuje 602. ZO Svazarmu, Wintrova 8, 160 41 Praha 6.

DŘU, DŘEŠ, DŘEME... CÉČKO /10/

Koncem listopadu jsem měl odevzdat poslední pokračování céčkového seriálu. Jenže...přišlo datum s číslem 17... Jako spousta jiných počítačových izolátů jsem nechal všechny proudy binárních dat protékat mimo, a vstoupil do viru funkčně důležitějších událostí. I když vím o jednom, který bez ohledu na vítr za okny celou revoluci doslova proprogramoval. Díkybohu, ten vítr nepřestává dout...už je leden, já pořád nevím, kam dřív skočit, a Mikrobáze se připomíná stále důrazněji. Mimochodem - všimli jste si, jak si počítače v naší sametové revoluci konečně našly svůj základní uživatelský, komunikační smysl? Nechtě jim vydrží i nadále.

Svůj slib o přehledu a užívání knihovnic funkcí z uvedených (časových) důvodů nemohu splnit. Doufám, že mi odpustíte. Na konci článku uvádím seznam literatury, v níž najdete potřebné. Ale abych neskončil jen samým omlouváním, připojuji prográmek, který je základní kostříčkou editoru. Je určen pro ZX Spectrum a jeho příbuzné. Lze jej samozřejmě přepracovat pro jakýkoli jiný počítač. Program je napsán trochu rozvláčněji, abyste si s ním mohli vyhrát podle libosti. Můžete jej hned začít testovat nabytými znalostmi a v dlouhých chvílích postupně obohacovat další editačními funkcemi podle svých nápadů a potřeb.

Základní myšlenkou programu je sestavování osmi bajtů grafické podoby znaku "shora dolů". Orientace pro umístění znaku podle pozice kurzoru probíhá jen podle horní adresy obrazové paměti, na niž se umísťuje horní bajt znaku. Já tomu říkám "rozvěšování znaků na prádelni šňůru". Pozice znaku na oně šňůře je dána obsahem nižšího bajtu obrazové adresy. Obrazovka je u ZX Spectra rozdělena na třetiny. Jejich první adresy jsou: 4000H, 4800H a 5000H. Prádelni šňůra (nižší bajty adres) jde pro každou třetinu od 00H do FFH. Celá finta je tedy v tom, že při pohybu kurzoru program dává pozor jen na to, kdy dojde k přeplnění nebo "podtečení" nižšího bajtu adresy, a podle toho se pak zachová. Pozice znaku nesmí být nižší než 4000H a vyšší než 50FFH. Při přechodu z jedné třetiny do druhé se adekvátně změní vyšší bajt adresy o plus či minus 8, jinak pracujeme jen s nižším.

Grafická podoba znaků se kopíruje z romky ZX Spectra (8 bajtů mezery - ASCII kód 32 - začíná na adrese 3D00H). Při věšení znaku na šňůru je každý jeho další grafický bajt na obrazovce shora o 256 adres dál než předchozí (viz cyklus for od ř.720).

Kurzorem je bajt z obrazové části barevných atributů. Podle toho, kam s ním s měřujeme, se rozsvěcí patřičná sousední pozice na obrazovce (a zhasíná původní) - funkce poke uloží do patřičné adresy JASNYATR, resp. SEDYATR. Tuto funkci využívá i zápis grafických bajtů znaku do obrazové paměti.

```
20 #define PRVATR 22528 /* adresy atributů */
30 #define POSATR 23295
40 #define JASNYATR 120 /* kurzor */
50 #define SEDYATR 56 /* zhas.kurzoru */
52 #define KODTLAC 23560 /* syst.proměnné */
54 #define FLAGS 23611
60 int nižadr=0; /* nižší TV adresa */
70 int vyšadr=0x40; /* vyšší TV adresa */
80 int atr=PRVATR;
90 int a,tvadr;
100 char kód;
110 typedef char *chp; /* pro poke a peek */
120
130 main()
140 {char testbit,resbit,bajt,c;
150 testbit=32; resbit=223;
160 tvadr=256*vyšadr+nižadr;
170 for(a=0;a<6144;a++)
180 poke(tvadr+a,0);
190 poke(atr,JASNYATR);
200 for(;;)
210 {for(;;)
220 (bajt=c=peek(FLAGS);
230 if(bajt&=testbit) break;
240 )
250 poke(FLAGS,c&resbit);
260 kód=peek(KODTLAC);
270 if(kód==8)
280 (klevý(); continue;)
290 if(kód==9)
300 (kpravý(); continue;)
310 if(kód==10)
320 (kdolů(); continue;)
330 if(kód==11)
340 (knahoru; continue;)
350 if(kód==12)
360 (klevý(); kód=32; znak(); continue;)
370 if(kód==13) break;
380 if(kód.31&&kód<128)
390 (znak(); kpravý();)
400 }
410 }
420 klevý()
430 {if(atr==PRVATR) return;
440 --nižadr; výpočadr();
450 poke(atr,SEDYATR);
460 poke(--atr,JASNYATR);
470 }
480 kpravý()
490 {if(atr==POSATR) return;
500 ++nižadr; výpočadr();
510 poke(atr,SEDYATR);
520 poke(++atr,JASNYATR);
530 }
```

```

530 kdolú()
540 (if(atr>POSATR-32) return;
550 nižadr+=32; výpočadr());
560 poke(atr,SEDYATR);
570 poke(atr+=32,JASNYATR);
580 )
590
600 knahoru()
610 (if(atr<PRVATR+32) return;
620 nižadr-=32; výpočadr());
630 poke(atr,SEDYATR);
640 poke(atr-=32,JASNYATR);
650 )
660
670 znak()
680 (int grzn,grb;
690 kód-=32;
700 grzn=0x3D00=8*kód;
710 tvadr=256*vyšadr=nižadr;
720 for(a=0;a<8;a++)
730 {grb=peek(grzn+a);
740  poke(tvadr+256*a,grb);}
750 )
760
770 výpočadr()
780 (if(nížadr>255)
790  (nižadr-=256;
800   vyšadr+=0x8;)
810  if(nížadr<0)
820   (nižadr+=256;
830    vyšadr-=0x8;)
840 )
850
860 poke(adr,val)
870 (*cast(chp)adr=val;)
880
890 peek(adr)
900 (return *cast(chp)adr;)

```

Rádky 220 a 230 testují, zda je stisknuto tlačítko (stav bitu 5 na adrese 23611, spectrovsky assemblerové BIT 5,(IY+1)). Když není stisknuto nic, program se vrací na řádek 210, jinak pokračuje vynulováním zmíněného bitu na ř. 250. Z adresy 23560 se odebírá kód posledně stisknutého tlačítka a ukládá se do proměnné kód. Na řádcích 270-350 jsou testy kódu a volání adekvátních funkcí. Blok

PC Telex

umožňuje program Telex Link britské firmy Braid Systems, kterým lze realizovat příjem a vysílání dálkopisných zpráv až 8 linkami. Telex Link lze optimálně využít při denní frekvenci kolem jednoho sta dálkopisů. PCW 7/89-str. 110

ZÁVADA NA PLOTTERU-PRINTERU SHARP MZ-1P16

L. Jeřábek, 602. 20

Na náš trh je již delší dobu dodáván plotter-printer SHARP MZ-1P16, jehož cena a kvalita jsou, mírně řečeno, diskutabilní. Bývá již téměř pravidlem, že během poměrně krátké doby dochází k poruše motorků, ať už horizontálního, či vertikálního posuvu. Bohužel znám případ mého klubového kolegy, který během jediného dne uplatnil dvě reklamace. Osobně obdivuji ochotu a trpělivost personálu pražské prodejny Mikropočítače ve Spálené, který má s těmito plottery již své zkušenosti.

Příčinou závad obecně bývá nekvalitní práce výrobce. Za těchto okolností nelze souhlasit s cenou 3 800,- Kčs. Další příčinou bývá přehřívání motor-

testů if si můžete upravit na vhodnější výběr case.

Řádky 390-650 obsahují funkce pro pohyb kurzorem. Funkce znak() zapisuje do obrazové paměti znak odpovídající ASCII kódu stisknutého tlačítka. Funkce výpočadr() hlídá přechody mezi třetinami obrazovky. Funkce pro výmaz (DELETE) tu sice není jmenovitě přítomna, ale je sestavena z pohybu kurzoru doleva a zápisem mezery (viz ř. 310-330). Chod programu se přerušuje stiskem ENTERu (ř. 340).

Tento velmi jednoduchý editor zapisuje přímo do obrazové paměti a neposouvá text žádným směrem. Pro rozumnou editaci je samozřejmě nutné zapisovat znakové kódy do vyhrazené části ramky a pracovat s nimi, jak náleží. Alternativ řešení je hodně, záleží jen na vašem výběru.

Doufám, že vám desetidílný seriálek, během kterého jsem se Céčko učil společně s vámi, něco dal, moc vás nepopletl, a že vás přinejmenším podnítil k osvojení tohoto dnes nejužívanějšího jazyka. Jaký asi bude ten příští? Pokud mohu doporučit, docela hezky na Céčko navazuje třeba Modula-2. Zkusíte se podívat na nějakou učebnici, která se jí věnuje - ono to totiž začíná vypadat tak, že - nestane-li se geopoliticky něco nevídaného - programování bude konečně mít obecný smysl a užitek (oproti nám tak dobře známému "samohonkovému" bastlení pro nic za nic). A co víc si mohou zeleně fosforeskující honáci bajtů přát?

-elzet-

Seznam literatury

- Programovací jazyk C; B.W.Kernighan, D.M.Ritchie; ALFA 1988
- Operační systém UNIX a jazyk C; J.Brodský, L.Skočovsky; SNTL 1989
- Jazyk C w systemie CP/M; J.Bielecki; WKL 1989 (PL)
- Wprowadzenie do jezyka C; J.Bielecki; WNT 1988 (PL)
- Jazyk Si; M.Waite a kol.; Mir 1988 (SU)
- Jazyk Si, vvedenje dlja programmistov; R.Berry, B.Meekings; Finansy i statistika 1988 (SU)
- The C Programming Language, Second Edition; B.W.Kernighan, D.M.Ritchie; Prentice-Hall 1988 (USA)

Komunikace

pomocí počítače včetně elektronické pošty je směr, kterým se ubírá stále více uživatelů osobních počítačů. Dva programy pro tyto účely - PROCOMM a ODYSSEY jsou srovnávány v časopisu PCW 7/89-str. 190-194.

ků i v klidovém stavu a následném rozjetí umělohmotných zátek rotoru.

Dáme-li plotter do opravy, asi tak po měsíční opravě máme doma zařízení s novým motorem, ale se starými bolestmi. Částečnou prevencí je odpojování plotteru od zdroje hned po skončení tisku.

Těm, kteří mají smysl pro precizní a přesnou práci, doporučuji u příslušného posuvu opatrně vyjmout motorek a opravit rotor. V praxi se ukázalo, že opravený motorek bývá kvalitnější než nový. Avšak důrazně varuji před všemi násilnostmi uvnitř přístroje, protože může dojít k vážným poškozením soukolí. Při návštěvě některé z pravidelných schůzek SHARP klubu můžeme zájemcům poskytnout příslušné rady. Ještě adresa pro písemný styk:

- SHARP klub při 602. ZO Svazarmu, dr. Z. Wintra 8, 160 41 Praha 6.

Z 88 v roce 1989

Jan Zoubek

Sir Clive Sinclair má pověst geniálního konstruktéra a nevalného obchodníka. Snad právě spojení těchto vlastností vzbuzuje sympatie jeho četných příznivců. Je autorem celé řady vynálezů. Za jeho činnost mu byl udělen šlechtický titul. Vešel do historie nejen jako tvůrce ZX80 a ZX Spectra, ale je považován i za vynálezce elektronické kapesní kalkulačky. Produkty "Sinclair made" se staly předmětem zájmu sběratelů. Jméno Sinclair nemůže již ale pro své produkty používat, postoupil je podivnou transakcí, zavánějící středověkem, Alanovi Sugarovi (AMSTRAD) v období finančních obtíží po několika obchodně neúspěšných projektech (počítač QL, elektrické městské vozítko C5). Pro vytvoření svého nového dítko, jak je Z88 často nazýván, založil Sir Clive novou firmu, Cambridge Computer LTD.

Dlouho před zahájením výroby předcházely Z88 nadšené komentáře, vstup na trh nebyl ale právě šťastný. Počítač byl nejprve dodáván jen na základě poštovních objednávek, dodávky přístrojů a příslušenství občas vázly, brzy po zahájení prodeje byla zvýšena cena základního přístroje ve Velké Británii na téměř 300 GBP, s optimálním základním vybavením cca 400 GBP, což zřejmě bylo pro řadu potenciálních kupců již příliš mnoho (dnešní základní cena Z88 po nedávném zlevnění je cca 200 GBP). První série nebyly prosty dětských nemocí, přístroj nezískal pověst zcela spolehlivého výrobku. Po zahájení distribuce bylo třeba upravit ROM. Krom softwaru, zabudovaného v počítači, nebyl k dispozici žádný další, tak jak je to obvyklé u počítačů, neodpovídajících zavedeným operačním systémům. Výrobci softwaru vyčkávají s vývojem, dokud nebude v oběhu dostatečný počet počítačů, zajišťující rentabilní odbyt programu, počítače nejdou na odbyt, neboť k nim není k dispozici software. Jediným, ale dosti neúplným zdrojem informací o počítači byl manuál, dodávaný s přístrojem.

K důležité události v historii Z88 došlo na podzim 1987, kdy byl založen Z88 User's Club a vyšlo první číslo klubového časopisu Z88 EPROM. Fotograf Roy Woodward, zakladatel a předseda klubu, redaktor, nakladatel a vydavatel časopisu a v prvních číslech i autor většiny článků (to vše jako nevýdělečná zájmová činnost!), umožnil majitelům tajemných černých skříněk sdělovat problémy a objevy, ke kterým dospěli. Při četbě časopisu se občas vtírá fascinující pocit, že se zkoumá předmět, vyrobený neznámou civilizací.

Na následujících řádkách se pisatel chce pokusit o stručný souhrn půlročních zkušeností s užíváním Z88. Je jeho příjemnou povinností úvodem sdělit, že považuje Z88 za vynikající přístroj, který by již velmi nerad postrádal.

Jednou z charakteristických částí Z88 je LCD displej rozměru 250 x 30 mm (výrobce EPSON). Při běžné praxi s textem je převážná, centrální část využita pro šest textových řádek. V pravé části je zobrazena "mapa" stránky, na které se právě pracuje (jeden charakter jedním bodem). Na tento způsob vnímání textu jsem si rychle zvykl. Písmena na displeji jsou sice malá, ale stabilní (nekmitají) a velmi dobře čitelná. Ani při delší práci nepocítuji únavu očí. Při nedostatku světla je nutno displej osvětlit. Při postupném ubývání intenzity světla přestane být displej čitelný podstatně dříve, než běžný tisk na papíře.

Z88 je lehce přenosný, rozměru i váhy jednoho dílu pražského telefonního seznamu. Psát na tiché, pružné, matně černé, silikonové klávesnici (další charakteristická část) je příjemný pocit, v jednom článku dokonce přirovnávaný k zážitku sexuálnímu.

Nejcennější je pro mne schopnost operovat s textem (Pipe Dream). Z88 umožňuje pracovat střídavě na řadě rozepsaných dokumentů, např. dopisů a zpráv, jejichž množství a délka jsou limitovány jen velikostí paměti. Velmi snadno lze přecházet z jednoho dokumentu na druhý. Kdykoli je možno práci přerušit a po "zapnutí" (Z88 ve skutečnosti nikdy úplně vypnut není) pokračovat právě tam, kde jsme skončili. Pipe Dream obsahuje i jednoduchý spreadsheet. Snadno lze přejít na další vestavěné funkce, např. kalkulačtor, diář, zvukový signál, nebo BBC Basic.

Pro napojení dalších periférií je počítač vybaven sériovým portem RS 232 (devítikolíkový typ). Paralelním kabelem, který je nabízen jako příslušenství, lze napojit i běžné tiskárny s paralelním portem.

Z88 má tři druhy paměti: RAM, ROM (vnitřní a vnější) a EPROM. Vnější paměti, výrobcem označené jako "cards", mají tvar malých plochých čtvercových krabiček. Do štěrbin v čele přístroje, přístupných po odklopení průhledné plastické zátky, lze zasunout tři takové karty. Prozatím jsou k dispozici maximálně 1/2 MB karty RAM. Na kartách ROM jsou v současné době k mání 2 spelling checkery, database, hry a link. Na karty EPROM (Erasable Programmable Memory) lze velmi jednoduše přehrát cokoli je v RAM paměti počítače, ale vymazat je možné jen celý obsah EPROM najednou, mimo počítač, zdrojem ultrafialového záření. Karty ROM jsou identické s EPROM, mají jen přelepen otvor pro vstup UV záření. Jestliže přestaneme potřebovat nahraný program, otvor možno odkryt a ROM kartu tak změnit v EPROM.

Z88 sice není s jinými počítači kompatibilní, data lze ale převést oběma směry zařízením, označovaným jako "link" či "adaptor". V současné době je k dispozici pro následující typy počítačů: IBM PC, Amstrad PCW, Archimedes, BBC, Macintosh a Atari ST. Prvně jmenované zařízení označuje výrobce jako PC link II a sestává z diskety (podle přání 3.5" nebo 5.25") obsahující příslušný software, z ROM karty a propojovacího kabelu. Po instalaci programu do obou počítačů a jejich propojení kabelem se přenos dat oběma směry ovládá z PC. Ve formě textu lze přenášet i programy psané v BBC Basic. Z hlediska přenosu dat mezi Z88 a PC je zajímavá a existující rozšířená verze Pipe Dreamu pro PC. Stejný typ Word Processoru je pak možno používat při práci s PC i Z88.

CCL přehání možnosti práce Z88 na vnitřní zdroj (4 tužkové baterie). I při použití alkalických se již po řádově hodinové práci objeví varování "BAT LOW", posléze se obraz začne trhat až nakonec zmizí. Po výměně baterií je v tomto případě obsah RAM zachován (baterie je možno měnit při napojení počítače na vnější zdroj, ale vestavěný kondenzátor umožňuje i jejich rychlou výměnu bez takové pomoci). Abych je šetřil, používám vnitřní baterie jen pro napájení paměti (je-li počítač "vypnut"). Při práci užívám vnější síťový zdroj. Doporučuje se též používat jako vnější zdroj pět tužkových Ni-Cd akumulátorů spojených do série. Tento způsob jsem nemohl realizovat, protože není k mání vhodný

držák (ani na čtyři baterie), o potřebném konektoru 2.1 mm nemluvě. Vážnější problémy mohou nastat při transportu počítače. Jsou-li náhodou trvale stisknuty některé klávesy, baterie se mohou zcela vybit, což má za následek ztrátu všeho, co bylo uloženo v RAM (tak mimochodem skončila první verze tohoto článku). V zimě může orosení po náhlé změně teploty být příčinou dočasné poruchy klávesnice, nebo v horším případě, může dojít opět ke ztrátě informací uložených v RAM.

Pro trvalé uložení dat slouží paměti typu EPROM. Jejich relativní cena (přepočtená na jednotku záznamu) je mnohonásobně vyšší, než jakého-

koli jiného srovnatelného media (disk, magnetofonová kazeta).

Manuál dodávaný k Z88 není napsán příliš přehledně a na první přečtení srozumitelně.

Při operaci s textem v rámci "Pipe Dreamu" postrádám obvyklé "Cut and paste".

+ přenosnost
klávesnice
pohotovost
vestěvný software
EPROM - vlastnosti

- rezerva el. energie
odolnost při transportu
EPROM - cena
další software

The Personal Computer Show 1989

Ota Luňák

Na 22 000 čtverečních metrech se ve dnech 27. září až 1. října v Londýnské hale Earls Court konala v pořadí již dvanáctá výstava všeho, co souvisí s osobními počítači. Zúčastnilo se jí 400 firem z celého světa, žádná však z naší republiky.

Největší stánek patřil firmě Atari - 1 100 m², firma Amstrad okupovala 450 m² a nejmenší stánek měl plochu 6 m². Náklady můžete vykalkulovat sami - 155 GBP za čtvereční metr výstavní plochy, případně včetně 178 GBP za policový systém, pokud vystavovatel nepoužil vlastní vybavení. Za zmínku stojí dokonalý systém organizace výstavy. Samozřejmým vybavením organizačních pracovníků byly přenosné radiotelefony. (firma Amstrad je dodává ke svým přenosným počítačům PPC512 zdarma!), které umožňovaly téměř okamžité řešení nastalých problémů. Pro zástupce tisku bylo k dispozici tiskové středisko, kde bylo možné nejen získat nepřeberné množství tiskových materiálů jednotlivých vystavovatelů (i když jsem vybíral jenom materiály, které mě skutečně zajímaly, nakonec jich bylo 25 kg...), ale i uskutečnit konsultace či konference. Samozřejmým bylo i samoobslužné občerstvení zdarma, které přišlo vhod hlavně proto, že zařízení tohoto charakteru ve výstavních prostorách byla neustále obležena mnoha zájemci. V obležení však byly i všechny stánky vystavovatelů, které nabízely snad opravdu všechno, co nějak souvisí s osobními počítači všech výkonostních, cenových a ostatních kategorií. Důležitým organizačním prvkem bylo oddělení zábavného prostoru od prostoru vyhrazeného pouze profesionálům, kam byl omezen přístup osobám mladším 18ti let. Odpadl tak nesmyslný, bezhlavý sběr prospektů adolescentní mládeží, což se proje-

vilo klidnější atmosférou, než jaká panovala v království her.

Dominantou stánků profesionálních firem byly bezesporu počítače řady 486. Ty bylo možné shlédnout u firmy Tulip a v tzv. Singapurské vesničce. Novinkou u firmy AMSTRAD byl levný a dobře vybavený Fax (599 GBP) a levné varianty (bez HD) počítačů řady 2000 s označením PC 1286 a PC 1386. Milou pozorností je automatický podavač řezaného papíru (cut sheet feeder) pro PCW9512 v ceně systému. Tím již PCW9512 představuje efektivní systém pro tvorbu dokumentace. Novým doplňkem Spectra+2 je světelná pistole (action pack), která má oživit uhasínající zájem o tyto počítače. Firma CAS Computers zaujala počítačem řady 386 s cenou pod hranicí 800 GBP (2160 DM). Novým účastníkem výstavy byla firma Hewlett-Packard, která prezentovala řadu tiskáren JET na jednom z největších stánků. Polovinu stánku firmy ATARI zabíraly produkty věnované obchodní činnosti - celkem 132 jednotlivých produktů včetně FoxBase a FoxGraphics. Z vlastních výrobků firmy ATARI bylo možné shlédnout ST Portable, Atari TT Unix, Atari STE a kapesní Portfolio. Neméně zajímavý byl také stánek firmy Commodore, ale ono bylo zajímavé téměř všechno, zvláště pro našince z Československa, vyhládlého po počítačové technice. Pro naše tvůrce programového vybavení je poučné zejména tempo zavádění novinek ve snaze udržet si postavení na trhu. FoxPro, konkurence pro dBase IV s multi-relačními vztahy, presenční grafikou, Quattro Pro s tiskovým výstupem v kvalitě DeskTop, hot-keys provazujícími 32 souborů mezi sebou a mezi presenční grafikou, Paradox 3.0 s vlastním aplikačním jazykem (PAL)... Tempo vývoje je tak závratné, že další PCW Show očekávám s mrazením v zádech a s myšlenkou, je-li vůbec možné tak komplikované produkty dokonale zvládnout v časovém limitu daném praktickým životem.

NEPÁJIVÁ KONTAKTNÍ POLE

Radek Kluger

Již delší dobu jsou ke koupi nepájivá kontaktní pole za 35 Kčs, jejichž výrobcem je 602. ZO Svazarmu. Tato pole jsou vhodná především pro experimentální zapojování jednodušších elektronických obvodů bez potřeby pájení. Rozteče kontaktních otvorů samozřejmě umožňují zapojovat i různé integrované obvody. Výhody tohoto způsobu propojování součástek jistě ocení každý, protože vytvoření požadovaného zapojení je velmi rychlé, bez poškození součástek a lze je snadno modifikovat.

Z vlastní zkušenosti mohu říci, že kontaktní pole fungují spolehlivě i po dlouhodobém používání. Přes popsané výhody má toto pole i určité nevýhody, jimiž jsou poměrně malé rozměry a dost vysoká

cena (i když ve srovnání např. s cenami objímek pro IO na našem trhu se kontaktní pole rozhodně nejeví jako předražené).

V současné době se připravují na trh vylepšená kontaktní pole, která mají větší rozměry, abecedně-číselné označení řad kontaktů a oddělené napájecí lišty. To umožňuje sestavit požadovaný tvar a velikost z několika základních polí. S výhodou lze přitom použít již připravené otvory k připevnění na podkladovou desku. Zde bych se však přimlouval za vhodnější umístění zmíněných otvorů, protože u stávajících se mohou po zašroubování samořezných šroubů zkratovat některé kontaktní plíšky.

V každém případě se však lze na nová kontaktní pole těšit a doufat, že budou na trh uvedena co nejdříve a hlavně v přijatelné ceně.

Postavte si s námi diskový řadič

/6/

Úvodem několik drobných oprav. V prvním dílu seriálu, v čísle 5/89, je ve schematu na str. 8 vývod č. 12 konektoru pro spojení s deskou systému označen jako -5V. Je to pozůstatek předchozích verzí. Na desce plošných spojů je zde zapojeno +5V, protože -5V se vůbec v zapojení nepoužívá. Odpor R13 je nadbytečný, protože je zapojen paralelně k R3. Proto R13 neosazujte - zhoršuje činnost zapojení! Ve druhém pokračování tohoto seriálu, v čísle 6/89, jsou na str. 10 chybně uvedeny kapacity disku pro formát Betadisku. Hodnoty 180, 360 a 720KB jsou chybné (jsou to kapacity pro IBM PC). Správné hodnoty měly být 160, 320 a 640KB. Dále jsou v tomtéž čísle v tabulce Basicových příkazů uvedeny příkazy pro přepínání počtu stop "4" a "8". Správné má být "40" a "80", tak jak je to uvedeno v příkladech napravo od tabulky. K chybě došlo při úpravě tabulky. Pokud zadáte zmíněné příkazy bez té nuly, bude systém hlásit "ERROR". U příkazu "*" pro přepínání disketových mechanik je uvedena za označením disku uvedena dvojtečka, tak jak uvádí výrobce, ale ve skutečnosti systém již nekontroluje zda za písmenem ještě něco je. Je tedy možné si ušetřit zbytečné shiftování. V tomtéž čísle vypadl chybový kód číslo 13 = chyba verifikace. Ve schematu desky systému 5.03 na str. 11 je vlevo dole mylně označen výstup invertoru IC1 číslem 6. Správné je to vývod 4. Vývod číslo 12 konektoru pro spojení s deskou řadiče je označen NC. Ve skutečnosti je zde +5V (viz výše). Poslední chyba o které zatím vím je v tomtéž schematu, kde jsem při překreslování mylně označil vývod číslo 27 EPROM jako NC, tedy nezapojený. Správné má být připojen na +5V, tedy spojen se sousedním vývodem 28. V obrazci plošných spojů tato chyba není. Je zajímavé, že díky této chybě zůstal na několika vzorcích řadičů mých přátel skutečně příslušný vývod nezapojený, přičemž systém většinou pracoval a začal stávkovat až začátkem letošního roku. Možná se zde projevila letošní zvýšená sluneční aktivita.

Protože jsem posledních pět čísel Mikrobáze dělal vlastně úplně sám, nemohu se vymlouvat, že by zmíněné tiskové chyby způsobil někdo jiný. Ostatně, těch chyb se vetřelo do oněch čísel víc. Za všechny tiskové a jiné chyby se velice omlouvám a doufám, že nikomu nezpůsobily vážnější potíže.

V jednom pokračování jsem slíbil, že příležitostně uvedu jednoduchý způsob jak přesunout po natažení začátek Basicu zpět na 23755, čímž lze eliminovat jeden z nedostatků Betadisku. Když jsem to nechal přečíst některým svým přátelům, kteří si podle mého návodu už Betadisk postavili, byl o tuto drobnost velký zájem. Možná, že i vás něco takového zajímá. Tak tedy - program v assembleru vypadá takhle:

```
LD HL,#5CCB      ;23755=nová adresa začátku Basicu
EX DE,HL        ;Přenesení do DE
LD HL,(#5C53)   ;PROG=momentální začátek Basicu
JP #19E5        ;RECLAIM-1
```

přesune začátek Basicu, ale také příslušným způsobem upraví všechny systémové proměnné kterých se to týká. Dalšího komentáře netřeba. Zbývá už jen to šikovně přidat do Basicového programu.

Někdo se možná pozastaví nad tím, proč nedávám #5CCB rovnou do registrů DE. To právě souvisí s tím vestavěním do Basicového řádku. Důvod je ten, že kód pro LD DE,nn je #11. Tato hodnota odpovídá řídicímu kódu pro nastavení barvy podkladu při listingu Basicu. Následující hodnota #CB by způsobila, že by od tohoto místa byl listing zablokovaný. To by ztěžovalo editaci Basicového programu.

Demonstrační program vypadá takto:

```
1 REM *****
2 REM * Program pro nastavení začátku Basicu *
3 REM *   na 23755 za chodu programu.   *
4 REM *           (c)1987 D. Meca           *
5 REM *           Jihlavská 76, Praha 4     *
6 REM *****
7 REM
10 REM Radky 30-80 uloží kód.
20 REM
30 LET c$="033,203,092,235,042,083,092,195,229,
    025"
40 RESTORE 120: REM Sem se zadá číslo radku,
    do kterého bude uložen kód.
50 LET a=(PEEK 23639+256*PEEK 23640+7): REM
    Adresa pro uložení strojového kódu.
60 FOR n=0 TO 9
70 POKE a+n,VAL c$(1+4*n TO 3+4*n)
80 NEXT n
90 GO SUB 200 : REM Tiskne začátek Basicu.
100 REM Radky 120-140 provedou přesun začátku.
110 REM
120 DATA "1234567890": REM Zde bude kód.
130 RESTORE 120: REM V radku 120 je uložen kód.
140 RANDOMIZE USR
    VAL "PEEK 23639+256*PEEK 23640+7"
    : REM Spuštění strojového kódu.
150 REM Hotovo!
160 REM
170 GO SUB 200 : Tiskne začátek Basicu.
180 STOP
200 REM Podprogram pro tisk začátku Basicu.
210 PRINT "Začátek Basicu = ";
220 PRINT PEEK 23635+256*PEEK 23636
230 RETURN
```

Program využívá té skutečnosti, že po provedení příkazu RESTORE ukládá systém na systémovou proměnnou DATADD adresu o jedničku nižší než je začátek řádku uvedeného v argumentu.

Po spuštění se vypíše napřed původní začátek Basicu a po provedení strojové rutiny již nový začátek, tj. 23755. Řádek 120 se v průběhu první části programu přepíše takto:

```
120 DATA "! THEN \ FOR *S\NOT RESTORE ?"
```


DATA přímo z klávesnice. Bylo by to možné s výjimkou toho otazníku na konci. Tam je totiž kód #19, který z klávesnice zadat přímo nejde. Pro hráčičky, kteří by přesto chtěli zadat kód z klávesnice, jsem si vymyslel takovouhle modifikaci:

```
LD HL,#5CCB ;23755=nová adresa začátku Basicu
EX DE,HL ;Přenesení do DE
LD HL,#39E5 ;Jako #19E5, ale nastavený bit 5
RES 5,H ;V HL je teď #19E5
PUSH HL ;Na STACK
LD HL,(#5C53) ;PROG=momentální začátek Basicu
RET ;Nepřímý skok na #19E5
```

Príslušný Basicový řádek pak vypadá takhle:

```
120 DATA " ! THEN \ FOR ! RESTORE 9 THEN AT RESTORE
*S\<>"
```

To je sice kód o čtyři bajty delší, ale je to vyváženo možností přímého zadání z klávesnice. Postup zadávání je takový:

```
1 2 0 DATA " ! THEN FOR kurzor vlevo \ kurzor
vpravo ! RESTORE 9 THEN AT RESTORE * S \ <> "
ENTER
```

Pozor, "<>" je jeden znak (kód 201, tj. RET)!

Podrobněji se tímto kratičkým programkem zabývá hlavně proto, že ho asi použijí hlavně ti, kteří si nejsou v programování tak jisti. Snad to vše vypadá pro začátečníka dost složité, ale vlastní použití je jednoduché. Číslo řádku 120, získaného z výše uvedeného ukázkového programu, je možno změnit naprosto libovolně. Podmínkou je, aby na něj ukazoval příkaz RESTORE, který je před příkazem RANDOMIZEUSR... (zde řádky 130 a 140). Druhá podmínka spočívá v nutnosti umístit příkaz DATA s příslušným kódem hned na začátek řádku. Jinak je možno příkazy z řádků 120-140 spojit dvojtečkami do jednoho řádku, nebo je použít samostatně. Je možné je zavolat hned při autostartu programu nahrátého z disku, nebo i později. Jen je třeba se přesvědčit, že příkaz RESTORE nekoliduje s jinými daty programu a případně zařadit ještě jeden příkaz RESTORE, který na ně bude ukazovat.

Do oblasti hrátek se systémem patří také možnost umístit příkaz DATA do řádku 9999 a tento řádek z monitoru přečíslovat tak, aby první bajt čísla měl hodnotu #40. Takový řádek pak není v listingu vypisován, není v Basicu spouštěn, ale finta s nalezením jeho adresy na něj stále funguje. stačí dát RESTORE 10000. Program s takto přečíslovaným řádkem nelze načíst příkazem MERGE. Ale to už jsem hodně odbočil od tématu.

Programěk může být použit také k odstavení ZX Interface 1.

V tomto místě jsem musel zasáhnout do svého vlastního textu a nahradit první část popisu ovládání řadičů Western Digital několika poznámkami ke stavbě a oživení Betadisku. Důvodem je to, že vzhledem k pokročilému času má řada čtenářů už celý řadič postavený. Zajímají se tedy o možnost naprogramování EPROM, o zkušenosti s oživováním a také o případné programy pro práci s Betadiskem. Protože (jak se můžete dočíst na jiném místě v tomto časopise) Mikrobáze tímto číslem končí, přičemž není zatím jasné kdy a kde budu moci v tomto seriálu pokračovat, vidím toto jako jediný způsob, jak alespoň nenechat na holičkách ty, kteří si podle mého návodu řadič staví.

Doposavad jsem nevyhodil nikoho z těch několika desítek lidí, kteří mě navštívili se žádostí o radu. Také jsem zdarma naprogramoval řadu EPROMek. Protože to je však dost velká ztráta času a zátěž pro mou rodinu, budu nucen další podobné služby poskytovat za mírnou úplatou na základě povolení NV.

Pochopitelně hledám a budu hledat vydavatele pro zbytek seriálu a nejen pro něj.

Tak tedy k vlastní stavbě. Nepodařilo se zatím zajistit výrobce oboustranných prokovených plošných spojů, který by je za rozumný peníz zájemcům dodával. Proto si bude každý muset poradit sám s pomocí předloh z Mikrobáze. Pokud by někdo výrobce našel, dejte mi prosím vědět. Použití plošných spojů bez prokovených děr je velké dobrodružství a hrozná práce. I to jsem však viděl.

Největší procento chyb bylo zaviněno nekvalitními plošnými spoji, proto radím důkladnou kontrolu. Zvláště zrádné jsou vlasové trhlinky. Slušné však dovede potrápiti i nedokonale prokovený otvor. Vyplatí se také zkontrolovat diody. Měl jsem už v ruce diody s opačně označenou katodou.

Zkontrolujte si také důkladně, zda nejsou na plošném spoji systému prohozené motivy pro stranu spojů a stranu součástek! V takovém případě by nesouhlasily vývody systémového konektoru a došlo by k poškození počítače.

Pokud jsou plošné spoje i součástky v pořádku a nedojde k chybě při osazování součástek, mělo by vše fungovat na první zapojení. Před připojením k počítači doporučuji připojit k napájení samotnou desku řadiče a překontrolovat napětí na konektoru pro počítač. Veškeré připojování a odpojování se samozřejmě musí dělat při vypnutém napájení.

Desku systému lze s počítačem zkoušet samostatně, tj. bez desky řadiče. Napájecí kabel ze Spectra se připojuje do desky Betadisku - Spectrum je napájeno přes konektor sběrnice. Bylo by samozřejmě možné naopak napájet Betadisk ze Spectra, ale při nasunutém interfejsu se už do Spectra napájecí konektor nevejde. Stabilizátor +5V musí mít chladič - i potom dost hřeje. Po zapnutí proběhne inicializace Spectra a pak se buď rovnou zahlásí TR-DOS, nebo se hlásí Spectrum, které však musí být normálně provozuschopné. Když se ohlásilo Spectrum, pak se TR-DOS zahlásí až po RANDOMIZEUSR 15616. Předpokládá se, že dvoupólový prepínač je přítom ve střední poloze, případně vůbec není připojen. Nahoře je nápis:

* TR-DOS Ver. 5.03 *
1986 Technology Research Ltd.
(U.K.)
BETA 128

Pokud je připojen ZX Interface 1, bude pod tím ještě napsáno "Interface 1 attached", nedoporučuji však při prvních pokusech mít cokoli ke Spectru připojeno. V každém případě se má vlevo dole objevit:

A> a kurzor.

Pokud je to tak, funguje stránkování EPROM dobře. TR-DOS totiž každý znak vypisuje přes ROM Spectra. Můžeme vyzkoušet příkazy "40" a "80", na které systém odpoví "O.K.". Na ostatní příkazy reaguje hlášením "No disk". Pokud se Betadisk takto nechová, doporučuji vše znovu řádně překontrolovat. V praxi se ukázalo, že někdy pomohlo zvýšit hodnoty odporů, které vedou ze spojených anod diod k +5V, z původních 4k7 na 15 až 22k. Bude vůbec lepší rovnou tyto hodnoty osadit. Dále celé věci prospěje zapojení takového odporu i do anod D2 a D3 (vývod 3 na IC1 v desce systému). Někdy se osvědčuje zapojit kondenzátor asi 68pF z výstupu číslo 11 IC3 na zem. Bez něj systém někdy nevysvětlitelně "padá". Může také pomoci zablokování napájení EPROM přímo u patice kondenzátorem M1. Zapojení pak už musí fungovat!

Při resetu na počítači by se mělo hlásit Spectrum. Při resetu prepínačem na řadiči by se měl hlásit TR-DOS. Z DOSu se dostaneme na Spectrum příkazem RETURN. Nyní můžeme zkusit RANDOMIZEUSR 15360. Ve střední poloze prepínače se i tak musí hlásit řadič. Pokud zůstaly zachovány systémové

proměnné TR-DOSu, nebude se již znovu vypisovat celé hlášení.

Teď už můžeme (pochopitelně při vypnutém napájení) připojit desku řadiče. Mechaniku zatím nepřipojujeme. Na chování TR-DOSu by se nemělo nic změnit. Nakonec (opět při vypnutém napájení) připojíme mechaniku. Ta bu měla být přepnuta jako D0, tedy drive A. Zapojení konektoru je následující:

>	1 Zem	2 NC
	3 Zem	4 NC
	5 Zem	6 /DRVD
	7 Zem	8 /INDEX
	9 Zem	10 /DRVA
	11 Zem	12 /DRVB
	13 Zem	14 /DRVC
	15 Zem	16 /HL (zde /MOTOR ON)
	17 Zem	18 DIR
	19 Zem	20 /STEP
	21 Zem	22 /WRD
	23 Zem	24 /WRG
	25 Zem	26 /TRO
	27 Zem	28 /WRT PROT
	29 Zem	30 /RD
	31 Zem	32 /SIDE1
	33 Zem	34 NC

Lomítko před označením značí negovaný signál. Na většině konektorů bývá vývod číslo 1 označen šípkou. Na tu stranu patří na plochem vodiči strana označená barevně. K Betadisku lze bez problému připojit mechaniky 3", 3½" a 5½" pro Double Density, případně 8" pro Single Density. Bezproblémovost připojení umožňuje především to, že není požadován signál /READY, který řada novějších mechanik (hlavně 3½") vůbec nemá. Systém si zatočí diskem a testuje přítomnost indexových impulzů.

Pokud není k dispozici na Betě formátovaný disk, musíme nejdříve provést formátování. Nejdříve zadáme příkaz "40", případně "80" (pokud to mechanika umožňuje). Potom už zadáme FORMAT "název" (a ENTER). Pokud má být disketa formátována jednostranně, musí být první písmeno názvu \$. Smaže se obrazovka a musí se roztočit mechanika. Po chvíli bude slyšet krokování. Po skončení formátování se vlevo dole objeví údaj o počtu bezvadných sektorů, pak lomítko a údaj o možném počtu sektorů. Pokud tato dvě čísla nejsou stejná, zkusíme formátovat znova. Když ani to nepomůže, raději disketu vyřadíme. TR-DOS totiž neoznačuje vadné sektory a neumí se jim pak vyhýbat.

Pokud deska řadiče nefunguje bezchybně, může tomu být na vině osazení odporu R13, který je paralelně k R3 - viz výše. Vadí to hlavně u některých sérií řadičů Siemens. Kromě toho se ukázalo rozumné zvýšit hodnoty odporů R2+R6 z 5k6 na 10k.

Při bližším prozkoumání zapojení lze zjistit několik drobných prohřešků. Například výstup 2 invertoru IO3 by měl být přes odpor připojen na +5V, protože jde o invertor s otevřeným kolektorem. Nepoužité vstupy všech IO by měly být připojeny správně k +5V přes odpory a ne přímo. Měl by

asi být blokován vývod č. 40 řadiče, který je připojen na +12V. Těchto posledních pár drobností by ale nemělo mít vliv na bezvadnou funkci řadiče.

Ne každému se podaří sehnat vhodný dvoupolový přepínač se třema polohami. Je možné použít jednoduchý dvoupolohový přepínač mezi diodou D6, zapojenou v Resetu a vývody /CLEAR a /PRESET klopného obvodu D2 v IC5. V jedné jeho poloze se po resetu ohlásí Spectrum, ve druhé Betadisk. Pokud je resetovací tlačítko na počítači, nemuselo by v tom případě už být na Betadisku. Potom už stačí jednoduchým spínačem uzemňovat anodu D8, čímž se vyřadí možnost volání na adresách 15360 a 15363.

Úpravu volání Betadisku pomocí R7, R8 a D8 nelze aplikovat při použití LECROM, protože spolu kolidují.

Pokud Betadisk odmítá fugovat v kombinaci s některým interfejsem, přesvědčte se, zda v tomto interfejsu není použito dekodování adres pomocí diod. Pokud tomu tak je, pomůže nahrazení diod hradlem. To se týká například rozšířeného UR4!

A ještě jedno upozornění. Při pokusu o čtení diskety neformátované, případně formátované jinde než v Betadisku, objeví se po delší odmlce hlášení o chybě ve stopě 0, sektoru 9. Tam totiž má TR-DOS svoji identifikaci. Při pokusu o čtení z diskety o větší délce sektoru než 256B dojde ke zhroucení systému. Je to tím, že si systém vytvoří buffer pro načtení jednoho sektoru délky 256B, přičemž čtecí rutina (jak jsem již vysvětlil dříve) čte sektor libovolné délky. Jak tento problém odstranit je nasnadě - stačí aby si systém napřed v adresových údajích zjistil délku sektoru a v případě delšího sektoru raději nečetl. Je to jedna z chystaných úprav pro novou verzi DOSu.

Přestože jsem už přesáhl původně vyhrazenou délku tohoto pokračování, řada důležitých věcí se sem už nevešla. V dalších chystaných kapitolách by měly být uvedeny průběhy signálů v důležitých bodech zapojení a jednoduchý program pro otestování některých částí řadiče. Dále je připraven úplný popis ovládání řadičů Western Digital, popis příkazů, analýza a využití stavových bajtů v programech. Dále měl následovat popis zajímavých partií nového CP/M od Jirky Lamače, návod na jednoduchý hardwerový tester mechanik, program pro testování mechanik pomocí řadiče Betadisku bez přídavného hardwéru, popis několika užitečných programů pro práci s Betadiskem a později i popis nového upraveného systému, navenek plně kompatibilního s TR-DOSem. Je připraven i popis úpravy Didaktiku Gamma pro spolupráci s Betadiskem a s CP/M. A ještě celá řada zajímavostí. Jen majitelům ostatních typů počítačů se musím omluvit, asi se mi už samotnému nepodaří dát dohromady slíbené aplikace pro jiné typy počítačů než ZX Spectrum a kompatibilní. Pokud by se tomuto problému někdo chtěl věnovat, napište mi domů. Snad se podaří časem vše někde publikovat. Zatím tedy nashledanou. Sledujte časopisy, snad se časem někde ozvu. A vy, tvořiví přátelé Betadisku a CP/M vůbec - ozvěte se!

Daniel Meca

NOVÁ SLUŽBA ATARISTŮM

Atari klub při DK ROH v Rožnově pod Radhoštěm přichází s novým iniciativním nápadem. Jedná se o disketový klub FLOP. Disketový klub FLOP sdružuje všechny zájemce o práci s disketovou jednotkou na osmibitových počítačích ATARI. Všem svým členům umožňuje výměnu zkušeností, užívání informačních bází, přístup k software typu Public Domain a výměnu informací o malé domácí výpočetní technice.

Po zaplacení manipulačního poplatku a klubového

příspěvku dostává každý člen 6x do roka příslušné informace nahrané na disketě, kterou si sám dodá. Přitom může i sám přispět k doplnění informací.

Bližší informace získá na adrese:

ATARI klub
při DK ROH
"FLOP"

756 61 Rožnov pod Radhoštěm

(Meca)

O KOHOUTKOVI A SLEPIČCE aneb JAK VZNIKALA MIKROBÁZE ROČNÍK 1989

Celkem jsme při výrobě prvních sedmi čísel ročníku 1989 spolupracovali s osmi tiskárnami. Veškerý styk s těmito tiskárnami (vyjednávání, předávání materiálů, náhledy, urgence, organizace přepravy, přejímka atd.) byl mojí starostí. K tomu úkoly spojené s vedením střediska (objednávky, faktury, smlouvy, výplaty, honoráře), a po nocích vlastní redakční práce. Ještě že alespoň čísla 1+5 sestavoval kolega Zajiček.

Obálka se tiskla v tiskárně A. K tisku však byly zapotřebí barevné výtažky, které zajišťovala tiskárna B. Protože tiskárna A nemá k dispozici vhodné rastry, musely se černobílé fotografie připravit k tisku v tiskárně C. Textová část se pro číslo 1/89 tiskla v tiskárně D. Zde během tisku došlo k poruše stroje a pak se musela shánět tisková barva. Text čísla 2/89 se tiskl v tiskárně E, která ale nebyla zařízena na tak velký náklad a navíc měla potíže s nekvalitním papírem. Fotografování stránkových předloh však prováděla tiskárna C. Vlastní tisk tohoto čísla trval tak skoro sedm týdnů čisté práce tiskárny.

Proto se čísla 3 a 4/89 tiskla v tiskárně F na ofsetové rotačce. Stránkové předlohy byly do tiskárny F dodány prostřednictvím tiskárny A, jenže se během transportu někde ztratilo několik stránek. Museli jsme tedy nahonem zjistit co se vlastně ztratilo, na laserové tiskárně znovu vytisknout články a rychle slepit nové předlohy. Tak se nakonec stalo, že se v čísle 3/89 opakují dvakrát některé zprávy. Mezitím se ukázalo, že předlohy fotografií na obálku z tiskárny C nevyhovují kvalitou a tak od čísla 4/89 připravovala tyto předlohy tiskárna G. Protože tiskárna F odmítla další tisk, bylo nutno nahonem dohodnout tisk čísla 5/89 v tiskárně H.

Předloha čísla 5/89 byla poslední na které se podílel kolega Zajiček, takže další čísla jsem navíc ještě sám sestavoval a musel jsem si zajistit také vlastní tisk na laserové tiskárně. Na té se však už nedalo tisknout ze ZX Spectra, na kterém jsem dosud Mikrobázi připravoval (a to ještě na svém vlastním). Po nekonečných urgencích jsem konečně začátkem září dostal zapůjčené přenosné PC Sanyo s LCD displayem. Přesto jsem čísla 6 a 7/89 předal do tisku koncem září. Během toho září jsem se naučil pracovat s Text 602, vytiskl jsem předlohy na laserové tiskárně a nechal nalepit předlohy. Nakonec se mi přece jen podařilo domluvit tisk

v tiskárně F, ovšem fotografie stránkových předloh musela zajistit tiskárna A. Při té příležitosti se někde zatoulaly předlohy některých obrázků, což jsem se dozvěděl až před vánocemi, přestože obě čísla byla předána do tisku koncem září. V období od poloviny listopadu do poloviny prosince nebylo možné nikoho ze zodpovědných pracovníků tiskáren sehnat, protože byli stále na schůzích. Navíc mne v té době začaly trápit průdušky, takže jsem nebyl dost mobilní. Když jsem konečně všechny obrázky znovu posháněl a osobně dohlédl na jejich přefotografování, bylo konečně možné zkompletovat předlohy, podle kterých pak byla provedena montáž v tiskárně. To jsem dělal na štědrý den. Po svátcích, přesněji řečeno v půlce ledna byla konečně obě čísla vytištěna a navezena do Tomosu (kde se sešivala všechna dosavadní čísla) k sešití. Protože tam však došla mimo program, poležela si tam přes měsíc. Nakonec byla expedována začátkem března.

Osud zbývajících čísel ročníku 89 je touto dobou stále nejasný, přestože čísla 8 a 9/89 byla do výroby připravena už v listopadu a číslo 10/89 v lednu. Dokonce i odraznou předlohu jsem slepoval sám, aby se vše urychlilo. Je pravda, že číslo 10/89 jsem pak musel celé podstatně přepracovat. Ne snad proto, že by obsah byl pro současnou politickou situaci nepřijatelný, ale protože jsem se mezitím dozvěděl, že číslem 10/89 má Mikrobáze skončit. Snažil jsem se tedy umístit hlavně aktuálnější příspěvky a nezačínat nové seriály. Také jsem se snažil umístit články na které už byla předem s autory podepsaná smlouva. Řadu článků pro číslo 10 dodali autoři s velkým zpožděním a menším rozsahem, než bylo domluveno (Céčko), některé nedodali vůbec (Teletext). Revoluční dění vyhnalo řadu lidí od počítačů do ulic a na nějaké psaní nebylo ani pomyslení. Dva studenti fakulty žurnalistiky, kteří měli zájem o spolupráci jako redaktori, měli po 20. listopadu přinést první reportáž, ale už se neobjevili. Navíc byl v této době problém také s poštou. Řada příspěvků i redakční korespondence nedošla vůbec, nebo došla také s neuvěřitelným dvouměsíčním zpožděním.

Situace v tiskárnách je v současné době značně komplikovaná. Vychází řada nových časopisů a novin. Není papír. Protože každá tiskárna potřebuje ke svému provozu řadu věcí dovézt ze západu, hledá si přednostně zakázky s devizovou spoluúčastí. To my rozhodně nejsme. Takže zpoždění narůstá...

kou číslo 10 až v červnu, netušil jsem, že ještě koncem května nebudou vytištěna čísla 8 a 9. Po tomto zjištění jsem znova předělal toto číslo, kde jsem ještě doplnil další zkušenosti se stavbou Betadisku a přidal toto post scriptum.

Slavnostně slibuji, že na tomto posledním čísle Mikrobáze nebudu už nic měnit. Doufám, že alespoň po tisku volebních lístků a plakátů, jízdních rádků, ceníků a dalších přednostních tiskovin, se konečně některá tiskárna vrhne na Mikrobázi a pomůže nám tak od velké starosti a dluhu. Děkuji vám všem za trpělivost.

Daniel Meca

RSX – rozšíření Basicu Amstrad/Schneider

Jan Teuschel

V operačním systému počítače AMSTRAD/Schneider CPC 464 existuje velmi zajímavá možnost definice vlastních příkazů jazyka Basic. Tyto příkazy jsou však použitelné i v jazyce Pascal HP4 (viz manuál k němu). Jde o tzv. RSX příkazy (Resident System Extension). Jejich použití vypadá takto:

```
|JMENO [arg1, arg2, ..., argn]
```

Počet uživatelem definovaných RSX příkazů je neomezený a je dán pouze kapacitou paměti. Takto si vlastně programátor může vytvořit svůj vlastní programovací jazyk pro Basic interpret. V tomto článku bych se chtěl zaměřit především na způsob návrhů RSX příkazů.

Nejdříve k argumentům. V zásadě jde o parametry volané hodnotou, tj. hodnota se předává směrem příkaz --> obslužný podprogram. Pomocí ukazatele na proměnnou - funkce @proměnná (funkce poskytuje adresu již deklarované proměnné) lze provádět datový přenos směrem opačným. Počet argumentů příkazu může být maximálně 32. Každý argument představuje 16ti bitové číslo. Reálná čísla se zaokrouhlují, čísla větší než $2^{16}-1$ způsobí chybu. Do paměti se argumenty ukládají následujícím způsobem: v registru A je počet uvedených argumentů za příkazem a registr IX je ukazatelem v zásobníku argumentů. Poslední argument stojí na adrese IX, předposlední na adrese IX+2, první na adrese IX+2*(A-1). Způsob ukládání argumentů nejlépe osvětlí příklad:

```
|JMENO,&12,&3456,&78,&ABCD
```

Obsah registru A...4

V IX adresa určená operačním systémem, argumenty leží v paměti takto:

```
IX+7 &00  
IX+6 &12  
IX+5 &34  
IX+4 &56  
IX+3 &00  
IX+2 &78  
IX+1 &AB  
IX &CD
```

Výstup hodnoty z příkazu je možný, jak už uvedeno, pomocí funkce @proměnná, čili:

```
|JMENO,@proměnná
```

A...1

```
IX+1 High byte  
IX Low byte ] adresa uložení proměnné
```

Výsledky naší procedury mohou tedy ukládat do proměnné, na níž mám uložený ukazatel na adresách IX+1, IX.

Nyní, jak se vlastně vytváří takový RSX příkaz? Programátor pro něj musí vytvořit dvě tabulky. Označím si je např. RSX a SYSTEM.

Tabulka RSX má tvar:

```
DEFW TABUL  
JP PRIKAZY
```

Na adrese TABUL je uloženo jméno příkazu, tak jak stojí za |.

Tabulka TABUL má strukturu:

```
TABUL: DEFN "JMEN"
```

```
DEFB "O"+&80 ; - označuje počítači poslední znak jména
```

```
DEFB 0 ; konec tabulky
```

Na adrese PRIKAZY začíná samotný obslužný program zakončený RET. Obslužný program může být libovolně složitý, nejsou kladena žádná omezení ani na jeho délku, ani na počet podprogramů, v podstatě se počítač chová tak, jako kdyby vykonával libovolný podprogram. Příkaz nijak neovlivňuje procesy přerušování a časování v počítači. Na konci obslužného programu se musí vyskytovat RET (buď explicitně nebo implicitně skokem JP &adr, kde adr je adresa programu z Firmware Jumpblock (to jsou podprogramy vždy zakončené RET - viz příklady dále.).

Operačnímu systému existenci nového příkazu oznámíme takto:

```
LD BC,RSX  
LD HL,SYSTEM  
JP &BCD1
```

Tuto krátkou sekvenci příkazů uložím do paměti od adresy adr, po zavedení dat okolo příkazu, pro něž jsou tyto tabulky konstruovány, z Basic zavolám CALL &adr. Od této chvíle je k dispozici příkaz |JMENO [arg1,arg2,...,argn].

Pokud je nově generovaných příkazů více, musím pro každý příkaz vytvořit novou tabulku a každý příkaz zvlášť oznámit operačnímu systému.

Tabulka SYSTEM má tvar:

```
SYSTEM: DEFS 4
```

a slouží jako datová paměť pro jádro operačního systému. Je nutné zajistit aby během programu nebyly tyto 4 byte přepsány.

Příklady:

FUNKCE / čeká na stisk jakékoli klávesy a po něm opět pokračuje hlavní program.

JMÉNO / |WKEY

```
ORG &A000
```

```
LD BC,RSX
```

```
LD HL,SYSTEM
```

```
JP &BCD1
```

} tento blok příkazů si označím jako OZN

; zde začíná vlastní deklarace

```
RSX: DEFW TABUL  
JP PRIKAZY  
TABUL: DEFM "WKE"  
DEFB "Y"+&80  
DEFB 0  
SYSTEM: DEFS 4  
PRIKAZY: JP &BB18
```

Tento příklad byl více než jednoduchý, k základnímu pochopení struktury však postačí.

Další příklad:

opět jednoduchý RSX, bez argumentu.

FUNKCE / přepne počítač na nejvyšší technicky možnou záznamovou rychlost dat - 3600 Bd
 JMENO / FAST
 ; na tomto místě bude uložen blok příkazů OZN
 ; vlastní definice příkazu
 RSX: DEFW TABUL
 JP PRIKAZY
 TABUL: DEFM "FAS"
 DEFB "T"+&80
 DEFB 0
 SYSTEM: DEFS 4
 PRIKAZY: LD HL,93
 LD A,10
 JP &BC68

;sekvence příkazů označená OZN
 ;vlastní definice
 RSX: DEFW TABUL
 JP PRIKAZY
 TABUL: DEFM "RESE"
 DEFB "T"+&80
 DEFB 0
 SYSTEM: DEFS 4
 PRIKAZY: CP 1 ; počet argumentů = 1 ?
 RET NZ ; pokud ne, návrat
 CALL &BC02
 CALL &BB4E
 CALL &BBBA
 LD A,(IX) ; argument do registru A
 JP &BC0E

Posledním příkladem je příkaz, který má jeden argument:

FUNKCE / nastaví všechny INK,PEN,PAPER,BORDER jak v textovém tak i grafickém okně do stavu, jako po zapnutí počítače, přepne do modu určeného argumentem mode, smaže obrazovku, kurzor umístí do levého horního rohu.
 JMENO / RESET,mode
 mode - celočíselný výraz s hodnotou 0,1,2 (viz čísla mode v Basicu)

Vhodné místo pro trvalé uložení RSX kódu do paměti je od adresy &A000. K dispozici pro něj mám rozsah adres &A000-&AB00, nad adresou &AB00 jsou již systémové proměnné.

Použití a definice příkazů RSX je, jak je vidno, velmi jednoduché. Uvedené příklady mají spíše výkladový charakter, jejich praktický význam je však značný. Záměrně není pro ně uveden žádný tzv. Basic loader pro kód těchto tří nových příkazů, neboť článek je určen pro programátory, nikoliv pro kopírovače programů. *

DOPISY - OHLASY - DOPISY - OHLASY - DOPISY - OHLASY - DOPISY - OHLASY

Vážená redakce,

Sokolov dne 1.9.1989

rozhodl jsem se podělit se s Vámi o svoji odyseu, týkající se realizace mého úmyslu koupit si ke svému počítači Spectrum + tiskárnu BT 100 s příslušenstvím (interface, kabely, programy). Domnívám se, že jejím otištěním by se další zájemci mohli lépe orientovat a uspořit tak spoustu peněz a času.

O koupi tiskárny jsem začal uvažovat vloni na podzim. V té době již byly v obchodech přístroje ST210T (magnetofon s tiskárnou), ovšem bez kabelů a interface. Programy v písemné podobě měly být součástí přístroje. Začal jsem tudíž v obchodech a u výrobce Tesly Přelouč po kabelech a interface pátrat. V obchodech nic nevěděli, v Tesle jsem zjistil, že kabely nejsou kvůli konektorům z Jihlavy a interface že má dodávat do obchodů Tesla Kolín, kde mi to bylo potvrzeno.

Koupě tiskárny a interface se ovšem změnila ve shánění s nechutnými scénami v obchodech. Nakonec jsem věci v důsledku protekce sehnal. Sehnal jsem koncovky vodiče a kabely si udělal a zapojil je do interface. Z dokumentace tiskárny jsem však s překvapením zjistil, že je zde písemně jen program k tisku obrazovky. Nařadil jsem jej do počítače, což s kontrolou trvalo několik hodin. Co by bylo ušetřené práce, kdyby to bylo na kazetě.

Další překvapení (samozřejmě nemilé, milá neexistovala) bylo po tisku obrazovky. Ačkoliv tiskárna je pro tisk na celou šířku A4, tisk byl jen na polovinu této šířky. U Tesly Přelouč jsem se dotázal na programy pro tisk textu, pro tisk listingu a obrazovky, vše na celou šířku papíru. Bylo mi k mé radosti sděleno, abych jim poslal kazetu, že mi programy nahrají.

Šest týdnů trvalo, než jsem žádané obdržel. Ovšem předtím jsem ale musel třikrát telefonicky urgovat a jednou psát až řediteli. První kazeta se totiž ztratila, byla mi zaslána obyčejnou poštou. Na obdržené kazetě byl program USR BT 100 ve strojním kódu o délce 1050 B bez jakýchkoliv pokynů. To znamenalo shánět programy dál.

Obrátil jsem se na jednoho člena klubu Spectristů z Karolínky, který soustřeďoval programy na tiskárnu T 85. Pro BT 100 jsem obdržel jen program LPRINTBT100 ve verzi 1 a 2. Po nahrání do počítače se nic nedělo. Zcela náhodou jsem z dokumentace k těmto programům zjistil, že zapojení OUT a IN tiskárny na konektor interface UR 4 je jiné, než bylo v dokumentaci k BT 100 (později jsem z časopisu Mikrobáze 2/89 zjistil, že program LPRINT je k počítači Didaktik Gama). Po přepojení na UR 4 program ve verzi 1 tiskl s chybami, verze 2 tiskla výborně.

Tisk obrazovky však byl stále na polovinu šířky A4.

Pátral jsem dále po textovém editoru a programu pro tisk obrazovky. V dalším Zpravodaji Karolínky byl inzerát na soustřeďování programů na BT 100. Obdržel jsem 10 programů, m.j. i Tasword, ovšem zase jen s tiskem na polovinu šířky A4. Současně jsem dostal informaci, že programy pro BT 100, konkrétně text.editory, upravuje D.Jenne z Českých Budějovic. Tam jsem ho nezastihl s tím, že má být v Berouně. Zde jsem se dotazoval na Zenitcentru, kde jsem zjistil, že prodávají programy Textový editor a program pro tisk obrazovky na BT 100. Bezeplatně jsem si je objednal. Avšak programy nefungovaly.

Dotazem na Zenitcentrum jsem nic nezjistil (nakupujte u odborníků). U programu pro tisk obrazovky byl uveden jeho autor. Podarilo se mi vypátrat jeho bydliště a po několika telefonních hovorech jsem zjistil, že OUT a IN má zase jiné zapojení na UR 4, než určovala Tesla Přelouč. Přitom jsem se dozvěděl, že dle Přelouče existují již tři zapojení.

Dle zmíněného autora jsem zapojil konektor UR 4 a tiskárna se rozeběhla. Ale textový editor opět tiskl jen na půl šířky A4 drobnými znaky. Program pro tisk obrazovky měl zvětšené znaky na šířku i výšku, řádky z obrazovky byly tištěny nikoliv na šířku, ale délku A4. Takže další neúspěch. Napsal jsem vedoucímu Zenitcentra v Berouně, proč takové programy vyrábějí a rozšiřují. Bohužel odpověď jsem nedostal.

V inzerátu Amatérského radia řady A z května t.r. byl inzerát ing. Vlčka, nabízejícího programy pro BT 100. Proto jsem ho požádal o textový editor, tisknoucí na celou šířku A4 a po zkušenostech i o informaci, na jaké zapojení tiskárny a UR 4 je program udělán. Program jsem obdržel na kazetě za 100 Kčs i s pokyny, kde ovšem chyběly informace o zapojení na UR 4. Musel jsem čekat 5 týdnů, než jsem obdržel odpověď, že se jedná o program na Didaktik gama. Znovu jsem proto přepojil vodiče na UR 4 z portu C na polovinu portu A a B a konečně začala tiskárna pracovat. Po seřízení tiskárny (zdvih jehly, změna rychlosti) je tisk kvalitní, na úrovni profesionální tiskárny. Jen vkládání papíru je ostudné a ta rychlost je pomalá, při 50 řádcích a málo mezerách v textu trvá stránka A4 přibližně 20 minut.

S pozdravem

Jiří Hušek
 Švabinského 1721
 356 05 Sokolov

Kopírovací programy

pro ZX Spectrum

Petr Piskovský

Nabídka programového vybavení pro osobní počítače v naší obchodní síti je nepatrná vzhledem k množství programů, které pro ně existují. Počítače ZX Spectrum, jako jedny z nejrozšířenějších, nejsou výjimkou a jejich řady se nadále rozrůstají o majitele počítačů DIDAKTIK GAMA. Nejschůdnější cestou, jak programy získat, je jejich okopírování od kolegy, nebo v klubu. O právních okolnostech tohoto postupu již bylo napsáno mnoho, ale odpovídající opatření se teprve připravují a na druhé straně by to ale znamenalo mít dostatek možností k legálnímu nákupu programů. Těm, kteří programy uchovávají na magnetofonových kazetách, "viry" nehrozí a pokusy programátorů vytvářet nekopírovatelné verze mají jen omezený účinek. Proto kopírovací programy zatím patří do základního programového vybavení.

Tento článek není podrobným návodem k používání kopírovacích programů, ale měl by posloužit k základní orientaci mezi nejrozšířenějšími typy a jejich zvláštnostmi. Kolikrát jste měli v ruce půjčenou kazetu s řadou programů, jejichž názvy vám nic neříkaly a často vám více neřekl ani majitel, neboť program kdesi okopíroval a dosud se k němu jaksi nedostal.

1. PRINCIPY KOPÍROVÁNÍ

Časově méně náročné, ale také méně spolehlivé je kopírování pomocí dvou magnetofonů s využitím vazby mezi zdičkami EAR a MIC na počítači. Ani není nutné používat pomocný program např. BITCOPY, nebo LOAD. Výhodnější je použít dvojkazetový magnetofon, ovšem kvalitní, nejlépe se zrychleným nahráváním a kvalitní pásky. Proti použití kopírovacího programu se ušetří tímto postupem polovina času, ale není možnost ověřit správnost nahrávky, což se někdy nemusí vyplatit.

Nejpoužívanějším způsobem kopírování je použití kopírovacího programu. Do počítače se nahraje kopírovací program, například TF COPY. Vybere se funkce LOAD a spustí magnetofon s kopírovaným programem. Kopírovací program si řídí ukládání bloků programu do paměti počítače. Pak se založí druhá kazeta a program se uloží na pásek volbou funkce SAVE. Většina kopírovacích programů umožňuje ověření správnosti nahrávky (VERIFY).

Kvalita nahrávky je při tomto způsobu vykoupena velkou spotřebou času. Program dlouhý 5 minut se nahrává, ukládá a ověřuje nejméně 15 minut. Doba nutná ke zkopírování hodinové kazety je značná.

2. JEDNOÚČELOVÉ A JEDNODUCHÉ KOPÍROVACÍ PROGRAMY

Jednoúčelové kopírovací programy vznikaly pro kopírování dlouhých programů, kdy bylo nutno umístit kopírovací program do přesně vymezené oblasti paměti. Druhý typ jednoúčelových kopírovacích programů slouží pro kopírování programů používajících jiný způsob nahrávání. Např. FULL THROTTLE COPY pro kopírování hry FULL THROTTLE, nahrávané vyšší přenosovou rychlostí. Z dalších jednoúčelových kopíráků uvedu alespoň COPY RAID, MATCH DAY COPY nebo PSYTRON COPY. Jejich význam pro kopíro-

vání je v dnešní době zanedbatelný, neboť moderní univerzální kopírky je plně nahradí.

První kopírovací programy, vznikající po uvedení Spectra na trh byly velice jednoduché pokud se týká nabízených služeb i komfortu obsluhy. Masovému kopírování programů bránily tím, že umožňovaly nahrát jen hlavičku nebo blok dat (S-COPY), či hlavičku i s blokem dat (THE KEY) a jedenkrát uložit. Kopírovat takto mnohoblokový program, např. BOMB JACK byla lahůdkou. CPY PROLOG uměl nahrát celou sadu bloků a měl 38634 bytů volné paměti. CLONING, nebo j.p. clone uměl nahrát i blok bez hlavičky.

I tyto kopírovací programy dnes zůstávají zapomenuty na magnetofonových páscích, jako svědci nezadržitelného pokroku a zdokonalování.

3. UNIVERZÁLNÍ KOPÍROVACÍ PROGRAMY

Tyto programy mají základní funkce LOAD, SAVE a většinou i VERIFY. Najednou lze kopírovat i programy skládající se z několika bloků. Každý pak umožňuje některé speciální funkce, které jej odlišují od těch druhých a lze si vybrat to co zrovna potřebujeme - přejmenovat hlavičku, vymazat blok, zadat POKE, nahrát maximálně dlouhý blok, pipnout na konci programu a další funkce. Jejich podrobnější popis je v kapitole 5.

4. DALŠÍ KOPÍROVACÍ PROGRAMY

Pro ZX Spectrum s pamětí 128 kB jsou vytvořeny kopírovací programy Mr. Bobby, Dr. COPY a MASTER COPY 128.

LC+ je označení kopírovacího programu pro Spectrum s pamětí 80 kB. Majitelé microdrive jistě používají program copier, nebo trans expres. Trans expres je pětidílný program, který spolupracuje s microdrivem i magnetofonem a umožňuje i přenos dat z magnetofonu na microdrive a naopak.

Další skupinou jsou programy, které ukládají data vyšší rychlostí než je standartní (1500 Baudů). Jsou to programy TWO SPEED COPY, TAPE SYS (který čte a ukládá data volitelnou rychlostí od 1431 do 3635 Bd.), SPEEDYLOAD (používající dvojnásobnou rychlost záznamu), nebo QUICKSAVE (Q-SAVE) (ten umožní až čtyřnásobné zrychlení záznamu. Vyšší rychlost záznamu umožňují i programy Mr. COPY++ a OMNICOPY 2.

5. CO UMÍ KOPÍROVACÍ PROGRAMY

Pro vyzkoušení kopírovacích programů jsem si vytvořil zkušební soubor, který obsahoval čtyři základní bloky vytvářené Spectrem, dva bloky uložené bez mezer mezi bloky, tak jak je vytváří například Mr. COPY, dva bezhlavičkové bloky (v dalším textu označené jako blok I) a blok, který měl v hlavičce menší údaj o délce (blok II). Některé kopírky totiž například po načtení hlavičky s údajem o délce 100 bytů nenahrávají blok s délkou 150 bytů, stejně jak takový blok nenahrávejte přímo do počítače.

V popisu jednotlivých programů je za názvem uveden rok vzniku, autor, délka volné paměti a označení základních funkcí: L-LOAD, S-SAVE, D-DELETE, V-VERIFY, C-vymazání všech uložených bloků, R-RESET a návrat do BASICU s vymazáním paměti. Následuje popis dalších zajímavých možností.

Začnu od těch starších:

1. THE KEY v.7

1983 / KEYSOFT / L,S

Program se nahraje příkazem LOAD""CODE a sám se spustí. Umožňuje nahrát jednu hlavičku s příslušným blokem a jedenkrát uložit na pásek. Stlačením "Y" lze nahrát další blok.

2. COPY PLUS

1983 / A.D. by DAKRASOFT / 41781 / L,S,R

Nahrave blok s hlavičkou i bez ní, přidá blok dat do paměti.

3. cloning

L i S bloku s hlavičkou i bez ní. Nezobrazuje se žádný údaj o nahraném bloku.

4. S-COPY

Úplně jednoduchý program umožňující L a S a to jen hlavičky nebo bloku.

5. CPY PROLOG

38634 bytů / L,S

Jednoduchý program, ale test nahraje celý.

6. jp clone

1984 / L,S

Nahrává se LOAD""CODE a startuje např. RANDOMIZE USR 23296. Nahrave jeden blok s hlavičkou i bez a jednou uloží. Návrat do BASICU program nemaže.

7. TAPE COPY

1984 / E. WEDEKIN / L,S

Nahrave blok s hlavičkou i bez. Blok II. hlásí chybu, ale do paměti se uloží. Uložit na pásek lze vícekrát.

8. ISO COPY 1

1984 / Individual Software / 42001

ISO COPY 2

1984 / Individual Software / 48794

Podrobný popis vlastností a použití obou programů je v /2/.

ISO COPY 1 má tři různé nahrávací rutiny. Jedna např. umožní LOAD bloku s poškozeným zaváděcím tónem. Funkce BREAKER zase umožní zkopírovat blok, který je tak dlouhý, že do paměti nevejde společně s kopírovacím programem. Funkce rozdělí tento dlouhý blok na dva samostatné bloky. Funkce MAXCOPY vymaže některé svoje podprogramy pro zrychlené nahrávání a uvolní 42493 bytů. Pro jednorázové zkopírování lze použít 49122 bytů.

ISO COPY 2 umožňuje nahrávat z pásku programy, které jsou uloženy jinou než standardní rychlostí. Pomocí funkce ADJUST si program otestuje skutečnou rychlost a tou pak provádí nahrávání.

Oba tyto programy patří spíše mezi speciální kopírky, jež naleznou uplatnění jen při využití svých zvláštních vlastností.

9. COPY COPY

1984 / Tadesz Wilczek / 42240 / L, S, V

Program s mnoha možnostmi. Jejich podrobný popis je v /5/ a /6/. K těm zajímavým patří možnost zadat POKE, spustit strojový program, provést výpis paměti, prohlédnout hlavičky souborů

uložených na pásku, na tiskárně vypsat seznam souborů načtených do paměti. Lze změnit název hlavičky včetně parametrů a zkopírovat blok dlouhý 49096 nebo 49152 bytů. Umi rozdělít dlouhý blok na dvě části. Testovací program kopíruje celý.

Stejné možnosti poskytují i programy WA COPY a SUPER COPY. Kromě jiného názvu jsem žádné rozdíly nenašel.

10. KOPIRAK

1984 / A.Skubanič / L, S, V, R

Ke kopírce patří program KOP-MAN popisující její použití. Nahrává bloky s hlavičkou i bez ní. Najednou uloží až 50 bloků s hlavičkami, ale jejich názvy nevypisuje.

11. CPY

1984 / Petr Adánek / 39260 / L, S, V, D, C, R

Program dále umí přesunout nahraný blok nebo jej zdvojit, přejmenuje hlavičku. Vypisuje pouze typ a jméno bloku.

Blok I. ani II. nenahrave.

12. Mr. COPY

1984 / P.Raczkowski / 42237 / L, S, V, C, R

Oblíbený kopírovací program, kterému však lze vytknout to, že neumožní DELETE a při SAVE ukládá bloky bez mezer, což může působit potíže při načítání do jiných kopírek nebo monitorům pásků (TAPE MONITOR). Tento nedostatek odstranili programátoři pracující pod značkou "amisin".

Další verzi programu je Mr. COPY+ od Ing. P.Kreybiga (KYSOFT) z roku 1986. Popis rozšířených možností je součástí programu. Vytváří mezery mezi bloky, na konci SAVE dává zvukovou značku (BEEP) a lze kopírovat metodou Bitcopy.

Již zmíněný Mr.COPY++ upravil Jan Šulc pro SAVE rychlostí 3500 Bd, ale bez meziblokových mezer.

Všechny uvedené verze nahrávají celý zkušební test. Z programu Mr.COPY vychází i verze PIRACY! (PIRATE).

13. MASTERCOPY

VATROSLAV SOBOT, Záhřeb / 45924 / L, S, C, R, D

Na obrazovce je zobrazen údaj jen o jednom bloku, ale záznamy lze listovat. Pro nahrání bloků I. a II. je nutno volit režim Headerless. Maxload nahrave 49152 bytů.

14. multicopy v2.2

41723 / L, S, C, R

Při ukládání ma pásek dává zvukovou značku. Má režim Maxbyt nezjištěné délky, Exit provede RESET počítače ihned. Může prohlížet názvy souborů na pásku.

Tímto programem jsem "zachránil" několik programů, které nenahrály jiné kopírky a nešly nahrát ani přímo do počítače.

15. OMNICOPI 2

41562 / L, S, V, D, C, R

Nahrave všechny bloky testu; při SAVE nevytváří meziblokové mezery; lze volit dvojnásobnou rychlost ukládání. Vytvoří novou hlavičku nebo v hlavičce zruší povel LINE pro autostart programu. Má režim MEGAFILE nezjištěné délky. Graficky pěkně provedený program.

16. CPY TWO

1985 / Nejedlý / 39000 / L, S, R

S nahranými bloky programu lze před jejich uložením na pásek provést přesun, zdvojení, vymazání. Hlavičku je možné přejmenovat nebo změnit údaj o autostartu či počáteční adrese. Mezeru

mezi bloky lze volit od 0 do 9 sec.

Z testovacího programu nenahraje blok I. ani II.

17. LADY COPY

1985 / Ladinek / 42237 / L, S, D, C, V, R
Grafický obdobný program jako Mr.COPY doplněný o funkci DELETE a další možnosti: změnu textu hlavičky a LONG COPY o délce 49101 bytů. Při ukládání nevytváří mezi bloky mezeru.

Test nahraje celý.

18. FREE COPY

1985 / HAJASOFT / 47055 / L, S, V, C, R
V programu lze nastavit délku pauzy mezi bloky. Pro počítání ukládaných bloků je připraveno třímístné počítadlo. Po ukončení nahrávání je nahrána i zvuková značka.

Testovací program nahrán celý.

19. PINK COPY

1986 / P.Raczkowski / 42240 / L, S, V, D, R
Další kopírka autora programu Mr.COPY. Program je doplněn o možnost čtení hlaviček na pásku, SAVE od A do B, delete od označeného bloku do konce a kopírování bloku dlouhého 49092 bytů.

Test se nahraje celý, ale na obrazovce se zobrazí jen 14 řádků.

20. TF COPY

verze 11/86, 6/87, 1/88 a 2/88

RNDr. Arnošt Večerka

Tento program se stal bezesporu hitem mezi kopírovacími programy a to jak komfortem obsluhy, tak hlavně kopírovací kapacitou. Přestože nejvyšší kapacita uvedená v menu je 44288 bytů, je velice málo programů, které je nutno kopírovat nadvakrát. Program využívá kompresi dat při nahrávání a jejich expanzi při ukládání. Podstatou je zvláštní způsob úsporného ukládání stejných bytů následujících za sebou. Například textových souborů z TASWORDU se vejde do paměti tolik, že součet jejich délek přesahuje kapacitu celé paměti RAM. Pro ilustraci: blok 10 000 nul zabral v paměti 649 bytů. Pro usnadnění obsluhy lze použít různé parametry. Např. povel

VERIFY 3-E ověří bloky od čísla 3 do konce.

Podrobný popis verze 1/88 byl v /4/. Zvláštností této verze je možnost programového zastavení magnetofonu. Verze 1/88 má základní rozsah paměti 44800 bytů a účinnější kompresi dat. 10.000 stejných bytů zabere 613 bytu paměti.

Verze 11/86 byla použita pro úpravy jinými autory. SUPER COPY je označení verze, která dále sdělí: B-D B. Kolarčík Frydek, ale to je celý rozdíl, který se mi podařilo zjistit.

CRAZY COPY umožní zvolit zvukovou značku na konci nahrávání. Další verze TF COPY tuto možnost mají také.

21. O-COPY v 1.1

1988 / Martim Moráček / 44288

Dále vylepšený program vycházející z TF COPY v 1/88. Umožňuje různé volit umístění zvukové značky nebo délky mezer mezi bloky. Nejvíce patrným vylepšením je průběžné zobrazování délky načítaného, ukládaného nebo ověřovaného bloku.

Při sestavování těchto údajů jsem zaznamenal i uživatele tohoto programu, kteří měli potíže s nahráváním programů kopírovaných pomocí O-COPY. V šumperském Sinclair klubu jsme však s ním dosud žádné potíže neměli.

Tolik o kopírovacích programech, které jsem měl možnost vyzkoušet. Jistě jsou i další kopírovací programy, ale mezi uvedenými je dost těch, které splní vaše náročné požadavky.

Literatura:

- /1/ Spectrum Software Introduction - vyd. Karolinka
- /2/ ISO COPY - překlad manuálu Ota Adametz, Karolinka
- /3/ Zápisník Z 89 č. 9; O-COPY
- /4/ Zpravodaj č. 10, Karolinka
- /5/ Mikrobáze 2/88 - COPY COPY
- /6/ Rozšíření možností ZX díl I., ČSVTS VZÚ NHKG Ostrava

Programové vybavení od Robotronu na jarním lipském veletrhu 1989

Ing. Erich Terner

Pro vědecké zpracování informací je určen programový systém AIDOS/M-2 DCP. Slouží k uchování a opětnému vybavování libovolné dlouhých informací jako např. pro anotace knih, záznamy časopiseckých článků, výzkumných zpráv, komerčních přehledů, seznamů výrobků, obsahů vynálezů, cestovních zpráv, lékařských nálezů aj.

Zpracování textů TEXT 40 K podstatně racionalizuje kancelářské práce v německém i v ruském jazyce.

Hotelový systém HCS/R-16 slouží v hotelích k rezervaci pokojů, k vyúčtování pobytu, správě materiálu a zboží i účtářské práci včetně statistiky.

Pro usnadnění sestavování programového vybavení je určen systém MORES DCP. Pracuje pomocí jazyka MODULA-2 a podstatně usnadňuje všechny práce na tvorbě programového vybavení.

Provozní systém mnoha druhů bank dat se nazývá ALLDBS SVP. Je použitelný pro interaktivní systémy tam, kde se vyžaduje složitá architektura banky dat.

Pro účely plánování v závodech, podnicích, mi-

nisterstvech a v plánovací komisi slouží systém PLANAP. Zpracovává plány podle ekonomických problémů vzhledem k potřebným parametrům.

Společným vývojem Robotronu a Centrprogram-systemu v Kalininu (SSSR) vznikl provozní systém pro banky dat INTERBAS. Je kompatibilní se systémem DBS/R.

Programové vybavení potřebné pro práce na umělé inteligenci je nový systém Robotronu na základě jazyků LISP a PROLOG.

Lineární optimalizaci slouží systém OPTI-LO DCP, který je užitečný při krátkodobém a střednědobém plánování.

Pro plánování dopravy a při optimálním nasazování vozidel je užitečné používat systému TOUR DCP.

Při pracích v účetnictví a statistice je důležitým pomocníkem systém POESY. Pomáhá při vypočítání nákladů, dále při vypočítání mezd, vedení základních a jiných prostředků, při všech pracích pro investiční účely, při skladovém účetnictví, finančním a odbytovém hospodářství.

Uživatelské funkce mikropočítače ZX Spectrum

Rostislav Gemrot

Výbornou vlastností Basicu implementovaného ve Spectru (a Didaktiku Gama) jsou uživatelem definované funkce. Mají podstatně širší možnosti než na jiných mikropočítačích užívaných v ČSSR (IQ 151, PMD-85). Umožňují definovat jak číselné funkce, tak funkce řetězcové, a to s neomezeným počtem předávaných parametrů.

Funkce číselné se definují DEF FN x (a1, a2, ..., an) = výraz, kde x je jednoznakové jméno definované funkce, a1, a2, ..., an jsou parametry funkce. Parametry mohou být jak číselné, tak řetězcové, avšak zásadně s jednoznakovým názvem. Parametry se chovají jako parametry volané hodnotou, jako v jazyce Pascal, tj. v momentě volání uživatelské funkce jsou parametrem a1, a2, ..., an přiděleny konkrétní hodnoty vypočtené z volací funkce FN x (b1, b2, ..., bn). Na místech b1, b2, ..., bn mohou stát libovolné výrazy.

Zajímavou předností je vytváření skutečných lokálních proměnných se jmény a1, a2, ..., an při volání dané funkce. Ve většině případů je to bezvýznamné, neboť hodnoty proměnných a1, a2, ..., an se nemohou ve výpočtu měnit, ale lze této skutečnosti, jak uvádím v příkladu (program 1), dobře využít.

Řetězcové funkce se definují touto formou: DEF FN x\$ (a1, a2, ..., an) = výraz. Vše platí ve stejném rozsahu, jako pro číselné funkce, avšak výraz musí být typu řetězec.

Definiční rovnice uživatelských funkcí mohou stát kdekoli v programu, funkce je interpretu známa právě tehdy, když je její definice zařazena jako programový řádek.

Nelze tedy definovat funkci v přímém režimu, i když interpret při takovém pokusu neohlásí chybu. Následné volání takto definované funkce však vyvolá chybu "P-FN Without DEF". Interpret totiž nemá zvláštní paměťovou oblast pro uživatelem definované funkce. Informace o jménu, parametrech a funkčním výrazu je obsažena v paměti pouze jednou, a to na příslušném programovém řádku.

Při volání uživatelské funkce FN x (b1, b2, ..., bn) postupuje interpret takto: prohledává postupně Basicový program od začátku a hledá instrukce DEF FN. Když najde tuto instrukci, srovnává jména funkcí. Projde-li celý program bez shody jmen, hlásí interpret chybu "P-FN Without DEF". V případě shody jmen je srovnáván počet a typy parametrů. Při neshodě typu či počtu parametrů je hlášena chyba "Q-Parameter error". Dále vypočte hodnoty výrazů b1, b2, ..., bn a získaná konkrétní čísla dosadí do připravených míst v příkazu DEF FN. Za každým formálním jménem parametru a1, a2, ..., an je ponecháno 5 bajtů pro jeho hodnotu. Pak legalizuje tyto parametrické proměnné a provede funkční výraz. Získaná hodnota je i hodnotou funkce FN x (b1, b2, ..., bn).

Z toho plyne důležitý poznatek: je třeba umísťovat příkazy DEF FN jako první řádky programu, výpočty těchto funkcí pak poběží rychleji, než když budou DEF FN umístěny někde uprostřed programu.

Díky skutečné tvorbě lokálních proměnných pro výpočet funkčního výrazu, je možné definovat funkce i rekurzivně. Např. je možné naprogramovat počítání faktoriálu takto:

```
DEF FN f(n) = VAL "n*FN f(n-1)-1+1"(12*(n=0)+1 TO)
```

Dále uvádím možnost efektního využití tvorby lokálních proměnných.

Požadujeme, aby program počítal libovolnou funkci, kterou uživatel zadá z klávesnice až při jeho běhu. Potom stačí, aby počítaná funkce byla uložena do nějaké řetězcové proměnné, např. a\$, a aby byla zadána v předem známé proměnné, např. x. Dále definujeme: DEF FN a(x) = VAL a\$. Potom vždy když chceme znát hodnotu funkce uložené v a\$, stačí volat FN a(m), kde m je libovolný výraz. Podle výše uvedeného postupu je totiž hodnota m přiřazena x a je i proveditelná funkce VAL a\$, neboť x skutečně existuje jako lokální proměnná. Tuto možnost uvádím v demonstračním programu 1, který počítá kořen rovnice metodou půlení intervalu. Pro srovnání je připojen i program 2, který počítá totéž, ale bez užití definované funkce, a který je na první pohled méně přehledný.

Uživatelské funkce znamenají nespornou výhodu pro programátora (jednoduše se pracuje s určováním hodnot nestandardních funkcí) i pro přehlednost programu (stává se jednodušším a mnohem přehlednějším). Jejich nevýhodou je však pomalost, se kterou jsou prováděny a v případě rekurzivních funkcí i paměťová náročnost. Rekurzivně definovaná funkce pro výpočet faktoriálu je např. 3x pomalejší než podprogram počítající totéž. Programátor musí proto zvážit optimální poměr mezi rychlostí a přehledností programu.

Program 1:

```
10 DEF FN a(x)=VAL a$
20 INPUT "Vlož funkci v proměnné x: ";a$
30 INPUT "Přesnost: ";e
40 INPUT "Meze od: ";a,"do: ";b
50 IF FN a(a)*FN a(b)>0 THEN PRINT #0;"Meze nevyhovující!":
  PAUSE 0: GOTO 40
60 LET c=(a+b)/2
70 IF FN a(c)*FN a(b)>0 THEN LET b=c
80 IF FN a(c)*FN a(a)>0 THEN LET a=c
90 IF FN a(c)<>0 and (b-a)>e THEN GOTO 60
100 PRINT "Kořen je: ";c
110 PRINT "S přesností: ";e
120 STOP
```

Program 2:

```
5 INPUT "Vlož funkci v proměnné x: ";a$
10 INPUT "Přesnost: ";e
20 INPUT "Meze od: ";a,"do: ";b
30 LET x=a: GOSUB 200: LET m=v
40 LET x=b: GOSUB 200: LET n=v
50 IF m*n>0 THEN PRINT "Meze nevyhovující!": PAUSE 0: GOTO 20
60 LET c=(a+b)/2
70 LET x=a: GOSUB 200: LET m=v
80 LET x=b: GOSUB 200: LET n=v
90 LET x=c: GOSUB 200: LET o=v
100 IF o*n>0 THEN LET b=c
110 IF o*m>0 THEN LET a=c
120 IF o<>0 and (b-a)>e THEN GOTO 60
130 PRINT "Kořen je: ";c
140 PRINT "S přesností: ";e
150 STOP
200 LET v=VAL a$: RETURN
```

Využití přerušeni u počítače Sharp MZ-800

Michal Kreidl

Mikroprocesor Z-80 má tři módy maskovatelného přerušeni. U počítače Sharp jsou prakticky využitelné jen dva a těmi se budu dále zabývat. Vše je popisováno čistě z hlediska uživatele, který má v úmyslu psát programy ve strojovém kódu s využitím přerušeni. Proto neuvádím zbytečné podrobnosti o průběhu zpracování přerušeni procesorem, ale spíše možné zdroje přerušeni a činnost programu po vzniku požadavku o přerušeni.

1) Mód přerušeni 1

Je-li přerušeni povoleno instrukcí EI, pak po přijetí požadavku a dokončení právě probíhající instrukce, provede procesor volání obslužného podprogramu na adrese 0038h (stejný efekt má provedení instrukce RST 38h). V dolní paměti ROM je na tomto místě skok na adresu 1038h. Současně je zakázáno další přerušeni. V obslužném podprogramu je nutné nejprve uložit instrukcemi PUSH všechny registry použité pro obsluhu přerušeni. Pokud se používá jen jedna sada registrů, je výhodnější ji uchovat instrukcemi EXX a EX AF,AF'. Následuje vlastní obsluha přerušeni. Po jejím ukončení se musí do použitých registrů vrátit jejich původní obsah instrukcemi POP, resp. EXX a EX AF,AF'. Poslední činností před návratem do přerušeniho programu bývá povolení přerušeni instrukcí EI. Návrat se pak provádí instrukcí RET.

Zdrojem přerušeni v módu 1 může být pouze výstup 3. čítače obvodu 8253 a to pouze tehdy, je-li 3. bit brány C obvodu 8255 nastaven na '1'. Vstup tohoto čítače je spojen s výstupem 2. čítače, na jehož vstup přichází periodický signál s kmitočtem přibližně 15611 Hz. Toto uspořádání umožňuje přerušování programu periodicky s periodou od zhruba 256 μ s do 76 hodin, nebo jednorázově po uplynutí dané doby ve stejném rozsahu. K dosažení této činnosti je nutné naprogramovat 2. čítač do módu 3 a 3. čítač do módu 0. Pro dosažení periodického přerušování je nutné do 3. čítače při obsluze přerušeni vždy znovu zapsat předvolbu. Délku periody, nebo časového intervalu, pak lze určit jako $64 \mu s \times N1 \times N2$, kde N1 a N2 jsou konstanty zapsané do čítačů.

Použití přerušeni v módu 1 ilustruje následující příklad. Uvedený program zajišťuje generování přerušeni v sekundových intervalech, např. pro nepřetržitě zobrazování reálného času. Předpokládáme, že počítač je v textovém módu (700) s připojenou dolní i horní paměti ROM:

```
DI          ; zákaz přerušeni
IM1         ; určení módu přerušeni
LD A,0C3H   ; uložení skoku do obslužného
LD (1038H),A ; podprogramu na adresu 1038h
LD HL,PRERUS
LD (1039H),HL
LD A,76H    ; nastavení 2.čítače do módu 3
LD (0E007H),A
LD A,15H    ; zápis konstanty 1301
LD (0E005H),A
```

```
LD A,5
LD (0E005H),A
LD A,0B6H   ; nastavení 3.čítače do módu 3
LD (0E007H),A
CALL NASTAV ; zápis předvolby
LD A,5      ; nastavení bitu PC2 na '1'
LD (0E003H),A
EI          ; povolení přerušeni
.
.          ; hlavní program
.
HALT
```

```
NASTAV: LD A,0CH   ; zápis konstanty 12 do 3. čítače
LD (0E006H),A   ; (12x1301x64us = 1s)
LD A,0
LD (0E006H),A
RET
```

```
PRERUS: PUSH AP   ; obslužný program
        PUSH BC   ; úschova registrů
        PUSH DE
        PUSH HL
        CALL NASTAV ; zápis předvolby do 3. čítače
        .
        .          ; vlastní obsluha
        .
        POP HL    ; obnovení registrů
        POP DE
        POP BC
        POP AP
        EI        ; povolení dalšího přerušeni
        RET       ; návrat do přerušeniho programu
```

2. Mód přerušeni 2

Tento mód je složitější než předchozí a hlavní rozdíl spočívá v tom, že zde procesor vyžaduje spolupráci přerušující periferie. Je-li přerušeni povoleno, pak po přijetí požadavku přerušeni proběhne speciální cyklus, ve kterém se přečte z periferie, od níž požadavek pochází, osmibitový údaj. Ten představuje nižších osm bitů tzv. vektoru přerušeni, jehož vyšších osm bitů je určeno obsahem registru I procesoru. Takto vytvořený vektor adresuje buňku paměti počítače, v níž leží nižší polovina adresy obslužného podprogramu, vyšší polovina leží na adrese o jedničku vyšší. Z tohoto vyplývá, že v módu 2 může existovat až 128 různých požadavků o přerušeni a pouhou změnou registru I lze naráz změnit jejich obsluhu. Volání obslužných podprogramů se provádí obdobou instrukce CALL s adresou získanou výše popsáním způsobem. Na rozdíl od módu 1 je zde bezpodmínečně nutné provádět návrat do hlavního programu instrukcí RETI a ne RET. To proto, aby přerušující periferie poznala, že její požadavek již byl obslužen. Důležité je to především při prioritním řazení více periferních obvodů. Jinak je tvar obslužných rutin stejný jako v módu 1.

Zdrojem přerušeni v módu 2 je v našem počítači standardně jen obvod Z-80 PIO (na sběrnici lze

samozejmé připojit další obvody, jako např. CTC, SIO atd.). Tento obvod v sobě v podstatě sdružuje dva zdroje přerušeni, jedním je kanál A, druhým kanál B. Kanál A čte mimo jiné i výstup čítače 0 obvodu 8253 a vertikální zatemňovací impulzy. Oba tyto signály mohou způsobovat přerušeni. Kanál B je využitý ke komunikaci s tiskárnou a může generovat přerušeni vždy po přijetí dat, vyslaných počítačem přes tento kanál. Toho lze využít pro paralelní běh programu a tisk dat z vyrovnávací paměti (tak jak to umožňuje např. BASIC). Pozor, na rozdíl od generování přerušeni v módu 1, které je přímo z výstupu čítače. Tam procesor reaguje na úroveň 1, výstupu čítače, kdežto zde obvod PIO reaguje na hranu signálu (polaritu lze zvolit). Také je vhodné vynulovat 3. bit brány C, obvodu 8255, přestože je toto nastavení provedeno při startu ROM monitoru.

Následující příklad ukazuje využití obvodu PIO pro generování přerušeni každých 20ms, což lze s výhodou použít např. pro testování stisku některých kláves, blikání kursoru apod. Program by fungoval v textovém i grafickém módu:

```

DI          ; zákaz přerušeni
IM2         ; mód přerušeni 2
LD A,10H    ; předpokládejme umístění tabulky
LD I,A      ; adres obslužných rutin na adrese
             1000h
LD HL,PRERUS ; adresa obslužné rutiny
LD (1000H),HL ; umístění adresy do tabulky
LD HL,PROGA ; tabulka řídicích slov pro kanál A
LD BC,5FCH  ; B=počet slov, C=adresa řídicího
             registru PIO-A
OTIR        ; naprogramování kanálu A
EI          ; povolení přerušeni
.           ;
.           ; hlavní program
.           ;
HALT

PRERUS: EX AF,AF' ; úschova pracovní sady registrů
EXX
.           ;
.           ; obsluha přerušeni
.           ;
EXX        ; přepnutí původní sady registrů
EX,AF,AF'
EI         ; povolení dalšího přerušeni
RETI      ; návrat do hlavního programu

PROGA: DB 0H   ; nižší byte vektoru přerušeni
DB 0CFH      ; bitový mód kanálu
DB 3FH       ; rozdělení bitů na vstupní a
             výstupní
DB 97H       ; přerušeni povoleno při úrovni 0
DB 0DFH      ; z bitu 5

```

Obdobně lze využít i signálu z časovače, ten je ale nutné nejprve naprogramovat. Pak lze generovat přerušeni jak jednorázově tak i periodicky. Ještě jednou upozorňuji na zcela bezpodmínečné ukončení obsluhy přerušeni instrukcí RETI (i v případech mimořádného ukončení jako zjištění chyby apod.). Při nedodržení této zásady dojde k zablokování všech dalších požadavků na přerušeni a je nutné instrukcí RETI vyslat ručně, nebo počítač vynulovat.

Třetí příklad ilustruje již zmíněnou aplikaci přerušeni při obsluze tiskárny. Obslužný program vytiskne při paralelním běhu programu hlavního obsah vyrovnávací paměti. Předpokládejme současné použití přerušeni z příkladu předchozího:

```

.
.
DI
LD HL,BUFFER ; adresa tištěné vyrovnávací paměti
LD (POZICE),HL ; uložení ukazatele na znak k
                vytištění
LD HL,TISK    ; uložení adresy obslužné rutiny
LD (1002H),HL ; do tabulky
LD HL,PROGB   ; řídicí slova pro kanál B
LD BC,2FDH   ; B=počet slov, C=adresa řídicího
                registru PIO-B
OTIR         ; naprogramování kanálu B
EI
.
.           ; hlavní program, zaplnění
.           ; vyrovnávací paměti
LD A,83H    ; povolení přerušeni z kanálu B
OUT (OPDH),A
CALL TISK   ; a vyslání prvního znaku
.
.
HALT

TISK: EX AF,AF' ; úschova pracovní sady registrů
EXX
LD HL,(POZICE) ; zjištění adresy znaku pro tisk
LD A,(HL)      ; vybrání tohoto znaku
INC HL        ; posun ukazatele na další znak
LD (POZICE),HL ; a uložení ukazatele
OR A          ; je-li nalezen znak ukončující
              ; tisk (0)
JR Z,KONEC   ; bude zakázáno další přerušeni
              ; z kanálu B
.
.           ; vyslání znaku z registru A na
.           ; kanál B a impulzu STB na bit 7
.           ; kanálu A,
.           ; nečeká se na potvrzení přijetí
.           ; tiskárnou
JR NÁVRAT

KONEC: LD A,3   ; zákaz přerušeni
OUT (OPDH),A

NAVRAT: EXX    ; přepnutí původní sady registrů
EX AF,AF'
EI
RETI

PROGB: DB 02H  ; nižší byte vektoru přerušeni
DB 0FH        ; kanál výstupní
POZICE: DS 2   ; ukazatel na výstupní
              ; vyrovnávací paměť

```

Doufám, že tento krátký přehled možností přerušeni v počítači MZ-800 pomůže zájemcům o programování ve strojovém kódu, proniknout do jedné z důležitých oblastí této problematiky. Na závěr ještě malá rada pro začátečníky: nevzdávejte to při prvních neúspěších, vždy se snažte najít vlastní chyby, počítač určitě funguje správně. Pokud se vám to nepodaří, obraťte se na zkušenější kolegy, zcela jistě vám rádi poradí (pokud ne, tak jsou na tom možná stejně jako vy!). *

dBase III stále půvabná

I po uvedení dBase IV na trh zůstává dBase III kvalitním nástrojem v rukou zkušeného programátora. Pro budoucí "zkušené" programátory uvádí Mike

Lewis v časopisu PCW 6/89 - str. 178 několik užitečných rad, které se jistě hodí i vám. Pokud hledáte o dBase III rozsáhlejší informace, najdete je v Interaktivním kursu databázového systému dBase III, který uvedla na trh 602. ZO Svazarmu.

Čeští programátoři dobývají svět?

Ota Luňák, Petr Potužník

I když jsme bohužel v posledních letech zaostali v mnoha oblastech za ostatním světem, neznamena to, že je to neschopností odborných kádrů. Oblast výpočetní techniky je poznamenána brutálním přístupem byrokratických řídicích struktur. Nedostatek osobních počítačů, odborné literatury a dalších faktorů nás poznamenal na dlouhou dobu. I v těchto neutěšených poměrech však vznikají programy na světové úrovni, a to jediné díky titánskému úsilí jedinců, ochotných věnovat výpočetní technice nejen všechny svůj volný čas, ale i nemalé finanční prostředky. Pokud budeme schopni vytvořit pro všechny nadané jedince odpovídající podmínky, snad budeme opět moci hovořit o "zlatých českých ručičkách" i v oblasti programového vybavení. Dokladem umu našich programátorů není pouze původní český textový procesor TEXT 602 z dílny 602. ZO Svazarmu (pomocí kterého tvořím i tento příspěvek), ale i následující řádky, které pro vás připravil Petr Potužník, nadějný programátor klubu uživatelů počítačů Amstrad 602. ZO Svazarmu. Jeho programy jsou tak dobré, že zaujaly redakci západoněmeckého časopisu Schneider Magazin. Program "Rubikova superkostka" získal dokonce titul "program měsíce". Nechme však hovořit Petra Potužníka.

"Jak se mi to podařilo?"

Nevím. Sám jsem byl překvapen. V lednovém čísle časopisu Schneider Magazin se v rubrice Tip des Monats objevilo jméno Petr Potužník a hra Super Rubik's Cube.

Je mi 20 let, studuji II. ročník fakulty všeobecného lékařství University Karlovy a programování je pouze můj koníček. Před třemi lety jsem si koupil počítač AMSTRAD CPC 6128 a od té doby jsem napsal už slušnou řádku programů. Některé byly dobré, jiné horší. Mezi kvalitnější patřil i program Super Rubik's Cube, který jsem poslal do redakce časopisu Schneider Magazin. Program simuloval známou Rubikovu kostku, ale místo obvyklého rozměru 3x3x3 umožňoval použít zvolený rozsah NxNxN. Jediným omezením byla rozlišovací schopnost monitoru. Program používá minimálně 6 barev (pro každou stranu kostky jednu) a na CPC zobrazuje tedy v modu 0 s rozlišením 160x200. V rozsahu 6x6x6 vypadá obrázek zcela přirozeně, ale dá se přemýšlet i nad 12x12x12. Kostka je zobrazena axonometricky s pomocí tří zrcadel, které ukazují odvrácené strany.

Výpočetní technice jsem se začal věnovat v 7. třídě; pracoval jsem s kalkulačkou Texas Instruments 58. Na gymnasiu jsem se naučil Basic počítačů Sharp PC 1521 a Sord M5. Pak přišel Amstrad CPC 6128 se systémem CP/M a Turbo Pascalem. Nyní se věnuji assembleru a jako pomocná vědecká síla na fakultě výpočetní techniky spolupracuji při tvorbě aplikačního softwaru a expertních systémů pro lékařství. Jsem aktivním členem Amstrad klubu 602. ZO SVAZARMU, který nabízí členům všechny dostupné informace o počítačích Amstrad / Schneider řady CPC."

Potud tedy Petr. Neodpustím si (po zkušenosti ze střediska VTI Svazarmu) však ještě poznámku: Petr ovládá němčinu a angličtinu, což je jistě jeden ze základních stavebních kamenů jeho umu. Co vy, spoléháte se pouze na to, co nám nabízí ministerstvo školství, nebo se snažíte i sami?

Z dílny Petra Potužníka

E I N S T E I N - T E S T P A M Ě T I

Na monitoru se objeví jednociferné číslo, které si máte zapamatovat. Po určité době číslo zmizí a objeví se otazník. Nyní je Vaším úkolem číslo zadat. V případě, že jste zadal jiné číslo, objeví se na monitoru výsledek - počet čísel, které jste si zapamatoval a hra skončí. Po zadání správného čísla se na monitoru objeví dvouciferné číslo - 1. číslice zůstává stejná, druhá je nová. Při bezchybném postupu se podobným způsobem v každém kole počet čísel zvětšuje o jednu; přibude vždy poslední číslice. Hra končí v okamžiku, kdy uděláte chybu.

Můj osobní rekord je dvacetičtyřmístné číslo, ale po pravidelném tréninku by se jistě dalo dosáhnout lepších výsledků.

Dobu, po kterou je číslo zobrazeno lze nastavit konstantou CAS na začátku programu. Časový interval by měl být v rozmezí mezi dvěma až pěti sekundami.

```
10 cas=2000
20 c$=HEX$(INT(RND*8)+1)
30 CLS:PEN 2:PRINT c$:PEN 1
40 FOR i=1 TO cas:NEXT
50 CLS:INPUT b$
60 IF b$<>c$ THEN 90
70 c$=c$+HEX$(INT(RND*9.9))
80 GOTO 30
90 PRINT:PRINT"Chyba!"
100 PRINT"Spravne cislo:";:PEN 2
110 PRINT c$:PEN 1:PRINT
120 PRINT "Zapamatoval jste si";LEN(c$)-1;"cisel":
    PRINT
```

P A T N Á C T K A

(Program byl otištěn pod názvem VERSCHIEBEFIX v časopise Amstrad International 4 / 1989 v rubrice 100,- DM fuer 1 kByte)

Jde o počítačovou simulaci hlavolamu "15". Program pracuje v rastru 640x400 bodů, při přenosu na jiné počítače je třeba u grafických příkazů upravit parametry. Tento program byl napsán tak, aby nepřesáhl kapacitu 1 kB; proto je ho třeba brát spíš jen jako námět nebo nápad pro další zpracování.

"Patnáctka" se ovládá cursorovými tlačítky.

```
10 CLS
20 DIM a(4,4)
30 FOR i=0 TO 4
40 MOVE 200,100+i*50:DRAWR 200,0
```

```

50 MOVE 200+i*50,100:DRAWR 0,200
60 NEXT i
70 FOR i=1 TO 4:FOR j=0 TO 3
80 a(i,4-j)=i+j*4
90 NEXT j,i
100 xp=4:yp=1
110 LOCATE 14,2:PRINT "Zamichani"
120 FOR i=1 TO 200:b=INT(RND*3.99)+1:
  GOSUB 200:NEXT i
130 LOCATE 14,2:PRINT CHR$(18)
140 FOR x=1 TO 4:FOR y=1 TO 4:GOSUB 230:NEXT y,x
150 a$=INKEY$
160 IF a$="" THEN 150
170 b=ASC(a$)-239
180 IF b<1 OR b>4 THEN 150
190 GOSUB 200:GOSUB 230:GOTO 150
200 IF (xp=1)*(b=4)+(xp=4)*(b=3)+(yp=1)*(b=1)
  +(yp=4)*(b=2) THEN 270
210 x=xp:y=yp:xp=xp+(4=b)-(3=b):yp=yp+(b=1)-(b=2):
  a(x,y)=a(xp,yp)
220 RETURN
230 TAG:MOVE 212+(x-1)*50,132+(y-1)*50
240 PRINT USING "##";a(x,y);
250 MOVE 212+(xp-1)*50,132+(yp-1)*50
260 PRINT " ";:TAGOFF
270 RETURN

```

Podprogram pro tvorbu MENU

Vlastní podprogram začíná od řádku 100. Před ním je demonstrace funkce podprogramu.

```

10 REM ** MENU (demo) **
20 PRINT:RESTORE:GOSUB 100
30 ON x GOTO 40,50,60,70,80
40 PRINT" Kresli graf":GOTO 10
50 PRINT" Vypocet parametru":GOTO 10
60 PRINT" Vypocet tabulky":GOTO 10
70 PRINT" Zapis na disk":GOTO 10
80 PRINT" Konec"
90 END
100 REM ** MENU (podprogram) **
110 READ a
120 FOR x=1 TO a
130 READ a$
140 PRINT x;"..... ";a$
150 NEXT x
160 PRINT:PRINT "Stisknete zvolenou klavesu!":
  PRINT
170 a$=INKEY$
180 IF a$="" THEN 170
190 x=VAL(a$)
200 IF x<1 OR x>a THEN 170
210 RETURN
220 DATA 5,"Kresli graf","Vypocet parametru",
  "Vypocet tabulky","Zapis na disk","Konec"

```

Seznam známých POKE pro AMSTRAD CPC

ANTIRIAD	- poke &57E8,0 - poke &57D4,0 - poke &57D5,0	EXOLON	- memory &24FF - load"exolon.nol",&2500 - poke &252B,počet životů - save"exolon.n01",b,&2500,&714D
GHOST'N GOBLINS	- poke &8282,0 - poke &5096,0 - poke &8564,0 - poke &8565,0 - call &5000	SPLIT PERSONAL.	- mode 0:border 0:for x=0 to 15: ink x,0:next - load"split" - poke &EF23,&B7 - call &C600
SCOOBY DOO	- load"scooby",1500 - poke &6595,0 - save"scooby",b,1500,34866	SIGMA SEVEN	- poke &8680,počet životů
1942	- poke 10117,x	NOMAD	- poke &223A,0
3D FIGHT	- poke &13FE,0	MONTY THE RUN	- poke &9CB9,0 - poke &9CC3,0
3D STARFIGHTER	-10 For I= &BE00 to &BE0A:READ A: POKE I,A:NEXT 20 DATA &CD, &7A, &BC, &AF, &32, &B4, &38, &32, &2C, &3D, &C9 - LOAD"STARFIG" - POKE &01F8,0:POKE &01F9,&BE:RUN	LIGHT FORCE	- poke &79DA,0
ANDROID ONE	- poke &7391,0	SUPER ROBIN	- poke &6C91,0
GREEN BERET	- poke &25F1,0 - poke &25F2,0	COMMANDO	- memory 9999 - load"com2",&2710 - poke &3B34,počet životů - save"com2",b,&2710,&5974
GUN FRIGHT	- poke &0EE6-9,0 - poke &122A,2	SABRE WULF	- memory &2FFF - load"sabprg.sbf",&3000 - poke &43C3,počet životů - save"sabprg.sbf",b,&3000,&3000
ZORRO	- poke &573D,0	CAULDRON	- memory &2FFF - load"caul3.bin",&3000 - poke &424D,0 - save"caul3.bin",b,&3000,&3100
COMMANDO	- poke &14BF,&FE		
FANT. VOYAGE	- poke &7E1C,0		

Program tvoří program

Ota Luňák

V poslední době se stává šlágr ve světě technika CASE, což znamená Computer Aided Software Engineering, neboli tvorba programů pomocí počítače. Smyslem tohoto směru vytváření programového vybavení je umožnit tvorbu programových produktů i neprogramátorům.

Následující program je tedy pro pohodlnější jedince (mezi které počítám i sebe) z řad uživatelů počítačů AMSTRAD CPC, protože sám vygeneruje malý program, který vám umožní vybrat si a "naladovat" jakýkoliv program na disku. Je omezen maximálním počtem 5ti programů, které lze zobrazit v jídelníčku. Lze však volat další program MENU, kde nadefinujeme případně další programy. Běžně se však na jednu stranu disku 3" neuloží více, jak oněch 5 programů (mám na mysli hry o celkové velikosti 35 kilobajtů - některé však zaberou i více). Program lze upravit tak, že výsledný program MENU zabere pouhý 1 kilobajt (nyní zabere 3 Kb) - to již ponechám na vaší libosti. Pokud ve výpisu najdete tento znak: ^, znamená to, že v místě jeho umístění ve výpisu programu musíte vložit sekvenci kláves CTRL a X, která zahajuje a ukončuje inverzní zobrazení (také CHR\$(24)).

Některé podprogramy můžete opět využít ve svých kracích. Zajímavý je tisk zobrazení znaků s dvojnásobnou výškou, načtení seznamu souborů na disku přímo z bufferu do pole proměnných a i vlastní mechanismus "vytvoření" programu zápisem na disk. Program MENU lze ovládat i pomocí ovladače, jinak také zvaného joystick.

GENERÁTOR MENU

```
100 'Menu Generator V1.0 O.Lunak
105 DEFINT c-z:SYMBOL AFTER 256
110 MODE 2:INK 0,0:INK 1,18:BORDER 0:SPEED KEY 40,4
120 DIM a$(65),b$(65),kilo(65),x(66),y(66)
130 OPENOUT"d":MEMORY HIMEM-1:CLOSEOUT
150 oldi=1:i=1:vy$=""
160 GOSUB 6000:GOSUB 5000:GOSUB 9000
200 p$="M E N U   G E N E R A T O R"
210 LOCATE 22,7:GOSUB 5500
220 p$=CHR$(164)+" O.LUNAK":LOCATE 36,12:PRINT p$
1000 REM main loop
1010 WHILE 1
1020 IF INKEY(8)=0 THEN oldi=i:i=i-1:GOSUB 4000
1040 IF INKEY(1)=0 THEN oldi=i:i=i+1:GOSUB 4000
1060 IF INKEY(0)=0 THEN oldi=i:i=i-6:GOSUB 4000
1080 IF INKEY(2)=0 THEN oldi=i:i=i+6:GOSUB 4000
1100 IF INKEY(9)=0 THEN kilo(i)=0:inflag=1:ii=0
1120 IF (INKEY(79)=0 AND kilo(i)=0) THEN kilo(i)=1:
inflag=0
1160 IF INKEY(54)=128 THEN POKE &A700,1:LOCATE#1,72,2:
PRINT#1,"B-->";MID$(vy$,1,1):
1170 IF INKEY(69)=128 THEN POKE &A700,0:LOCATE#1,72,2:
PRINT#1,"A-->";MID$(vy$,1,1):
1180 LOCATE#1,5,2:PRINT#1,"Drive:";CHR$(PEEK(&A700)+65):
LOCATE#1,72,2:PRINT#1,CHR$(PEEK(&A700)+65);"-->";
MID$(vy$,1,1):
1190 LOCATE #1,35,2:PRINT#1,"Ctrl+Copy/Del"
1200 IF INKEY(62)=128 THEN GOSUB 3000
1220 IF INKEY(52)=128 THEN GOSUB 8000:GOSUB 10000
1260 IF INKEY(58)=128 THEN END
1900 WEND
3000 REM CAT --> a$(i)
3020 CLS#0:LOCATE #1,35,2:PRINT#1,"Please wait..."
3040 POKE &BDD9,&C9:CAT:POKE &BDD9,&C3
3060 IF PEEK(&A702)=1 THEN files=PEEK(&A922)-2 ELSE
files=PEEK(&A912)-1
```

```
3080 buffer=PEEK(&A79C)*256+PEEK(&A79B)
3100 FOR i=1 TO files
3120 a$(i)=STRING$(15," ")
3140 FOR z=1 TO 14
3160 MID$(a$(i),z)=CHR$(PEEK(buffer))
3180 IF CHR$(PEEK(buffer)) < " " THEN MID$(a$(i),z)=" "
3200 IF z=13 THEN kilo(i)=PEEK(buffer)
3220 buffer=buffer+1
3240 NEXT z
3260 a$(i)=MID$(a$(i),2,8)+"."+MID$(a$(i),10,3)
3280 IF a$(i)=" " THEN files=files-1
3300 NEXT i
3320 FOR i=1 TO files
3340 LOCATE x(i),y(i):PRINT a$(i)
3360 IF (kilo(i)=0 OR i=1) THEN GOSUB 4100
3380 NEXT i
3390 i=1:REM first file after cat
3400 GOSUB 8000
4000 REM print inverze
4020 IF i<1 THEN i=files
4040 IF i>files THEN i=1
4050 IF kilo(oldi)=0 THEN inflag=1
4060 IF inflag=1 THEN GOTO 4100
4080 LOCATE x(oldi),y(oldi):PRINT a$(oldi)
4100 LOCATE x(i),y(i):PRINT " "+a$(i)+" "
4110 LOCATE (x(oldi)-1),y(oldi):PRINT " ":
LOCATE (x(i)-1),y(i):PRINT">"
4120 inflag=0
4140 RETURN
5000 REM coordinates
5020 a=1:b=6
5040 FOR n=1 TO 11
5060 RESTORE 5200
5080 FOR m=a TO b
5100 y(m)=n
5120 READ x(m)
5140 NEXT m
5160 a=a+6:b=b+6
5180 NEXT n
5200 DATA 2,15,28,41,54,67,79
5220 RETURN
5500 REM high text
5510 SYMBOL AFTER 32: FOR a=1 TO LEN(p$):
b=ASC(MID$(p$,a,1)):adr=(b-32)*8+1+HIMEM
5520 FOR s=0 TO 7:s(s)=PEEK(adr+s):NEXT s
5530 SYMBOL 254,s(0),s(0),s(1),s(1),s(2),s(2),s(3),s(3)
5540 SYMBOL 255,s(4),s(4),s(5),s(5),s(6),s(6),s(7),s(7)
5550 PRINT CHR$(254)CHR$(10)CHR$(8)CHR$(255)CHR$(11):NEXT a
5555 RETURN
6000 REM windows
6020 WINDOW #0,2,79,2,16:CLS#0
6040 WINDOW #1,1,80,18,20:PEN#1,0:PAPER#1,1:CLS#1
6060 WINDOW #2,2,79,22,24:CLS#2
6070 ORIGIN 0,0:DRAWR 0,399,1:DRAWR 639,0,1:DRAWR 0,-399,1:
DRAWR -639,0,1:
6080 ORIGIN 0,76:DRAWR 639,0,1
6090 ORIGIN 0,130:DRAWR 639,0,1
6100 REM ORIGIN 1,82:DRAWR 636,0,0:DRAWR 0,47,0:
DRAWR -635,0,0:DRAWR 0,-47,0
6200 ORIGIN 2,2:DRAWR 634,0:DRAWR 0,397:DRAWR 1,0:
DRAWR 0,-397
6900 RETURN
8000 REM cls #1
8010 LOCATE#1,32,2:PRINT#1," "
8100 RETURN
9000 REM cls #2
9010 CLS#2
9030 LOCATE#2,20,2:PRINT#2,"CTRL+....`A`/`B`....`C`atalog
....`G`enerator....`E`nd"
9500 RETURN
```

```

9530 CLS#2:LOCATE#2,17,2:PRINT#2,"CTRL+....";
CHR$(PEEK(&A700)+65);"-->"A"....";
CHR$(PEEK(&A700)+65);"-->"B"...."Save program
.... To menu"
9600 RETURN
10000 REM generator
10020 CLS#2
10030 GOSUB 9530
10040 WHILE 1
10050 IF INKEY(69)=128 THEN vy$="A":LOCATE#1,72,2:PRINT#1,
CHR$(PEEK(&A700)+65);"-->"A"
10060 IF INKEY(54)=128 THEN vy$="B":LOCATE#1,72,2:PRINT#1,
CHR$(PEEK(&A700)+65);"-->"B"
10070 IF INKEY(51)=128 THEN GOSUB 9000:GOTO 10900
10080 IF INKEY(60)=128 THEN GOSUB 12000
10090 LOCATE#1,72,2:PRINT#1,CHR$(PEEK(&A700)+65);"-->";
MID$(vy$,1,1)
10100 WEND
10900 RETURN
12000 REM main loop
12020 kde=9:ii=0:CLEAR INPUT:CLOSEIN:CLOSEOUT
12030 IF MID$(vy$,1,1)="?" THEN vy$="A:"
12031 ven$=vy$+"MENU.BAS":OPENOUT ven$
12040 FOR i=1 TO files:IF kilo(i)<>0 THEN ii=ii+1:GOTO 12100
12060 GOSUB 8000:LOCATE#1,34,2:PRINT#1,a$(i)
12070 CLS#2
12080 LOCATE#2,10,2:INPUT#2,"Program name (11 chars):";b$(i)
:b$(i)=UPPER$(b$(i))
12085 IF LEN(b$(i))>11 THEN 12080
12090 b$(i)=MID$(b$(i),1,11):
b$(i)=b$(i)+STRING$(11-LEN(b$(i))," ")
12100 NEXT i
12110 IF ii=files THEN CLS:PRINT"NO FILE SELECTED!":
GOTO 15810
12120 GOSUB 9530:GOSUB 8000:CLS:LOCATE 25,10:
PRINT"SAVING PROGRAM MENU.BAS....."
12200 RESTORE 16000
12201 x$(1)="M E N U ":x$(2)="Select: "+CHR$(240)+" "+CHR$(
241)+" + [ENTER]": x$(3)="Ctrl+A/B.....Drive:":
x$(4)=" O.LUNAK "
12205 prg$="10 a$(1)="+CHR$(34)+x$(1)+CHR$(34)+"a$(2)="+
CHR$(34)+x$(2)+CHR$(34)+"a$(3)="+CHR$(34)+x$(3)+CHR$(
34)
12206 PRINT#9,prg$
12207 prg$="12 a$(4)="+CHR$(34)+CHR$(164)+x$(4)+CHR$(34):
PRINT#9,prg$
12210 FOR i=1 TO 46
12220 READ prg$
12230 PRINT#9,prg$
12240 NEXT
12400 FOR i=1 TO files:IF kilo(i)<>0 THEN 12500
12410 prg$=STR$(5100+i)+" DATA"+CHR$(34)+b$(i)+CHR$(34)+" , "
+CHR$(34)+a$(i)+CHR$(34)
12420 PRINT#9,prg$:xx=xx+1:IF xx=5 THEN CLS:PRINT"FIVE
NAMES USED ONLY!":GOTO 12700
12500 NEXT i
12550 FOR i=1 TO (5-xx)
12600 prg$=STR$(5200+i)+" DATA"+CHR$(34)+STRING$(11,32)+
CHR$(34)+" , "+CHR$(34)+STRING$(12,32)+CHR$(34)
12620 PRINT#9,prg$
12640 NEXT i
12700 prg$="5300 DATA"+CHR$(34)+"Other menu "+CHR$(34)+
CHR$(44)+CHR$(34)+"MENU .BAS"+CHR$(34)
12710 PRINT#9,prg$
15600 CLOSEOUT
15700 CLS:PRINT"PROGRAM MENU.BAS INSTALLED ON DISK

```

```

DRIVE ";vy$
15800 IF ii=files THEN CLS:PRINT"NO FILE SELECTED!"
15810 GOSUB 9530:GOSUB 8000
15900 RETURN
16000 REM data for program Menu.bas
16100 DATA "14 PEN 1:IF PEEK(&8000)=255 GOTO 100"
16120 DATA "16 SYMBOL AFTER 32:MODE 1"
16140 DATA "18 INK 0,0:INK 1,18:INK 2,6:BORDER 0"
16180 DATA "20 LOCATE 17,2:GOSUB 1000"
16200 DATA "30 LOCATE 11,22:PRINT a$(2)"
16220 DATA "40 LOCATE 11,24:PRINT a$(3)"
16240 DATA "100 RESTORE:EVERY 150,1 GOSUB 4200"
16260 DATA "120 LOCATE 16,8:FOR a=1 TO 6:READ j$(a),r$(a)"
16280 DATA "140 PRINT TAB(16) j$(a)"
16300 DATA "180 PRINT"
16320 DATA "200 NEXT a"
16480 DATA "220 GOSUB 3000:GOSUB 4400"
16500 DATA "250 a=1:x=16:y=8:LOCATE x,y:PRINT chr$(24)+
j$(a)+chr$(24)"
16520 DATA "260 WHILE INKEY(18)<>0 AND JOY(0)<>16"
16540 DATA "270 IF (INKEY(0)=0 OR JOY(0)=1) AND a<=1 THEN
olda=a:oldy=y:a=6:y=18:GOSUB 4000"
16560 DATA "275 IF (INKEY(0)=0 OR JOY(0)=1) AND a<=6 THEN
olda=a:oldy=y:a=a-1:y=y-2:GOSUB 4000"
16580 DATA "280 IF (INKEY(2)=0 OR JOY(0)=2) AND a>=6 THEN
olda=a:oldy=y:a=1:y=8:GOSUB 4000"
16600 DATA "285 IF (INKEY(2)=0 OR JOY(0)=2) AND a>=1 THEN
olda=a:oldy=y:a=a+1:y=y+2:GOSUB 4000"
16620 DATA "290 IF INKEY(54)=128 THEN POKE &A700,1:GOSUB
4400"
16640 DATA "295 IF INKEY(69)=128 THEN POKE &A700,0:GOSUB
4400"
16680 DATA "335 WEND"
16700 DATA "340 IF j$(a)=space$(11) THEN GOTO 260"
16720 DATA "350 PEN 2:GOSUB 4060:"
16740 DATA "400 SYMBOL AFTER 240"
16760 DATA "800 POKE &8000,255: RUN r$(a)"
16780 DATA "900 END"
16800 DATA "1000 REM high text"
16820 DATA "1020 FOR a=1 TO LEN(a$(1)):
b=ASC(MID$(a$(1),a,1)):adr=(b-32)*8+1+HINEM"
16840 DATA "1040 FOR s=0 TO 7:s(s)=PEEK(adr+s):NEXT s"
16860 DATA "1060 SYMBOL 254,s(0),s(0),s(1),s(1),s(2),s(2),
s(3),s(3)"
16880 DATA "1080 SYMBOL 255,s(4),s(4),s(5),s(5),s(6),s(6),
s(7),s(7)"
16900 DATA "1100 PRINT CHR$(254)CHR$(10)CHR$(8)CHR$(255)
CHR$(11);:NEXT a"
16920 DATA "1120 RETURN"
16940 DATA "3000 ORIGIN 220,310:DRAWR 210,0:DRAWR 0,-210:
DRAWR -210,0:DRAWR 0,210"
16960 DATA "3010 MOVER 213,-2:DRAWR 0,-210:DRAWR -209,0"
16980 DATA "3020 MOVER 2,-1:DRAWR 209,0:DRAWR 0,208:
ORIGIN 300,110"
17000 DATA "3500 RETURN"
17020 DATA "4000 REM inverzion"
17040 DATA "4010 IF a=7 THEN a=1:y=8"
17060 DATA "4020 IF a<1 THEN a=6:y=18"
17080 DATA "4050 LOCATE x,oldy:PRINT j$(olda)"
17100 DATA "4060 LOCATE x,y:PRINT chr$(24)+j$(a)+chr$(24)"
17120 DATA "4100 RETURN"
17140 DATA "4200 IF TEST(2,2)=1 THEN a$(0)=chr$(24)+a$(4)
+chr$(24) ELSE a$(0)=a$(4)"
17160 DATA "4300 LOCATE 16,18:PRINT a$(0):RETURN"
17180 DATA "4400 LOCATE 31,24:PRINT CHR$(PEEK(&A700)+65):
RETURN"

```

Z E S V Ě T A

ZX Spectrum+3

Je posledním vývojovým stupněm v řadě počítačů ZX. +3 vznikl po zakoupení firmy Sinclair firmou Amstrad a je proto poznamenán konstrukční filozofií techniků fy Amstrad. Tak bylo doplněno

Spectrum o vestavěnou diskovou jednotku 3" a o možnost pracovat pod OS CP/M 2.2 a 3.1. Také se zmenšil počet integrovaných obvodů "uvnitř", avšak při rozšíření počtu vestavěných rozhraní. +3 disponuje nejen rozhraním RS232 a Centronics, ale i výstupem standardu MIDI.

21

Před časem jsme našli v redakční poště zajímavý příspěvek od našeho čtenáře z Kralup. I když obvykle neuveřejňujeme tak rozsáhlé návody k programům, tentokrát děláme výjimku. Činíme tak nejen vzhledem k závažnosti materiálu, ale také kvůli tomu, že máme zas jednou možnost přetrumfnout nejen populární "Tři švestky" ale i Karolínku a řadu jiných. Po řádném prostudování výhod popisovaného programu, můžeme jen doufat, že se brzy objeví i jeho česká verze D-Basic. Velice se za to přimlouváme. Všechny, kteří se pustí do experimentů s tímto programem, upozorňujeme na to, že k používání příkazu GIGA bude asi mít připomínky EZU.

REDAKCE MIKROBAZE
602. ZO SVAZARMU
Wintrova 8
160 41 PRAHA 6

Kralupy/Vlt 13.4.1989

Vážení přátelé,

před časem jsem objevil v časopise "BAJTEK" článek popisující FINAL BASIC pro COMMODORE C-64. Vlastnosti této verze BASICu mě zaujaly. Protože Sinclair ZX SPECTRUM má k dispozici velké programové vybavení, začal jsem pátrat, zda neexistuje podobná verze i pro Spectrum. Samozřejmě že existuje, a dokonce dokonalejší (jak jinak).

Stručný přehled příkazů TERA BASICu Vám zasilám. Jedná se o tzv. nevyžádaný tiskárnopis (netřeba vracet).

Prosba na závěr: pokud se Vám podaří zjistit další detaily, týkající se tohoto super jazyka, rád bych se s nimi seznámil.

Zdeněk ŠPIČKA
V zátiší 1 kB
278 01 KRALUPY/VLT

TERABASIC MACROSOFT (C) 1988

Pro ZX Spectrum je k dispozici mnoho různých rozšířených interpretrů BASICu. Dobře známé jsou např. Beta BASIC, Mega-BASIC, Sky Line BASIC, nebo Laser BASIC. Nyní se na trhu objevuje Tera BASIC firmy Macrosoft.

Hned v úvodu je nutné konstatovat - Tera BASIC je něco kvalitativně nového. Něco, o čem každý Spectrista vždycky snil. Uživatel této verze BASICu získává prakticky neomezené možnosti.

Následující text seznamuje jen s některými rozšiřujícími příkazy a funkcemi jazyka. Úplný manuál údajně připravuje 666. ZO Svazarmu Praha. Vše by prý mělo být hotovo, ovšem jsou problémy s tiskárnou.

Uvedeme si něco málo přes 30 nových příkazů Tera BASICU. Domnívám se, že komise ISO, pracující na návrhu normy BASICu, by je při své práci neměla přehlédnout.

Na závěr už snad jen to, že reklamní slogan, jímž MACROSOFT uvedl svůj nový softwarový produkt na trh:

T O S E E T E R A - T O S H E D T E A R

je v tomto případě zcela namístě. Následující poznámky se týkají verze 2.0. Prosím jazykové puristy, aby tolerovali přepis anglické výslovnosti, daný omezeními D-TEXTu.

ALTER TO a\$ [óltr tu] (změň na)

Po provedení příkazu se Spectrum změní v emulátor počítače, jehož zkratka je uvedena jako řetězcový parametr. V některých případech musí tomuto příkazu předcházet příkaz GIGA. Ve verzi 2.0 jsou přípustné parametry - IBM, APL, ATR, TRS, TIN, HNW, CAS, SHP a PMD.

AND/OR [end/ó] (a/nebo)

Logická funkce. Příklad: výraz X AND/OR Y Z - je vyhodnocen: je-li pravdivý výraz X AND/OR Y, pak je TRUE pravděpodobné i Z, ale ne vždycky.

ARGUM () ['ágjum] (argument)

Funkce nemá parametr, avšak vrátí uživateli dostatečné množství argumentů zdůvodňujících, proč je jeho počítač lepší než kolegův. Argumenty jsou použitelné i při diskusích s rodiči, manželkou ap.

BANK a\$,n [bank] (banka)

Příkaz použitelný jen s tiskárnou. Umožňuje vytisknout bankovku libovolné měny v hodnotě 10..9999. (Ovšem pozor na

nestandardní numerické hodnoty). Řetězcový parametr obsahuje znak příslušné měny - např.: \$, f, DM ap. Kčs bohužel příkaz neakceptuje (chybové hlášení 47 Invalid currency). Nejlepší výsledky byly dosaženy s tiskárnou LASERWRITER.

BUT [bat] (ale)

Operátor predikátové logiky. Např.: A BUT B POSS C, znamená - platí-li A ale B není FALSE (alespoň pravděpodobné), pak je B mnohem větší než součet A+B. Výraz je často používán společně s dotazovými funkcemi WHY (proč) a HOW (jak) např. v konstrukcích typu BUT WHY nebo BUT HOW.

BWCOL [bi ui kol] (černobilá->barva)

Příkaz, který přeneše obsah obrazovky na tiskárnu. Hardcopy přeneše obsah display file přes černou pásku mozaikové tiskárny barevně. Bohužel je nutný speciální papír fy Polaroid, který není nejlevnější.

CALLCLOCK () [kól klok] (doslova volej hodiny, čas)

Funkce vrátí argumenty potvrzující skutečnost, že čas strávený u počítače, místo při domácích pracech, je zanedbatelný vzhledem ke stáří vesmíru. Viz také ARGUM.

CUCKOO	[kukú]	(kukačka)	HOW	[hau]	(jak)
Příkaz je určen k použití při odladování programů (i ve strojovém kódu). Nalezne překlepy i některé logické chyby programátora. Při použití spolu s ARGUM vrací odůvodnění proč k chybě došlo.			Příkaz často používaný začátečníky. Viz také WHY.		
DEATHMAN a\$	[detmen]	(mrtvý muž, mrtvola)	HURRYUP	[hari ap]	(pospíchej)
Funkce vrací seznam příznaků nemoci, jejíž latinský název je řetězcovým parametrem funkce. Lze použít při návštěvě OÚNZ. Příliš časté používání tohoto příkazu však může mít za následek těžké poudrazové stavy. Např. LPRINT DEATHMAN "bufoviridis" ap.			Zrychluje práci s disketovou jednotkou typicky 12x. Funguje přímo se systémy DISCIPLÉ, OPUS, KEMPSTON a BETA. Zrychluje práci i s mikrodrajvy, ale jen asi 8x. Příklad pro použití s KEMPSTON 2.0:		
DO ... DO	[dú ... dú]	(délej ... délej)	10 PRINT #4: LOAD "Název": PRINT 1: HURRYUP		
Varianta smyčky FOR ... NEXT. Usnadňuje práci v reálném čase. Např.: DO NOW DO TOMORROW.			Vedlejším efektem příkazu je tišší chod disketové mechaniky.		
DBLSCREEN	[dabl skrin]	(dvojitá obrazovka)	KICK n,n	[kik]	(kopni)
Zvýší rozlišitelnost obrazovky (něco jako HIRES). Místo normální rozlišitelnosti 256+192 bodů máte po použití příkazu k dispozici rastr 384x256. Je nutné postavit dvě televize na bok (těsně vedle sebe).			Obdoba POKE. Umožňuje "poukovat" v dolních 16 kB.		
DREAM (...)	[drim]	(sen)	LISTEN	[lisn]	(poslouchej)
Nejuniverzálnější příkaz - na místě teček napíšete své tajné přání. Je uloženo do ULA a splněno v době do pěti let. Lze použít i místo kteréhokoliv příkazu BASICu (i vysněného). Zápis přání, které by někomu mohlo způsobit škodu, má za následek zastavení programu s chybovou zprávou "51 Dream error CONT no repeats".			Umožňuje zadávání příkazů hlasem (voice input). Nezbytné jsou následující hardwarové doplňky: Kurzweilův interfejs, dva citlivé mikrofony (nejlépe SHURE) a předzesilovač o výkonu cca 2 kW.		
EMU a\$	[emjuleit]	(napodobit)	MAYBE proměnná	[meibi]	(možná)
A\$ je čtyřznakový řetězec - viz dále. Umožňuje uživateli přepnout Spectrum na emulátor domácího zařízení. Ve verzi 2.0 jsou přípustné následující parametry: BLDR (pračka), MCWV (mikrovlnná trouba), RFGR (lednička), STOV (sporák), TLPH (telefon), TSTR (opékač topínek) a SHVR (holící strojek).			Příkaz podobající se LET, ale mnohem pružnější. Např.:		
EVER/BETTER	[evr/betr]	(vždy/lépe)	IF k>1 AND y=z THEN MAYBE a=b IF NOT THEN WHY NOT		
Používá se hlavně v konstrukci IF ... THEN. Např.:			MIMICRY a\$		
IF a EVER=5 THEN BETTER B=12			[mimikry]		
je-li A vždy rovno 5, pak by bylo nejlépe, aby B bylo rovno 12.			(mimikry)		
GIGA	[džaiga]	(giga)	Příkaz přeruší jakýkoliv běžící program (NMI) a na obrazovce se objeví odstrašující sloupce čísel. Doporučitelné bližší-li se nepovolaná osoba a vy se právě zabýváte programem typu SLIDE SHOW, STRIP DICE ap. Parametrem je jednoznakový řetězec definující klávesu, na jejíž stisknutí k popsané akci dojde. Klávesa funguje jako flip-flop.		
Jeden z velmi silných příkazů Tera BASICu. Funguje na verzi 16 KB i 48 KB (na 128 +2 s malým omezením). Příkaz zvýší kapacitu paměti na jeden Gigabajt. Je generován impuls, který má za následek reorganizaci struktury paměti RAM. POZOR, před provedením tohoto příkazu je nutné vyměnit elektrický rozvod ve zdi a jističe. Manuál také požaduje tmavé brýle a dokonalé uzemnění uživatele.			MORCOL n		
HISCORE	[hai skó]	(nejvyšší skóre)	[mókal]		
Malý dárek firmy MICROSOFT vášnivým hráčům her. Používá se ve spojení s IF ... THEN. Zajišťuje maximální bodový zisk v libovolné hře. Použijte před LOAD **. Např.:			(více barev)		
10 IF ja THEN HISCORE			Zvýší počet barev použitelných na Spectru na n*8 barev. Záporný parametr má za následek zprávu "73 Colour out of range".		
20 IF kolega THEN ZERO			PLEASE		
30 LOAD **			[plis]		
			(prosím)		
			Příkaz je doporučován k použití při spouštění obzvláště vzpurných softwarových produktů (zvláště vlastních), které chronicky odmítají "chodit". Lze použít zkratku PLS. Příklad:		
			PLS:RUN nebo i PLS: LOAD **		
			POSS		
			[,posi'bility]		
			(možnost)		
			Tento příkaz raději nepoužívejte. Vzhledem k chybě v ROM má za následek totální destrukci Z80.		
			RANDOMDUMP start,length		
			[rendomdamp]		
			(náhodně ulož)		
			Umožňuje přenos oblasti paměti, definované parametry start a délka do naprosto nepředvídatelné oblasti. Pokud bychom tento příkaz potřebovali a nemáme k dispozici Tera BACIS, můžeme použít následující konstrukce Sinclair BASICu:		
			10 LET n=RND*65536		

20 FOR i=start TO start+length-1
 30 POKE n, PEEK i: LET n=n+1
 40 NEXT i

tom, že příkaz blokuje výstup znechucující zprávy "R Tape loading error".

READMAG [ridmeg] (čti časopis)

Umožňuje přečíst program z libovolného počítačového časopisu přímo do paměti (DMA). Nutná je standardní syntaxe Sinclair BASICu (popř. rozšíření Tera), nicméně firma OCEAN pracuje na univerzálním překladači. Podmínkou správné funkce příkazu je složení stránky s programem na čtverec o délce strany 5 1/4" (nebo 3.5", máte-li náhon 1581).

SCRIB [skraib] (písař)

Univerzální příkaz, který uživateli umožňuje psát na obrazovku doslova čímkoli (pero, tužka, fix, štětec, olůvko ap.) Je nutný speciální interfejs.

SHUFFLE [šaf1] (míchat)

Jeden z nejdokonaleji naprogramovaných příkazů Tera BASICu! Současně jde o nejrychlejší známý algoritmus míchající setříděná data. Setříděné numerické pole o velikosti 30 kB dokáže dokonale promíchat za dobu kratší než $3 \cdot 10^{-7}$.

SIT [sit] (sedni)

Příkaz lze používat jen v přímém režimu (ne v programu). Z této kategorie příkazů uvedme ještě FETCH a\$ (přines), LIE DOWN (lehni) a SPEAK a\$ (mluv).

SLOAD a\$ [seifty loud] (bezpečné načtení)

Syntaxe příkazu je stejná jako u normálního LOAD. Rozdíl je v

SMOKE a\$,b\$ [smouk] (kouř, dým)

Použitelné k odehnání dotěrné návštěvy od vašeho počítače. Řetězový parametr má strukturu "hh:mm:ss". Po uplynutí času daného prvním parametrem, začne ze Spectra stoupat kouř a na obrazovce se objeví náhodné barvy a grafika. Po dobu danou druhým parametrem nelze počítač spustit (nefunguje ani RESET, nepomůže ani vypnutí počítače).

TRANSLATE a\$ [trens'leit] (přeložit)

Použijete-li příkaz před načtením programu, zajistí překlad výstupů textu na obrazovku (nebo tiskárnu) z angličtiny do jazyka, jehož zkratka je řetězovým parametrem příkazu. Autoři interpretru vtipně použili mezinárodních značek pro motorová vozidla. Bohužel verze, která je k dispozici, připouští pouze parametry H, D, I, J a GB. Poslední parametr se může zdát holou zbytečností, ale zjistil jsem, že výstup ošetřuje případné pravopisné chyby a převádí text do spisovné oxfordské angličtiny.

WHY [wai] (proč)

Univerzální instrukce. Lze použít samostatně nebo v kombinaci s ostatními logickými instrukcemi.

WCRTTV [dabl jú sí á tí tí ví] (monitor->televizor)

Pomocí fotoelektrické syntézy změní kvalitu zobrazení monitoru tak, že se téměř blíží kvalitě zobrazení televizoru (NTQ). Používá se často ve spojení s příkazy GIGA, DOUBLE a MORCOL.

---◇---

To byl tedy nejstručnější přehled možností, které získáte, stanete-li se vlastníkem programu Tera BASIC. Přehled nemůže ukázat všechny výhody této verze BASICu. Zkušenost ukázala, že i velmi zběhlým programátorům trvá nějakou chvíli, než dokážou dokonale využívat všech možností, které tato implementace poskytuje. Přehled se například také téměř nezmiňuje o rozšířených chybových hlášeních, kterých je spolu s původními skoro 200! Některá vznikla k ošetření možných chyb při použití nových příkazů - např. "111 Specs not present CONT repets" - "Nejsou brýle - CONTINUE opakuje" u příkazu GIGA. Další ošetřují obecnější chybové situace, např. "131 Finger error", chybně stisknutá klávesa, "164 Don't tipple too", mírněji řečeno - nepij tolik, "96 Insufficient power" - slabé napájení ap.

Lze očekávat, že tato verze BASICu přispěje svou hřivnou i v oblasti vývoje umělé inteligence. V některých případech tento programový systém první slibné náznaky projevuje. Mnoho zdaru s Tera BASICem přeje TOPSOFT.

Literatura:

1. TERA BASIC Owner's Handbook Macrosoft 4585 Scotts Valley CA 95066
2. COMPUTE! GAZETE 2/88
3. BAJTEK 6/88

 TRANSLATE & DATLING NOT (C) 1989 TOPSOFT KRALUPY/VLT

Zdeňkovi Špičkoví se omlouváme za pozdní zveřejnění článku, nemohli jsme se totiž rozhodnout, zda jeho správná adresa je v zátiší 1000, nebo 1024, protože jak je vidno z popisu příkazu GIGA, používá někdy malé k nejen ve významu 1000, ale i 1024.

Disketová jednotka pro IQ 151 a CP/M

Naše zkušenosti s prací disketové jednotky a operačního systému CP/M na IQ 151.

Miroslav Pilát

Do škol přichází postupně nová periférie pro IQ 151 - Disketová jednotka Consul 8". K počítači se připojuje pomocí modulu Disc 2. S touto periférií se školní počítač IQ 151 stává konečně nástrojem plně využitelným pro školní praxi. Poněkud však zaráží finanční náklady na její pořízení - mechanika + řadič přijde na 37 000,- Kčs. Školní orgány proto doporučují tuto periférii přednostně zařazovat pro práci v síti. Ty jsou bohužel ve školství doposud dvě a to Variel a Felnet. Naše škola má možnost zapojení jak do sítě tak i samostatně.

První problém, který musí uživatel řešit, je volba paměti EPROM. Státní podnik Komenium dodává totiž dvě EPROM a záleží na uživateli, kterou si vybere, zda Variel či Felnet. Nejlepší řešení je nainstalovat do dvou počítačů po jedné EPROM - instalace je velmi jednoduchá - a mít tak možnost vyzkoušet obě dvě verze operačního systému CP/M. Obě jsou upravené na jedné ČVUT - VOC a elektrofakultě! Naštěstí stačí jeden řadič. Dodávají ho Kancelářské stroje Teplice, které také montují mechaniku vyrobenou ve Zbrojovce Brno.

Proč píšou o dvou systémech, když se jedná o jeden operační systém CP/M. Systém VOC pracuje totiž jako CP/M, má základní možnosti CP/M jako je FORMAT, PW, Sysgen a pod. Autoři si však zjednodušili práci tím, že se při práci s programy v Basicu, kterých na školách není málo, předává řízení modulu Basic. Disková mechanika pak začne pracovat jako rychlý magnetofon, jenže příliš drahý. Práce je řízena uživatelsky: pomocí jednoduchého menu, ale pozor, při programování nelze samostatně nahrát data na disketu, pouze jako součást programu v datech. Chybí zde Basic, který by uměl spolupracovat s disketou, tj. ukládat data a číst data pomocí příkazů pro práci se soubory. Verze CP/M VOC to však neumí. Program pro výpočet výsledků závodu branné všestrannosti středních škol, který vznikl na naší škole, musel být přepracován pro druhou verzi CP/M Felnet, jak se zmíním dále. Po úpravě BIOSU lze provozovat na IQ 151 také program DBASE. Tuto úpravu nám velmi ochotně na VOC provedli.

Verzi VOC máme nainstalovanou na počítači v učebně branné výchovy a všechny programy pro výuku jsme si převedli na disketu programem IQBASIC.COM. Po jeho nahrání z diskety se zobrazí MENU:

Vyvolej soubor	: 0
Ulož soubor	: 1
Vypíš obsah	: 2
Vyvolej stroják	: 3
Ulož stroják	: 4
Basic	: 5
Monitor	: 6
Reset	: 7
Obnov Basic	: 8
Konec	: 9

Po zadání číslice a zmáčknutí CR je možno spustit požadovanou činnost. Je nutno upozornit na skutečnost, že programy nahrané při zasunutém modulu Basic 6 se musí spouštět zase jen při přítomnosti modulu Basic 6. Nelze kombinovat s modulem

Basic G. Při použití disketové jednotky a zapsání všech programů na disketu odpadá zdlouhavé nahrávání z magnetofonu, které je doprovázeno množstvím chyb a někdy i nutností několikerého nahrávání.

Bude na místě vzpomenout některé závady, které se vyskytly při uvádění disketové mechaniky do provozu. Nešla uzamykat dvířka, špatně šlo formátování, ozýval se hluk při roztočení diskety, upadl kryt dveří. Tyto závady bylo nutno řešit přes vedení KS Teplice s. Hozu. Firma zajistila v současné době opravy přes oblastní pobočku KS a tak je pomoc v případě poruch rychlá. Při závadě na řadiči zašlou KS Teplice obratem nový modul. Je nutno si předem ověřit, zda jde o vadný modul, mechaniku či počítač. Malé odbočení pro majitele prvních IQ 151 - pokud jste si náhodou nechali rozšířit paměť vašeho počítače na 64 Kb, můžete s tím mít potíže. Pro práci s disketou sice stačí 32 Kb, ale pro práci s DBASE je nutno 64 Kb. IQ 151 první verze sice pracuje jako 64 Kb, ale ne vždy je řadič schopen reagovat na staré obvodové řešení IQ 151. Díky pochopení ředitele KS Teplice ing. Nováka a týmu konstruktérů firmy se přišlo na problém neustálé výměny modulu DISK 2. Vyskytovaly se chyby ve formátování a funkci DBASE. Pro práci s disketovou jednotkou je nutno pracovat jen s počítačem IQ 151G (!), verze 32 nebo 64 Kb.

Pokud nejste postaveni při používání počítače před problém ukládání datových souborů na disketu, vystačíte se systémem Variel-VOC ČVUT Praha. Síť Variel montuje ELEKTROKOV Znojmo pod vedením obětavého pracovníka s. Hujňáka. Podrobný návod pro práci sítí Variel vydalo Komenium 1988 (autor M. Feil). Ten také napsal podrobný návod pro používání disketové jednotky.

Jste-li však náročnější uživatelé počítače IQ 151, musíte využít druhé EMPROMky (FELNET). Ta vám umožní využití firemních disket KS Teplice, jenž jsou přiloženy v dodávce Komenia. Na disketě je mimo jiné program DBASIC, který umí ukládat datové soubory na disketu. DBASIC si zachovává kompatibilitu s BASIC G. Programátoři si však musí dávat při práci pozor při zadávání příkazů pro ukládání a čtení datových souborů. Autor DBASIC nedodržel standard CP/M a pro ukládání dat zvolil pouze 80 znaků místo 256, jak je standard CP/M. Při psaní programů je nutno pracovat pouze s jednou mechanikou a jednou disketou, jinak se hlásí jako disk určený jen pro čtení.

Přes tyto drobné nedostatky je nutno konstatovat, že DBASIC plně vyhovuje pro spolupráci diskety s magnetofonem. Ostatní programy jsou standardem v oblasti CP/M (FORMAT, PW, Sysgen atd.). Nelze však zapomenout na základní podmínku pro použití FELNETu, což je bezpodmínečně IQ 151G s pamětí 64 Kb, jak v síti, tak i samostatně. Podrobnosti všem zájemcům poskytne ing. Bílý FE ČVUT Praha. Felnet umí také pracovat s programem DBASE.

Naše dosavadní zkušenosti s provozem disketové jednotky a OS CP/M jsou po odstranění prvotních závad, chyb a omylů velmi dobré. IQ 151 dostalo periférii, která z něj udělala nástroj pro serióznější práci ve škole. Šetří čas a nervy.

Tento příspěvek byl napsán pro ty, kteří teprve přijdou do styku s disketou a OS CP/M na IQ 151. Doufám, že napomohu k rychlejšímu pochopení situace a tím vyvarování se chyb a omylů, kterých jsme se dopustili. Závěrem děkuji ing. Wurstovi, Dr. Šafářovi, ing. Bílému, ing. Novákovi a s. Hujňákovvi za trpělivost při spolupráci. *

Program pro kontrolu českého pravopisu

RNDr. Januš Drózd, RNDr. Jan Hajič

Když se před několika léty rozšířily osobní počítače a s nimi i programy na zpracování textu, mělo to značný dopad na způsob, jakým jsou texty vytvářeny. Zatímco dříve se hotový rukopis předával profesionální písaře, dnes mnozí uživatelé počítačů dávají přednost osobnímu vkládání textů. Získávají tím možnost jak obsahového dopracování textu, tak i jeho typografické úpravy přesně podle vlastních představ. Nejsou frustrováni osudovostí okamžiku odevzdání rukopisu a mohou si s textem více "pohrát".

Neprofesionálové u klávesnic však brzy zjistili, že se sotva dokážou vyhnout překlepům, ba co hůř, že ani po mnohonásobném přečtení textu nenajdou všechna prohozená písmenka, chybějící háčky a další produkty nepozornosti. Tak vznikla potenciální poptávka po programech, které dokážou kontrolovat pravopis (anglicky spelling checking). Vzápětí se na trhu objevila řada korekčních programů, ať už zabudovaných do známých textprocesorů (Wordperfect, Wordstar Professional, Chiwriter), nebo samostatných a univerzálně použitelných (Turbo Lightning).

Všechny tyto programy jsou v první řadě orientovány na angličtinu. Některé z nich sice umožňují úpravu pro jiný jazyk, nejsou však s to se vypořádat se slovanským tvaroslovím. Proč? Čeština má sice o něco méně slov než angličtina, zato však mnohonásobně víc tvarů. Pokud je tedy základem korekčního programu slovník tvarů o maximální velikosti 1 000 000 položek, pro angličtinu to bohatě stačí, pro češtinu však nikoli.

Autoři tohoto článku nedávno vytvořili program pro kontrolu českého pravopisu. Rádi by se proto podělili o některé zkušenosti. Neskromně se totiž domnívají, že některé přístupy k tomuto problému mají obecnější platnost a že zkušenosti s vytvářením rozsáhlejšího programového díla nebudou pro čtenáře bez zajímavosti.

1. Analýza požadavků

První otázka, kterou je nutno si položit, zní: "Jaké chyby má program odhalovat?". Češtinář by patrně chyby rozdělil do tří kategorií:

1. Chyby vznikající špatným zapsáním slova, např. nesprávné nebo chybějící písmeno, prohozená písmena a podobné.
2. Chyby ve struktuře věty, např. neshoda podmětu s přísudkem, přívlastku s podmětem, pád neodpovídající předložce atd.
3. Slohové chyby, např. špatné srozumitelné nebo mnohoznačné formulace.

Které z těchto chyb lze zachytit programem? Bod 3. budeme určitě muset oželeť. Sloh je těsně svázan s obsahem textu, a proto posouzení slohu není možné bez porozumění obsahu. Otázky porozumění jsou sice intenzivně studovány, ale ani zdaleka nejsou zvládnuty.

Strukturu vět lze analyzovat na základě české gramatiky. Tato úloha se dnes řeší všude tam, kde se zabývají strojovým zpracováním českého jazyka, např. při strojovém překladu. Na MFF UK Praha byl vyvinut a ve VÚMS Praha implementován program pro překlad technických manuálů z češtiny do ruštiny. Jeho podstatnou nevýhodou je značný objem potřebných dat a pomalost - na osobním počítači trvá překlad jedné stránky okolo 1 hodiny, z toho analýza češtiny zabírá více než 50%.

Z těchto důvodů se program, který má pohotově kontrolovat texty vytvářené na osobním počítači, musí omezit na chyby uvedené pod bodem 1. Na text se bude dívat jako na posloupnost slov a bude ignorovat fakt, že slova vytvářejí složitější struktury - věty, souvětí. Každé slovo bude posuzovat zvlášť, bez ohledu na kontext. Navíc nedokáže odhalit ani takové překlepy, při nichž z jednoho českého slova vznikne jiné (správné) české slovo: např. napíšete-li "káva" místo "kráva", program neohlásí chybu - k jejímu odhalení by musel rozumět obsahu.

Rozhodneme-li se vytvořit takový program, nemůžeme vědět vůbec nic o struktuře českých vět. Musíme ho však naučit rozhodovat, co je české slovo a co není. Jinými slovy, musíme do počítače nějakým efektivním způsobem vložit několik miliónů českých tvarů.

2. Návrh reprezentace českých tvarů

K problému vnitřní reprezentace českých tvarů v programu existují dva principiálně odlišné přístupy. Buď se na množinu všech českých tvarů můžeme dívat jako na neznámou množinu textových řetězců a hledat způsob, jak ji komprimovat, tj. zaznamenat v co nejmenším množství paměti. Nebo můžeme vycházet z toho, že tvary vznikají do značné míry pravidelným skloňováním a časováním (tzv. ohýbáním), a pokusit se naučit počítač jeho pravidlům i výjimkám.

Prozkoumejme nejprve první způsob. Obecné metody komprese sotva dokážou stlačit průměrné množství paměti připadající na jedno slovo pod jeden byte. Proto reprezentace všech tvarů by vyžadovala řádově megabyty paměti, a to je příliš mnoho. Existují však také pravděpodobnostní metody, které mají podstatně menší nároky na paměť, ale neposkytují plnou záruku správnosti. Jedna z nich je popsána v [1]. Pokud bychom se spokojili s tím, že program odhalí 90% chyb, metoda by byla velmi dobře použitelná. Kdybychom však požadovali správnost v 99.9% (což je, myslím, opodstatněný požadavek), množství potřebné paměti by opět nepřiměřeně vzrostlo.

Existují také metody kontroly pravopisu založené na simulaci neuronových sítí, podobných těm, které jsou popsány v [2]. U nich lze pravděpodobnost správné odpovědi zjišťovat pouze empiricky, a dosavadní výsledky nejsou příliš povzbudivé.

Zkusme tedy využít druhý přístup, tj. postavit korekční program na poznacích o ohýbání slov.

Základní myšlenka je velmi jednoduchá. Slovo má kmen a koncovku. Kmen se při ohýbání nemění, koncovka ano. Množina koncovek, které lze k danému kmeni připojit, je určena vzorem. Korekční program proto vybavíme slovníkem, který bude obsahovat kmeny a vzory jejich ohýbání. Ke každému vzoru bude ve slovníku uvedena množina všech možných koncovek.

Tento jednoduchý model však bohužel neodpovídá skutečnosti. Při ohýbání dochází k řadě změn i v kmeni, např. ke krácení kmenových samohlásek, záměně souhlásek atd. Např. slovo "bůh" má 4 různé kmeny: bůh, boh+a, bož+e, boz+i. Tento problém lze řešit asi třemi způsoby:

1. Do algoritmů pro analýzu slov zabudovat schopnost rozpoznat změny v kmeni.
2. Za kmen považovat pouze neměnnou část slova a zbytek prohlásit za koncovku.
3. Dívat se na tvar slova se změněným kmenem jako na jiné slovo a nezávisle jej zařadit do slovníku.

My jsme se rozhodli pro kombinaci způsobů 2. a 3. Pravidelné změny kmene řešíme podle bodu 2., na prosté výjimky rozepíšeme podle 3.

Pokusíme-li se sestavit slovník pro korekční program, zjistíme, že neexistuje dostatečně přesná gramatika češtiny. Ohýbání, které je popsáno v učebnicích (cca 14 vzorů pro podstatná jména, 2 vzory pro přídavná jména, 5 slovesných tříd) oplývá řadou výjimek. I dosavadní systémy počítačové analýzy češtiny (např. analýza pro již uvedený systém strojového překladu) byly vytvořeny pro větší či menší speciální obor, a navíc umožňovaly analyzovat i "nesprávné" tvary - v takových systémech nejde totiž o kontrolu, nýbrž o hlubší analýzu co největšího počtu vět (a bylo by jisté pošetilé se pro takový účel zbavovat možnosti analyzovat větu jen proto, že některý tvar slova je v ní nespisovný). Přesto dokumentace k těmto systémům (viz [3], [4], [5]) byla dobrým vodítkem pro naši práci. Bylo totiž nutno vytvořit systematickou gramatiku češtiny, v níž jsme slova rozdělili do cca 240 vzorů. Výsledkem je, že pro každý vzor existuje jednoznačně určená množina koncovek, které se dají připojit k neměnnému kmeni.

Na jiném místě si však lze gramatiku zjednodušit. Je-li naším cílem pouze kontrola pravopisu bez analýzy struktury vět, můžeme zařazovat slova do slovních druhů podle jejich ohýbání, nikoli podle významu. Tak např. podstatné jméno "stavbyvedoucí" a zájmeno "který" můžeme zařadit pod přídavná jména.

Na tomto místě by pravděpodobně bylo možno skončit. Existují však ještě další pravidelnosti tvarosloví a slovo tvorby, jichž lze využít ke zmenšení paměťových nároků. Rozhodli jsme se, že program bude rozpoznávat:

1. negování slov předponou ne-;
2. tvoření 3. stupně přídavných jmen předponou nej-;
3. vytváření sloves pomocí předpon za-, vy-, při-, roz-, pře-, s-, z-, do- atd.

Dosud popsaný model totiž vůbec neuvažoval předpony. Připojením předpony by vzniklo nové slovo, které by vyžadovalo samostatné heslo ve slovníku. Tak např. kromě "dělat" by ve slovníku muselo být také "dodělat", "předělat", "rozdělat", "udělat",

"vydělat", "zadělat", "nedělat", "nedodělat" atd. To by nebylo zrovna efektivní.

Není zcela jasné, jak se optimálně vypořádat s předponou ne-. Česká gramatika nedává žádná omezení na to, kdy lze negaci použít. Proto jsme povolili použití předpony ne- před každým slovesem a před většinou přídavných jmen i přesto, že jsme tím zahrnuli do jazyka některé málo frekventované a exoticky znějící tvary (např. nezvířecí).

Předponu nej- stačí prostě povolit v 3. stupni přídavných jmen. 3. stupeň lze ve většině případů rozpoznat podle koncovky.

Nejvíce problémů přináší ostatní předpony. U každého slovesa uvádíme, které předpony mu lze předřadit. To však nestačí. Některé tvary lze totiž tvořit pouze od dokonavých nebo pouze od nedokonavých sloves. A předpona může (ale nemusí) změnit vid slovesa. Např. přechodník přítomný lze tvořit od nedokonavého "dělat" ("dělaje"), ale nikoli od dokonavého "udělat" ("udělaje" je špatné). Opačné je tomu u přechodníku minulého - "udělav" je správné, "dělav" chybné. U dokonavého "zout" předpona vid nemění - "zujíc" i "vyzujíc" je špatné, "zuv" i "vyzuv" správné. A tak dále...

3. Analýza požadavků II

Po zvládnutí záludnosti českého jazyka je třeba se vrátit ke specifikaci požadavků na korekční program a uvážit, jakým způsobem se bude používat.

Pro uživatele je nejpříjemnější, když je kontrola pravopisu zabudována do textprocesoru. Nebyl by problém k tomuto účelu vytvořit nový textprocesor. Žádný uživatel však nepřechází rád na nové programové vybavení a nevzdává se rád svých zvyklostí. Proto je potřebné, aby korekční program mohl spolupracovat se stávajícími textprocesory.

Jedno řešení problému integrace s textprocesorem vymyslela (pravděpodobně) firma Borland. Korekční systém je paměťové rezidentní program a vyvolává se stisknutím vhodné kombinace kláves za běhu libovolného textprocesoru. Nevýhodou je, že rezidentní program nezná vnitřní uložení textu v paměti a proto může kontrolovat jen to, co vidí na obrazovce, nebo to, co vstupuje z klávesnice. Jeho použití je navíc omezeno na ty textprocesory, které pracují v textovém režimu zobrazování.

Druhou možností je kontrolovat již vytvořené texty uložené v souborech. Protože však v průběhu kontroly nepracuje textprocesor, nelze provádět přeformátování opravovaného textu ani náročnější editační zásahy (aktualizace odkazů v dokumentu, změna rozlámání do stránek atd.).

Protože ani jeden z těchto způsobů není jednoznačně lepší, pokusme se zformulovat požadavky na implementaci obou.

Od rezidentního programu bychom rádi očekávali toto:

- schopnost zkontrolovat slovo, na němž je kurzor;
- schopnost zkontrolovat všechna slova na obrazovce;
- schopnost průběžně kontrolovat text psaný na klávesnici nebo opravovaný na obrazovce.

Při kontrole obsahu obrazovky si lze představit dva režimy práce. Buď se chybná slova pouze nějak označí na celé obrazovce najednou, nebo se budou procházet po jednom a u každého slova uživatel dostane možnost jej na místě opravit. Program by měl umět oboje.

Kontrola obsahu obrazovky vyžaduje, aby program věděl, v jaké části obrazovky je editovaný doku-

ment. Jinak by kontroloval i ostatní slova, která jsou zrovna za obrazovce, třeba jméno souboru, název textprocesoru atd. Informaci o umístění dokumentu lze programu zprostředkovat tak, že mu sdělíme, s kterým textprocesorem spolupracuje. Avšak v těch systémech, které dovolují otevřít editační okno kdekoliv na obrazovce (Framework II), nebude omezení kontroly na správný výřez zaručeno.

Průběžná kontrola psaného textu je složitější. Představme si, že přesuneme kurzor na nějaké slovo, opravíme v něm jedno písmeno a s kurzorem zase odjedeme. Aby program mohl posoudit opravené slovo, nestačí mu sledovat klávesnici, musí si je přečíst z obrazovky. Musí také poznat, které klávesy způsobují přemístění kurzoru a smazání znaku. Musí tedy znát některé detaily ovládání textprocesoru, s nímž spolupracuje. Průběžná kontrola musí být vypínatelná. Je to potřebné nejen při psaní jinojazyčné části dokumentu, ale také tehdy, když korekční program nestačí rychlosti pisárky.

Kontrola textů uložených v souborech je úloha podstatně jiná. Musíme si uvědomit, že každý textprocesor ukládá text ve svém vlastním formátu, který kromě dokumentu obsahuje řadu řídicích znaků popisujících rozlámání, fonty, typografické zvláštnosti atd. Korekční program musí tento formát znát alespoň do té míry, aby dokázal rozlišit slova, řídicí znaky a rozdělovací znaménka.

Je-li v kontrolovaném souboru nalezena chyba, je třeba poskytnout operátorovi možnost ji opravit. Zároveň však nelze poskytnout všechny editační možnosti textprocesoru. Zde je nutný kompromis, který jsme vyřešili takto: na obrazovce otevřeme 5-řádkové okénko s textem, v němž lze provádět omezenou editaci vkládáním a rušením znaků a s pomocí kláves pro přesun kurzoru (šipky, Home, End). V okénku jsou zobrazeny i řídicí znaky, ty však nelze vkládat ani rušit, aby se neporušila vnitřní struktura textu.

V návrhu korekčního systému je řada rysů, které jsou nezávislé na tom, zda jde o rezidentní program nebo o program kontrolující soubory. Je třeba se např. rozhodnout, co dělat s chybnými slovy. Když program prohlásí slovo za chybné, znamená to, že jej nezná. Může se snadno stát, že jde o slovo specifické pro nějaký vědní obor, které prostě nebylo zahrnuto do slovníku. Je proto žádoucí, aby si uživatel mohl vytvářet vlastní slovník odborných termínů. Kromě toho existují slova, která se vyskytují v nějakém konkrétním dokumentu a v něm je chceme považovat za správná, ale jinde jsou chybná. Uživatel si proto může vytvořit slovníček, který platí pouze po dobu práce s programem.

Pamatovat je nutno i na to, že se slova na konci řádky dělí a korekční program by měl být schopen spojit obě části rozděleného slova.

4. Návrh reprezentace dat a algoritmů

Z předchozích úvah vyplynulo, že korekční systém se sestává z programu a slovníku. Rozsah slovníku však může být takový, že se celý nevejde najednou do paměti. To platí zejména o rezidentní verzi, protože v paměti musí být současně přítomen i textprocesor. Proto je výhodné slovník rozdělit do dvou částí: menší část s nejfrekventovanějšími slovy, která bude trvale sídlit v paměti, a větší část, která bude na vnějším paměťovém médiu. Pokud se posuzované slovo nenajde v menším slovníku, přečte se do paměti úsek většího slovníku a slovo se hledá v něm.

Při návrhu struktury slovníků vycházíme ze dvou částečně protichůdných požadavků. Slovník musí být

co nejmenší a musí se v něm co nejrychleji vyhledávat. Pro vyhledávání jsme použili metodu, které se říká indexsekvencní přístup. Slova jsou uspořádána abecedně a navíc je slovník rozdělen do úseků. Ke každému úseku existuje informace, jaké je jeho první slovo, takže lze snadno zjistit, ve kterém úseku se má slovo hledat. Úseky jsou vytvořeny tak, aby o začlenění do úseku se dalo rozhodovat na základě prvních tří písmen slova. Úsek má obvykle velikost mezi 70 a 200 slovy a prohledává se sekvenčně.

Sekvenční alfabetské uspořádání není sice nejvýhodnější z hlediska rychlosti vyhledávání, ale dovoluje značné zhuštění dat. Využili jsme techniky, které se v oblasti databází říká zkracování klíčů. Vychází z toho, že při abecedním uspořádání se sousední slova liší převážně až na konci a začátek mívají stejný. Proto není nutno tento začátek stále opisovat, stačí u každého hesla uvést, kolik písmen má stejných jako předchozí heslo a pak odlišná písmena.

Ve slovníku se za takto zkráceným slovem nachází jednobytový údaj o vzoru a pro slovesa také dvoubytová informace, které předpony lze ke slovu připojit. Celkově jsme dosáhli toho, že pokud se nějaké slovo liší od předchozího nejvýše dvěma písmeny, zabere ve slovníku zpravidla 3 byty (není-li to sloveso) nebo 5 bytů (je-li to sloveso). Takové 5-bytové heslo může v případě sloves reprezentovat přes tři tisíce tvarů.

Uvažujme např. slovo >nejnevykrystalizovanějšímu<. Jeho zpracování probíhá takto:

a/Odpojí se předpony nej-, ne- a vy-.

b/ve slovníku se nalezne "kmen" >krystaliz<.

c/Podle vzoru se zjistí, že k tomu "kmeni" lze připojit "koncovku" >ovanějšímu< a že smí mít předponu ne-.

d/Ověří se, že koncovka >ovanějšímu< zaručuje 2. stupeň přídavného jména a tím je oprávněno odpojení předpony nej-.

e/Ověří se, že "kmen" >krystaliz< připouští předponu vy-.

5. Technické řešení

Po dokončení specifikace požadavků a po návrhu datových struktur a algoritmů je před započítím programování dobré uvážit, jak bude návrh ovlivněn specifickými podmínkami na počítači, pro nějž je určen.

Za cílový stroj jsme si vybrali počítače kompatibilní s IBM PC, které jsou v Československu nejrozšířenější. Jejich operační systém MS-DOS do značné míry ovlivnil technický návrh. Výhodou MS-DOSu je, že umožňuje vytváření rezidentních programů. Nevýhodou je, že pro ně nestanovuje čestná ani jasná pravidla hry. Když rezidentní program otevře soubor, systém mu jej může bez jeho vědomí a svolení uzavřít - např. tehdy, když skončí textprocesor, s nímž rezidentní program spolupracoval. Dojde-li při práci korekčního programu ke kritické systémové chybě, musíme zajistit, aby ji systém neoznámil textprocesoru - co by si s ní ten chudák počal? Podobných problémů je celá řada.

Rezidentní verze programu musí mít přístup k obrazovce textprocesoru. Přístup přes BIOS je příliš pomalý, proto je nutno pracovat s obrazovou pamětí přímo. Program pak musí hlídat jak režim zobrazení, tak i číslo právě aktivní stránky.

Pro aktivaci rezidentního programu je třeba používat takové klávesy, které nejsou obsazeny

žádnou funkci textprocesoru. Wordperfect však umožňuje využít všechny standardní kombinace kláves. Pak je nutno se buď uchýlit k nestandardním kombinacím (a řešit problémy s odlišností klávesnic připojitelných k IBM PC), nebo umožnit uživateli definovat, které kombinace chce používat. Rozhodli jsme se pro druhé řešení, protože textprocesory zpravidla dávají uživateli k dispozici několik kombinací, které lze, ale není nutno, použít pro vyvolávání maker.

6. Kódování

Korekční programy jsme zapsali převážně v jazyce C, který nám poskytl potřebnou flexibilitu při práci s nestandardními datovými strukturami a při spolupráci se systémem. Časové kritické části programu (např. hledání ve slovníku) jsou naprogramovány v assembleru. Při ladění jsme postrádali účinné běhové kontroly a proto jsme doprovodné programy (instalační a konfigurační program) vytvořili v Pascalu.

Každá verze programu má cca 3500 řádek velmi hutného kódu v C a assembleru. Vytvořili jsme ještě třetí verzi, která prochází text uložený v souboru a pouze označuje chybná slova. Uživatel si pak může otevřít textprocesor, najít značky a opravit chyby. To je výhodné u těch textprocesorů, které po opravě vyžadují přeformátování textu (např. Wordstar, Chiwriter).

7. Výsledek a jeho distribuce

Korekčnímu programu jsme dali nenápadité jméno KOREKTOR. Jeho slovníky obsahují cca 60000 položek, což odpovídá více než 100000 slovníkových hesel. Celkový počet tvarů, které KOREKTOR rozpozná-

vá, přesahuje 6 miliónů. Uživatel si může vytvářet vlastní doplňkové slovníky. Zjistili jsme, že tento rys je potřebný, protože zejména v odborných textech se setkáváme se slovy, která ve slovnících ještě nejsou.

Výsledný program se dá provozovat na počítačích plně kompatibilních s IBM PC vybavených alespoň 512 KB operační pamětí, obrazovým adaptérem CGA, EGA nebo VGA a buď pevným diskem nebo dvěma jednotkami pružných disků.

V současných podmínkách si autor programu může vybrat, zda distribuci programu bude zajišťovat sám, nebo ji svěří zavedené firmě. My jsme se (hlavně z pohodlnosti) rozhodli pro druhou možnost a uzavřeli jsme smlouvu s JZD AK Slušovice. Testování programu bylo ukončeno v září 1989 a od té doby je v běžném prodeji.

Literatura

- [1]Fiala, P.: Počítač v roli češtináře. In: Sborník SOFSEM 1986, 2. svazek, pp. 187-190.
- [2]Sejnowski, T.J., Rosenberg, C.R.: Parallel networks that learn to pronounce English text. In: Complex Systems 1(1), pp. 145-168.
- [3]Králiková, K., Sgall, P., Weisheitelová, J.: Morfemic Analysis of Czech. In: Explizite Beschreibung der Sprache und automatische Textbearbeitung VII, MFF UK Praha, 1982.
- [4]Panevová, J. et al.: Lexical Input Data for Experiments with Czech. In: Explizite Beschreibung der Sprache und automatische Textbearbeitung VI. MFF UK Praha, 1981, 160 pp.
- [5]Kirschner, Z.: MOSAIC - A Method of Automatic Extraction of Significant Terms from Texts. In: Explizite Beschreibung der Sprache und automatische Textbearbeitung X. MFF UK Praha, 1983, 124 pp.

POMŮCKA PRO NÁVRH PLOŠNÝCH SPOJŮ

Jaroslav Kusala

Při návrhu rozsáhlejšího obrazce plošného spoje obvykle používáme vhodný rastr, ve kterém hledáme nejvhodnější rozmístění součástek s ohledem na jejich propojení podle schématu. V obvodech s diskrétními prvky vystačíme s rastrem na milimetrovém nebo čtverečkovaném papíře. Vývody integrovaných obvodů však mají rozteč 2,54 mm a rastr na milimetrovém papíru už není nejvhodnější. Program, jehož výpis je uveden v tabulce č. 1, zhotoví pomocí počítače IQ 151 a zapisovače Minigraf 0507 síť bodů s roztečí 2,54 mm. V tomto rastru můžeme přesně navrhnout umístění vývodů integrovaných obvodů a dalších součástek včetně jejich propojení. Hotový návrh je v měřítku 1:1, takže jej lze po přenesení na pauzovací papír přímo použít ke zhotovení plošného spoje fotografickou cestou.

Po spuštění programu zadáme počet bodů ve vodorovném (X) a svislém (Y) směru a na připojeném Minigrafu se zobrazí síť X.Y bodů. Ve vodorovném směru je vzhledem k rozměrům pracovního pole Minigrafu vhodné volit maximálně 60 bodů, ve svislém směru maximálně 90 bodů. Vzhledem k tomu, že Minigraf kreslí s krokem 0,125 mm, vznikne po nakreslení tří bodů (60 kroků) chyba 0,120 mm. K jejímu eliminování udělá Minigraf vždy po nakreslení tří bodů jeden krok "naprázdno". Použitím této korekce vznikne při kreslení 60 bodů pouze zanedbatelná odchylka 0,1 mm.

Na rastr s hotovým návrhem (ze strany plošných spojů) přiložíme pauzovací papír a na příslušná místa obtiskneme obrazy pájecích bodů pro součástky a objímky integrovaných obvodů z aršíku suchých obtisků PROPISOT. Tyto pájecí body propojíme podle návrhu tuší a získáme klišé pro zhotovení plošného spoje fotografickou cestou. Pokud bychom nechali místo bodů nakreslit kroužky o průměru asi 1,8 mm, získali bychom po ofotografování klišé pro zhotovení univerzální desky plošného spoje pro zapojení číslicových zařízení.

Tabulka 1

```
10 REM - rastr bodu 2.54 mm -
20 ORG 0,500: MOVA 0,500
30 CLS: PRINT "POCET BODU RASTRU"
40 INPUT "Vodorovne";X
50 INPUT "Svisle";Y
60 X=X/3: Y=Y/3
70 FOR L = 1 TO Y
80 FOR K = 1 TO 3: FOR I = 1 TO X
90 FOR J = 1 TO 3: POINTR 20,0
100 NEXT J: MOVR 1,0: NEXT I
110 FOR I = 1 TO X
120 FOR J = 1 TO 3: MOVR -20,0
130 NEXT J: MOVR -1,0
140 NEXT I: MOVR 0,20
150 NEXT K: MOVR 0,1: NEXT L
```

MONITOR *informací*



Modula-2

Tento programovací jazyk, vyvinutý profesorem N. Wirthem jako následovník Pascalu, obsahuje prostředky pro implementaci paralelních procesů a modulární tvorbu programů. Analýza možností využití tohoto mocného moderního programovacího jazyka je obsahem článku v časopisu PCW 7/89-str. 222-227. Pro zájemce uvádíme, že publikaci "Programování v jazyce Modula-2" vydal ÚV Svazarmu - avšak pouze pro potřebu vlastního aktivu. Není to škoda?

LOG.EXE

je název jednoduchého programu pro evidenci přístupu jednotlivých uživatelů k programům. Zdrojový text v jazyce TURBO C V1.5 je součástí zajímavého článku v časopisu PCW 7/89-str. 233.

PCW WORLD FOCUS

je příloha časopisu Personal Computer World, která se zabývá určitou tematikou ze světa osobních počítačů. Číslo 7/89 se zabývá programovým vybavením pro "obchodní" sféru, jak ukazuje obsah:

COMPUTERISING YOUR OFFICE	/	SPREADSHEETS
PROJECT MANAGEMENT	/	TRAINING
ACCOUNTS	/	TIME MANAGEMENT
CAREERS	/	GLOSSARY

dGE - grafika pro dBase

je nadstavbou databázových programových nástrojů typu dBase III - IV, Clipper, Foxbase, Eagle, Quicksilver a lze jej využít v kompilátorech jazyka C. Umožňuje tvorbu přesnější grafiky pomocí 80 příkazů s možností ovládnutí periférií pro tisk výstupu - maticovými tiskárnami počínaje a souřadnicovými zapisovači konče. Systém obsahuje prostředek pro tvorbu znakových sad a speciální prostředek pro interaktivní tvorbu grafického výstupu na obrazovce, kdy po ukončení návrhu je vygenerován zdrojový text v dialektu dBase. PCW 7/89-str. 173.

Za hranice VGA

vás zavede článek v časopisu PCW 7/89-str. 184-188, který specifikuje nároky na výkonné grafické adaptéry a popisuje cesty, které vedou k úspěšnému řešení těchto nároků.

CASE - program tvoří programy

CASE je zkratkou pro Computer Aided Software Engineering, neboli tvorba programového vybavení pomocí počítače. MATRIX LAYOUT od firmy Matrix je předmětem analýzy Nicka Dockinga v časopisu PCW 7/89-str. 150-154. Výsledkem úvah autora článku je zásada používat tento programový nástroj pro rychlou tvorbu projektů s efektivním uživatelským grafickým prostředím - není však určen pro začátečníky. Článek obsahuje zajímavé zkušenosti a postřehy.

ADA

Studie amerického ministerstva národní obrany ukázaly, že programové vybavení pro potřebu DD (Department of Defense) bylo v roce 1979 psáno ve 450 (!) druzích programovacích jazyků. Analýza této problematiky, shrnutá ve zprávě STRAWMAN, vyústila v několik dalších zpráv a závěrečná zpráva nazvaná STEELMAN znamenala konečný impuls pro zrod nového programovacího jazyka třetí generace poeticky nazvaného ADA. Popis jazyka ADA včetně příkladů zdrojových textů najdete v časopisu PCW 7/89-str. 156-160.

Elektronická pošta - hit roku 1990?

Podle tvrzení zahraničních odborníků bude nejpopulárnějším programovým vybavením v roce 1990 elektronická pošta, jinak také E-mail (Electronic mail). Známý Steve Jobs připravil pro svůj superpočítač Next také superprogramové vybavení Multimedia Mail. Toto programové vybavení umožní revoluční formy vzájemné komunikace mezi uživateli osobních počítačů a to až na úroveň celosvětových komunikačních sítí. Fascinující podrobnosti se dočtete v časopisu PCW 7/89-str. 118.

1.44 "mega" ze 720 "kilo"?

Zajímavou úpravu, která umožňuje 3.5" diskety s kapacitou 720 kilobajtů formátovat na 1.44 megabajtů popisuje Nick Pettefar v časopisu PCW 7/89-str. 126 (DIY Disks - zkratka DIY znamená Do It Yourself).

Přenosný IBM P70 PS/2

s plasmovým displejem VGA, 2 konektory MCA, procesorem 80386/20MHz a 8 MB RAM je předmětem testu v časopisu PCW 7/89-str. 131-134.

ACORN A 3000

je nejrychlejším osobním počítačem současné doby. Je vybaven procesorem ACORN ARM (32bitů/8MHz) a je vybaven pamětí s kapacitou 1-2 MB (RAM) a 512 KB (ROM). Maximální obrazové rozlišení je 1056x256 bodů v paletě 4096 barev. Podrobný popis je v PCW 7/89-str. 136-140.

Běžné neprogramující uživatele dělila od používání dosavadních objektově orientovaných programů nepřekonatelná hráz. Tyto programy jsou rozděleny do jednotlivých objektů či modulů obsahujících různé části programu a dat. Vzniklá modularita nabízí možnost určitých změn v jednotlivých programových objektech bez nepříznivých vedlejších účinků v jiných modulech. Pochopitelné, že ovládnutí programu vyžaduje nejen znalost syntaxe příkazů, ale i určitou míru chápání, o co vlastně ve věci jde. Firma Maxem (USA) přichází s databázovým aplikačním generátorem Cause pro vizuální programování volbou z umně sestavených menu. Tím zcela obchází problém syntaxe psaných příkazů. Když zakončíte rozvržení své budoucí aplikace, systém uloží vaši volbu jako program ve speciálním objektově orientovaném metajazyku. Firma sama přichází s nabídkou podobných hotových programů nazvaných Effects. Ty nelze měnit přímým kódováním, ale opět prostřednictvím generátoru Cause, který tak vlastně funguje jako vizuálně programovaný interface. Systém má interpreter i kompilátor, vše je převoditelné mezi Macintoshem a IBM PC. Cena \$495. Za dalších \$100 dostanete licenci na jeden rok, během něhož můžete prodat libovolné množství svých vlastních aplikací.

"Vždycky začínají úplně v levém rohu dole a pak si to pod úhlem 45 stupňů šinou vzhůru", prohlásil Bill Krause z firmy 3COM. Touto žertovnou průpovědkou měl na mysli grafické vyjádření nárůstu prodeje produktů počítačových výrobců. BYTE 12/88 se spolu s computerovými futurology zamýšlí nad tím, co nás očekává v prvních letech bližící se dekády. Kompatibilita všeho se vším to ještě nebude (to prý až tak na začátku dalšího tisíciletí). Jak tedy bude za pár let vypadat typické pecéčko? Bude mít floppy 3,5" pro 2-4 MB, hard disk 3,5" pro 80-140 MB, optický disk 0,5-1 GB, 32-bitovou CPU, 4-16 MB paměti RAM, monitor s miliónem grafických bodů, spoustu interfaců a komunikačních možností. Většina softwaru bude v křemíku ve formě programovatelných procesorů. Bude převažovat architektura Micro Channel a IESA. Pokud se podaří vytvořit univerzální zvukové vstupy pro rozeznávání řeči, předpokládá se, že jim uživatelé budou dávat přednost před klávesnicí či myší. Počet integrovaných obvodů v počítači se sníží asi na 40 (PC AT v roce 1982 jich mělo 200, dnešní PS/2 model 50 kolem 80). Možné je, že se u laptopů prosadí hard disky 2,5" pro 20 MB. Značně se rozšíří lokální sítě - bude v nich 50 až 90 procent všech počítačů (oproti dnešním 13%). Data se v nich budou přenášet rychlostí 100 Mb/s (dnes 10 Mb/s). Většina systémů bude mít procesor Intel 80386, zatímco 8088/86 bude v menšině (opak dnešní situace). Systémy s Motorolou 680x0 budou nadále populární, ale ne dominantní. Pokud jde o věčné soupeření operačních systémů, kterého už mají uživatelé plné zuby, John Roach z firmy Tandy se k němu na jedné konferenci vyjádřil slovy: "O tom rozhodnou lidi, ne vy, co tu sedíte." A ještě něco - pohyb cen dolů se bohužel nepředpokládá.

"Vaše firma si kvůli vašemu budoucímu softwarovému výtvoru nechala uvalit hypotéku na svou farmu. Dlouho očekávaná demonstrace hotového programu však představenstvo doslova nadchla. Distribuce klapla na čas. V tom telefon: "Ten váš software dělá takové nějaké podivné věci..." Váš kredit, vaše peníze, váš rozum, všechno je v sázce. Namáháte mozek, abyste na něco přišel. V čem to,

kde to tak asi...? A ke všemu je váš program hrozně dlouhý. Spolu s jeho tabulkou symbolů a debuggerem se do paměti nevejde. Náhle vás napadá, že myšlenka otevřít si krám s jízdami koly není vlastně tak špatná..." Těmito slovy uvádí firma Atron svůj produkt 386PROBE Bugbuster (Krotitel štěnic). A pokračuje: "Děvět z deseti předních producentů softwaru spí klidně díky nepřetržité asistenci našich hrdwarových debuggerů v reálném čase. Můžete si nastavit podmíněné body přerušování, které zastaví program jen v případě špatně provedené procedury. Můžete dokonce detekovat i obskurtní, sekvenčně závislé problémy zastavením v bodu přerušování, když se neobjeví přesně stanovený průběh operací v přesně stanoveném pořadí. Pak se můžete podívat, k čemu vlastně došlo. 386PROBE má buffer pro uchování posledních 2 Kcyklů programových exekucí. Ačkoli jiné firmy dělají něco podobného, Atron je jediná, která dokáže řadit průběh operací a dat souběžných/přerušovaných průběhů. 386PROBE má 1 MB stínované paměti pro uložení debuggeru a tabulky symbolů vašeho programu. Tato paměť je chráněna před zápisem. Tak do debuggeru nemůže vaše štěnice proniknout." Povídání je doplněno roztomile vyvedenou štěnicí, vykukující z obrazovky. Jde o ten nejsubtilnější computerový druh, tzv. "mimo dostřel". Objevuje se jen na krátký moment přerušování, po němž je přepsána. Bohužel, cena osazené desky se softwarem není uvedena.

V americkém filmu Last Trek (Poslední cesta) se posádka lodi z daleké budoucnosti zastavuje v naší současnosti. Jeden z jejích členů usedá k počítači Macintosh, bere ze stolu myš, přikládá si ji k ústům a říká do ní své povely... Kino se směje. Firma Advanced Products and Technologies (USA) však už do oné budoucnosti nakročila. Její počítač The Voice (Hlas) dokáže přijímat hlasové povely v přirozeném jazyce - a to nezávisle na jeho dialektu! Představitel firmy tvrdí: "The Voice se stane nejprátelštějším počítačem na světě." Svými rozměry je jen o málo širší než dvě videokazety položené na sobě. Samozřejmě je přenosný. Se 16-řádkovým LCD displejem váží cca 1,8 kg. Ještě neuvěřitelnější vypadá předpokládaná cena této novinky - jen \$2000. V porovnání s IBM PC má stotisíckrát rychlejší přístup k souborům, šestinásobnou kapacitu paměti (4 MB) a větší počet procesorů (8- i 16-bitových). Nová koncepce je samozřejmě dána i novým operačním systémem, speciálním softwarem a zákaznickými obvody. Firma k tomuto kroku přistoupila proto, že: "současné operační systémy jsou příliš pomalé, software pro rozeznání řeči příliš archaický." Z hlediska uplatnění své hlavní funkce je Voice značně flexibilní. Jeho práce je závislá na vložených (vyměnitelných) aplikačních kazetkách velikosti mikrofloppey. Některé z kazetek zajišťují překlad mezi různými jazyky. Firma naznačila, jak na to jde: "Rozpoznávací systémy se dosud snažily pracovat s celým jazykem. My jsme jej rozdělili do jednotlivých aplikačních oblastí. Náš systém vnímá jen to, co je potřebné pro danou aplikaci." Chytré, že? I když s poezií to asi bude trochu slabší. Jeden z programů umožňuje telefonické zavolání (ústně) jmenované osoby, vyvolání její adresy apod. za dobu kratší než jedna vteřina. Advanced Products zatím nechce prozradit, které obří firmy upravují svůj software pro Voice. Leccos se ale dá vytušit z odpovědi na otázku, zda bude Voice kompatibilní s nejužívanějším softwarem: "Tady nebude žádný problém."

PROGRAMOVÁ NABÍDKA



(pokračování z čísla 9)

8. vybavení on-line help, tj. nápovědou na stavové řádce ke každému příkazu (položce v menu nabídky)
9. definice makro instrukcí formou provádění příkazů editoru s možností uložení na disk
10. inteligentní práci s adresáři a soubory (výběr, zobrazení) s možností provádění příkazů MS-DOSu přímo z editoru nebo odchod do operačního systému se zpětným návratem (tzv. OS SHELL)
11. elektronický kalkulátor s vložením výsledku do textu, zobrazení aktuálního času, vkládání data a času do textu, prohození malých písmen za velká ap.

Součástí editoru Text602 je obsažná uživatelská příručka.

2. Disketa #1 (pro programátory)

1. FC602 - program pro formátování zdrojových textů v jazyce "C". Ideální pomůcka pro tvorbu jednotné dokumentace všude tam, kde je zaveden jako hlavní programovací jazyk "C". Formátováním do normy K&R programy získávají na přehlednosti a dostávají sjednocující charakter.
2. MCRT a MHELP - knihovny pro práci s textovými okny (tvorba menu a dalších prvků kvalitního uživatelského prostředí (MCRT) a pro zavedení interaktivního helpu (MHELP) v programech psaných v Turbo Pascalu (5.0)
3. MOUSE602 - s pomocí této unit pro Turbo Pascal (5.0) můžete ve svých programech jednoduše ovládat myš (Microsoft kompatibilní) buď v textovém nebo v grafickém režimu. K dispozici je i knihovna pro práci s kalendářními údaji.

3. Disketa #2 (pro zpracování textu)

1. MP602 - univerzální makroprocesor, který je určený pro dávkové editování textu, generování programů, vytváření tiskových sestav, filtrování textových souborů ap. MP602 je jazyk, který umožňuje neobvyklou šíři použití.
2. KODING - program pro překódování textových souborů (a nejen textových) prakticky z čehokoliv na cokoliv. Má vlastní editor kódovacích tabulek (zámena sekvence až 32 x 32 znaků). Tři typy kódování: standardní, filtrované a změnové.

VFORM a SEKTOR - programy pro přenos souborů z disket formátovaných pod operačním systémem CP/M. Umožňují přenos ze všech známých CP/M formátů do IBM PC (pouze 40 stop).

Poznámka: Tyto programy jsou upravené public domain software do češtiny a nejsou kalkulovány do ceny diskety.

C. Zakázkové programové vybavení

Pro organizace přicházíme s nabídkou nové služby. Jedná se o zakázkové programové vybavení (pro libovolný okruh problémů) řešené pomocí databázových systémů typu dBASE (FoxBASE, Clipper). Zakázka je komplexní službou a obsahuje:

1. analýzu problému
2. realizaci
3. předání dokumentace
4. zaškolení obsluhy

Máte-li o vytvoření individuální databázové aplikace zájem, zašlete objednávku na adresu:

602. ZO Svazarmu, Dr. Z. Wintra 8, 160 41 Praha 6

V objednávce uveďte bližší specifikaci požadavku, na jehož základě provede odpovědný pracovník naší organizace předběžný rozbor. Po vzájemné dohodě bude uzavřena řádná hospodářská smlouva, která obsahuje body 1 až 4.

Poznámka: Programová nabídka pro osmibitové počítače, uveřejněná v předchozích číslech, zůstává v platnosti.

OBJEDNÁVKOVÝ LIST

Název organizace :

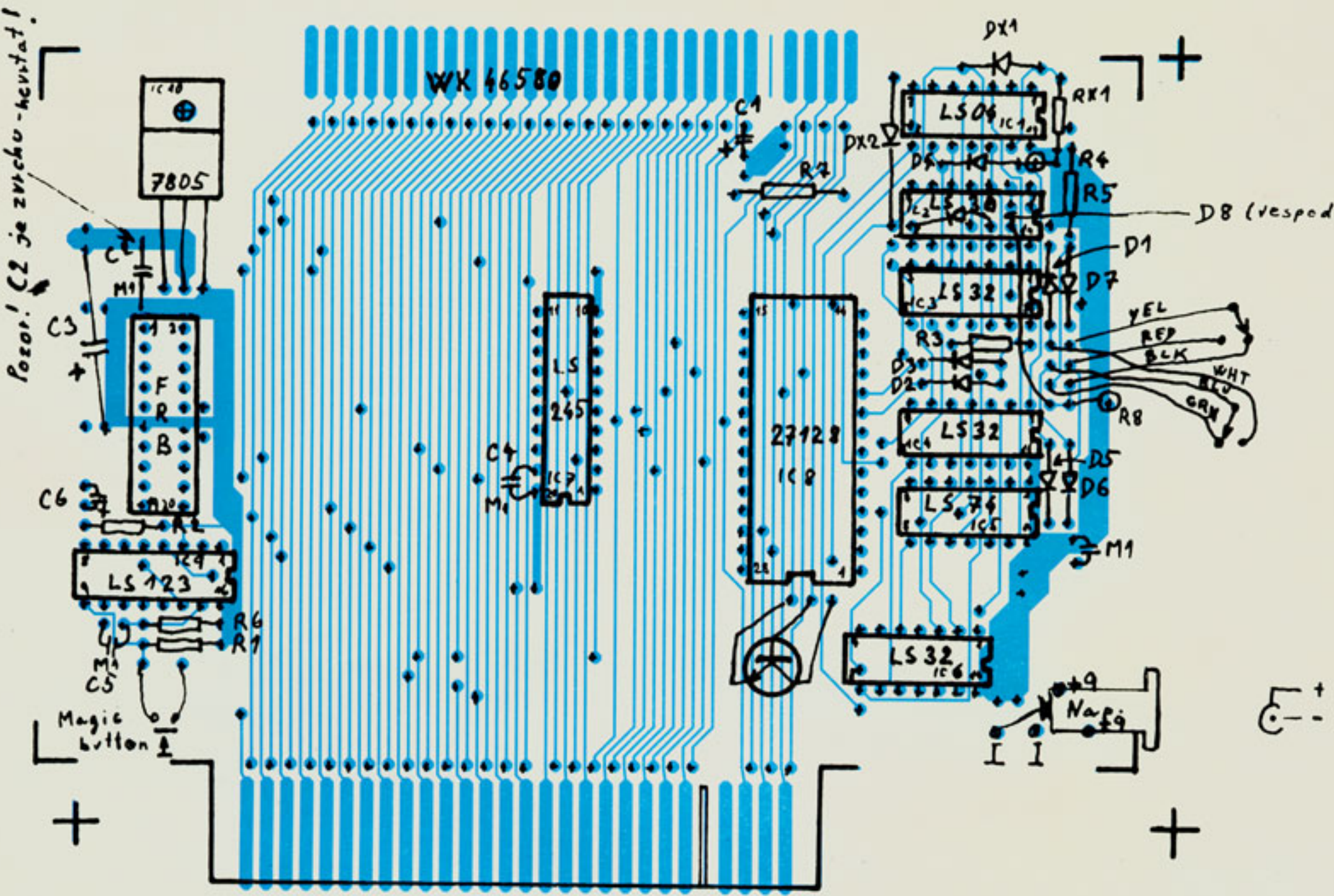
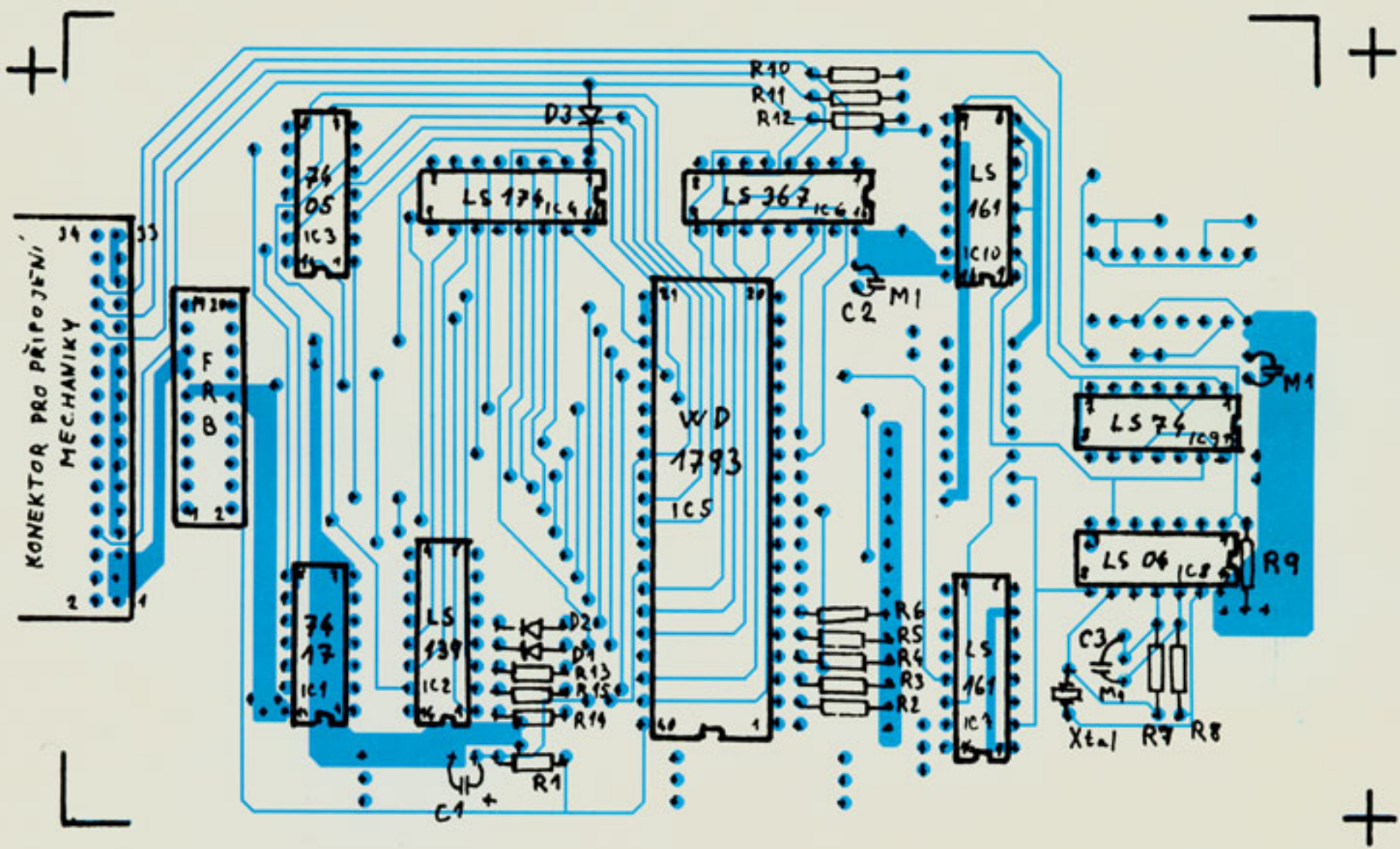
Přesná adresa :

Vyřizuje pracovník :

Programové vybavení	Cena Kčs	Počet kusů
Kurs MS-DOS	998,-	
Kurs dBASE III Plus	998,-	
Textový editor Text602	2998,-	
Drivery Text602 (maticové tiskárny)	689,-	
Drivery Text602 (laserové tiskárny)	1189,-	
Disketa #1 (pro programátory)	798,-	
Disketa #2 (pro zpracování textu)	798,-	

POSTAVTE SI S NÁMI DISKOVÝ ŘADIČ

Rozložení součástek.
 Nahore deska řadiče.
 Dole deska systému v5.03.





Z 88 V R O C E 1 9 8 9

Sir Clive Sinclair má pověst geniálního konstruktéra a nevalného obchodníka. Snad právě spojení těchto vlastností vzbuzuje sympatie jeho četných příznivců. Je autorem celé řady vynálezů. Za jeho činnost mu byl udělen šlechtický titul. Vešel do historie nejen jako tvůrce ZX80 a ZX Spectra, ale je považován i za vynálezce elektronické kapesní kal-

kulačky. Produkty "Sinclair made" se staly předmětem zájmu sběratelů. Jméno Sinclair nemůže již ale pro své produkty používat, po řadě obchodních neúspěchů byl nucen je postoupit Alanovi Sugarovi (AMSTRAD). Pro vytvoření svého nového dítěte založil Sir Clive novou firmu, Cambridge Computer LTD.

(Z článku uvnitř čísla.)

