

číslo 3. - březen 1992 - ročník 3.



AMATÉRSKÝ
PROGRAMÁTOR



Informační
a propagační bulletin
soukromé fi SECON pro
začínající uživatele
počítačů ZX5-Didaktik.
Udává Petr Černý.
Tisk vlastní, vychází
vždy 15.v měsíci.
Redakce: SECON
Staré Křečany
407 61

OBSAH:

- Z redakce
- Nebojte se strojíku
- Kompilátor
- Poznávám svůj počítač
- Basic pro začátečníky
- Noše nabídka
- Příspěvky čtenářů aj.

PROGRAM MAKER V PUBLIC DOMAIN
pro floppy disk i kazety

co nabízí SINCLAIR CLUB OSTROV



v pravidelných
rubrikách najdete:

- PROGRAMOVÝ ROZSAH BASICU
- NOVÁ VERZE ? PROGRAMU → »DENÍK« ← ?
- STRUKTURA PROCEDURU - »ŘADIČ«
- FORMÁT DISKETY N° 433152 KB
- »RUN FOR D-40« ↔ DRUHÁ VERZE ←
- BŘEM JEŠTĚ (A ZASE) I JINAK
- KOMPILÁTOR »MCDER«
- »RENAME DISK«
- FUNKCE RAND
- PROGRAM »POKE«
- KLUB D-40





Ještě než se budu venovat odpovědím na vaše dotazy, rád bych v úvodu poděkoval redakci FIFA za velice pěkný úvod k rozhovoru s »AP« v čísle 11.

Dále chci požádat své čtenáře, aby se již neobraceli s žádnými dotazy ohledně Spectra a programů na firmu MS-CID Brno. Firma se již Spectru nevěnuje (přešla na PC) !

Pan Pokorný ze Šlapanic nám sděluje, že se mu podařilo sehnat program D-TEXT od fy LOETECH.

Je to program pro tisk na BT100 (party B225-PA+PB). Tiskne sady velikosti 6, 8, 10 a 12. Na disketě je také EDITOR 12x8 pro tvorbu vlastních sad DTEXTu a 18 je již vytvořených. Cena programu je s manuálem i disketou 129,-Kčs. Program nabízí Leoš Šik, Ráj 23, 627 00 BRNO.

Pan P.Petro má výhrady k ceně časopisů, respektive k podivnému rozdílu mezi cenou AP na předplatné a ve volném prodeji. Je to skutečně neobvyklé, aby ve volném prodeji byl časopis levnější než na předplatné. Do předplatného je totiž započítána 1,-Kčs poštovné a 1,-Kčs paušál na SPD. Proto je tedy ve volném prodeji AP levnější o 2,-Kčs. Ten kdo si zakoupí AP volně, však nemá možnost získávat programy z SPD. Proč je tedy ještě nutné posílat 10,-Kčs ? Je to částka, která by měla pokrýt

poštové a částečně ještě přispět na amortizaci kopírovacího zařízení. Chyba je však v tom, že předplatitel, který SPD za celý rok ani jednou nevyužije, se vlastně podílí na krytí amortizace zařízení. Pan Petro by spíše uvítal jisté zvýhodnění předplatitelů, což je obvyklé u většiny časopisů, a raději více zaplatil za SPD vždy, když ho využije. Je to jistě správná připomínka a pro mne námět, ale obávám se, že pro letošní rok s tím asi již nic nebudu moc dělat. Prozatím tedy mohu pouze doporučit; využijte sítě SPD na plno.

Pan Dr.Holas z Ostravy má zájem o Text Machine ve verzi pro D-40. Psal proto p.Mihulovi, ale ten mu prý dosud neodpověděl. Chci tedy touto cestou ještě jednou požádat všechny zájemce o tento program, aby p.Mihulovi neposílali žádné své dotazy ! Disketová verze TextMachine již existuje a v současné době se hledá distributor. Pan Mihula nám slíbil, že jakmile to bude možné, program

nám pošle a budeme vás včas o tom informovat (já už se na něho těším!).

Pan Spáčil z Opavy nás žádá, zda bychom nemohli zveřejnit alespoň stručný popis programu MRS, neboť si neví rady se zadáváním některých instrukcí. Shodou okolností vyšel popis MRS v 11. čísle FIFA. Pokud FIFO neodebíráte, jistě vám ho někdo rád zapůjčí.

O rozdílnostech mezi počítačem Didaktik Gama a M jistě již ví více jejich uživatelé. Největší rozdíl je však v tom, že Gama má ještě i tzv. banku. Pokud nám pane Poláku z Chebu svůj program vytvořený na Gamě pošlete, povím Vám proč nechodí na Didaktiku M. Z Vašeho dopisu to však nemohu posoudit.

Petr Černý

NEBOJTE SE STROJÁKU

pokračování z č.2/92



PROGRAMOVÝ ROZSAH BASICU

Číslo řádky, totiž 10 je umístěno v prvních dvou buňkách ve tvaru:

Číslo řádky = 256 * PEEK první adresa + PEEK druhá adresa

Všimněte si, že konvence (dohoda) Z80R - násobit obsah druhé adresy 256 a připočíst k obsahu první adresy, se nepoužívá.

Konvence se však používá u dalších dvou paměťových míst, 23757 a 23758, která uchovává v paměti společně délku zbývajících řádky, která začíná v buňce 23759. V tomto případě číslo uchované v paměti je $14 + 256 \times 0 = 14$

Stejně začíná další řádka v buňce $23759 + 14 = 23773$

V buňce 23759 je uchováno v paměti číslo 234. To je znakový kód REM. Dalších 12 buněk obsahuje znakové kódy 11 písmen a prázdného znaku v titulu:

Peek program

Konečně obsahuje buňka 23772 číslo 13, které je kódem pro ENTER, ten ukazuje, že řádka je u konce. Následující tabulka shrnuje metodu pomocí které se kódují programy v programovací oblasti. V tabulce chybí popis metody, která se používá k uchování v paměti hodnot,



kteří se vyskytují v programu. Tuto metodu lze zkoumat následujícím způsobem: nahradíme v programu z minulého čísla řádku 10 takto; 10 LET a=1443

Vyobrazení ukazuje výsledek, jestliže zpracováváme program v tomto tvaru pomocí RUN.

pam.buňky obsah

1 a 2 Číslo řádky uchované v paměti v obráceném pořadí ke konvenci Z80A

3 a 4 Délka řádky bez prvních 4 buněk

5 kód instrukce

poslední Znak ENTER, číslo 13

Tvar ve kterém je uchována v paměti programové oblasti řádka

10 LET a=1443	
23755	0
23756	10
23757	14
23758	0
23759	241
23760	97
23761	61
23762	49
23763	52
23764	52
23765	51
23766	14
23767	0
23768	0
23769	163
23770	5
23771	0
23772	13

Buňky 23755 až 23758 obsahují stejné hodnoty jako dříve. Následující kódy pro LET a = a 4 číslice po sobě, které společně vytvoří číslo 1443. Dalším bodem je číslo 14 v paměťové buňce 23766. To je znakový kód, který udává, že následujících paměťových buněk obsahuje číslo 1443 v číselném tvaru. Řádka končí v buňce 23772 znakem ENTER jako předtím.

zobrazování čísel pomocí 5 bytů

Používá se 5 paměťových míst k tomu, aby se uchovávalo v paměti každé číslo, které se vyskytuje v programu Basic (kromě číslic řádků, jak jsme již viděli).

Celá čísla mezi -65535 a 65535 se uchovávají v paměti způsobem, který je souhlasný s konvencí Z80A. Pro tato čísla obsahují obě první paměťová místa a poslední buňka nulu; třetí a čtvrtá buňka obsahují číslice ve tvaru

číslo=PEEK 3. buňka + 256 x PEEK 4. buňka

Tak se např. 16553 uchovává v paměti v pěti buňkách takto:

0 0 169 64 0

protože $169 + 256 \times 64 = 16553$ Číslice, které nejsou celé číslo, se uchovávají v paměti ve tvaru s pohyblivou řádkovou čárkou a to s exponentem v první buňce a mantisou v následujících čtyřech buňkách, tzn.

číslo = mantisa x 2[↑]exponent

První buňka mantisy se také používá ke stanovení znaménka čísla. Obsahuje-li buňka číslíci mezi 0 a 127 je číslo kladné, jinak záporné.

Následující program se může používat k tomu, aby rekonstruoval čísla, které není celým číslem z hodnot svých pěti komponentů:

```
10 PRINT "udejte exponent a 4 byty man-
tisy. Vsechny udaje musi byt mezi 0 a 255
vcetne"
20 INPUT e,a,b,c,d
30 PRINT "Exponent=";e
40 PRINT "Mantisa=";a,b,c,d
50 PRINT ">>>>=";(2*(a<128-1)*2↑(e-160)
*(11(256*(a+128*(a<128))+b)*256+c)*256+d)
```

oblast proměnných

Oblast proměnných začíná u paměťové-
ho místa, které je uchováno v paměti u
systému proměnných VARS. Ten je opět
uložen v 23627. Při každém deklarování
nové proměnné, buď v programu nebo z
klávesnice, se jí dá k dispozici přiměřený
dostatek místa v této oblasti.

Názvy všech proměnných musí začínat
nějakým písmenem, přičemž se zachází s
velkými písmeny stejně jako s malými
(překódování). To umožňuje zacházet
šikovně se znaménkovým kódem
předního písmene každé proměnné; a
sice tak, že může rozeznávat 6

dovolených typů pro-
měnných tím způsobem,
že se podívá na rozsah
číslic ve kterých leží
kód. Všechny číslíkové proměnné s
jedno-písmenným názvem mají
např. kód, který leží mezi 97 a 122;
písmeno »a« je kódováno »97«,
písmeno »b« je »98« atd. Stejně
mají číslíkové Arrayové kódy mezi
129 a 153; »a« je kódováno jako
»129«, »b« jako »130« atd. Rozsah
kódů je shrnut v tabulce, která
kromě toho udává délku každého
typu proměnné.



typ prom.	rozsah zn. kódu	délka oblasti proměnných
Číslíkově ná- zev s 1 znákem	97 až 122	6
Číslíkově ná- zev s více znaky	161 až 186	5+délka názevu
Číslíkově Array (pole)	129 až 154	4+2 počet dimenzí+5 celkový po- čet prvků
Oběžná pro- měnná smyčka FOR-NEXT	225 až 250	18
Řetěz	65 až 90	3+délka ře- tězce
Znak Array	193 až 218	4+2 počet dim.+celk. poč.prvků

programování v basicu pro

kompilátor M CODER

Poprvé jsem se o použití kompilátoru Basicových programů zmínil v 11. čísle loňského roku. Až dosud jsem však hovořil pouze o jediném, který je znám pod označením »COLT«. Kládl jsem také otázku, zda kompilovat nebo ne.

K čemu vlastně může být kompilátor dobrý, když v současné době je spíše uživatel nabádán aby se učil Assembler. Ten však již vyžaduje jisté

vědomosti o struktuře počítače, je tedy nutno dobře vědět, kde jak co funguje. Basic je o mnoho snadnější, ale také mnohem pomalejší. Proto existují kompilátory Basicových programů. Neznámější z nich je právě COLT a M CODER.

O »COLTu« jsem se již zmínil a zveřejněn byl také program pro kompilaci tímto kompilátorem. Já však chci hovořit o »M CODERu«, neboť je kratší, a to je důležité, a programování pro něho se již trochu podobá strojnímu programování.

Je nutno si uvědomit, že M CODER »udělá« z našeho Basicu strojní program. Proto nemůžeme použít některé vylože-

né Basicové funkce a příkazy. Nelze vymezit prostor pro řetězcové proměnné příkazem DIM, stejně jako nelze použít příkazů pro převod řetězce do číselné hodnoty VAL a STR\$. Jiné už je to s funkcí CHR\$, která je pro M CODER přirozená. Cyklického průběhu FOR-NEXT sice použít můžeme, ale nemůže následovat STEP. Rovněž nemůžeme za INPUT použít LINE, nebo za PRINT dvojitý křížek « a ani psát do řádků AT 22 a AT 23. Skoky GOTO a GOSUB musí být přesně určeny číslem řádku, nelze tedy použít např. GOTO start a pod.



Pro porovnávání stlačené klávesy nebo jakýchkoliv jiných hodnot nemůžeme použít operátor logického součtu OR. Vůbec nejde použít osvědčeného programování tzv. »úsporným Basicem«. To znamená, že nemůžeme určit třeba LET a=NOT Pi, neboť to je výraz vhodný skutečně jen do Basicu. Strojní funkce musí být zcela jasně konkrétní. Použití nelze ani CLEAR a LIST nebo čtení z obrazovky příkazem SCREEN\$ (x,y).



Možná, že se vám teď bude zdát, že MCODER je vlastně k ničemu. Není to však pravda. Basicový program zkompilovaný MCODERem pracuje neuvěřitelně rychle. Jen je třeba si Basicový program připravit tak, aby se co nejvíce podobal strojnímu programu.

Všechny řetězcové i číselné hodnoty a proměnné je lépe ukládat přímo jako kódy do konkrétních buněk. Ušetříme tím i hodně místa, neboť nebude nutné vytvářet oblast pro ukládání proměnných. Všechna data se pak zpracovávají jako kódy bez dalšího zbytečného převádění. Stejně tak i soubor takto vytvořený se uloží na disk nebo kazetu jako CODE. Místo určování proměnné třeba LET a=128 použijeme raději POKE adr,128. Až budeme proměnnou potřebovat, vyzvedneme si ji PEEK adr. Pokud je hodnota proměnné



větší než 255 použijeme rozložení této hodnoty na dvoubajtové číslo a takto ji uložíme do dvou buněk. Návod na rozložení čísla většího než 255 na dvoubajtové číslo a jeho zpětné složení, najdete na straně 10 čísla 10 ápička z loňského roku. Budu však o tom znovu v tomto seriálu hovořit, neboť si ještě ukážeme jak číslo rozložit i na více bajtů než je dva, a vůbec si budeme muset podrobněji ukázat, jak řešit třeba sčítání větších čísel než je 30000, neboť MCODER je jinak píše jako minusová. Ale o tom až zase příště.



struktura

procesoru

Ing.
František
Sláma

řadič

Řadič (řídící jednotka) v angl. nazývaný »CONTROL UNIT«, koordinuje všechny funkce mikropočítače. Má tyto tři hlavní úkoly:

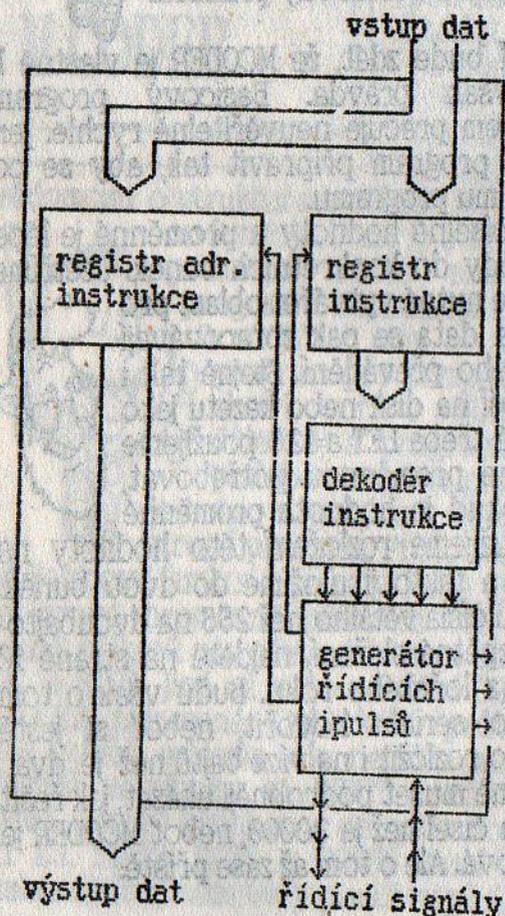
- řídí pořadí, v němž jsou prováděny instrukce programu
- dekóduje instrukce a v případě potřeby je modifikuje
- vysílá do ostatních částí počítače všechny číselné řídicí signály potřebné pro provádění instrukcí

Jakým způsobem řadič tyto tři úkoly řeší, vidíme na obrázku vedle.

Důležitou částí řadiče je REGISTR ADRESY INSTRUKCE (progr. registr), který může obsahovat čísla od nuly až do hodnoty adresy paměťové buňky. Tato hodnota určuje maximální kapacitu paměti, kterou lze k danému procesoru připojit. Po zapnutí počítače se registr adresy instrukce nastaví na

určitou výchozí hodnotu, což obvykle bývá nula. Výstupy tohoto registru jsou spojeny s adresovými vodiči, které vedou k paměťovým

obvodům mikropočítače. Program, který se má provádět, musí být v operační paměti uložen od adresy



číslo nula. Registr adresy instrukce je často umístěn také v aritmetické jednotce. Má tam stejnou funkci a je řízen prostřednictvím spojovacích vodičů z generátoru řídicích impulsů umístěného v řadiči.

Po zapnutí počítače nebo po stisknutí nulovacího tlačítka se registr adresy instrukce vynuluje. Tímto nulovým obsahem se adresuje paměť počítače a na jejím výstupu se objeví číslo, které odpovídá kódu (operačnímu znaku) první instrukce. Tento kód se prostřednictvím datových vodičů převede do REGISTRU INSTRUKCE v řadiči. Tam se instrukce nejprve zapíše, aby připojený DEKODÉR INSTRUKCE měl dostatek času tento kód instrukce dekodovat. Z dekodéru je pak do GENERÁTORU ŘÍDICÍCH IMPULSŮ předána informace, která instrukce se má provést.

Generátor řídicích impulsů vysílá v určité časové posloupnosti impulsy do všech ostatních jednotek mikropočítače, především do aritmetické jednotky. Tím umožňuje provádění instrukce. Je-li instrukce provedena nebo jsou-li k jejímu provedení potřebná další data z operační paměti, vydá generátor řídicích impulsů povel registru adresy instrukce, čímž se jeho obsah zvětší (nejčastěji o jedničku).

Na adresových vstupech paměti se tak objeví adresa následující instrukce nebo následujícího údaje. Informace, která je na této adrese zapsána, je vyslána na datové vodiče a celý cyklus začíná znovu od začátku.

Na obrázku je nápadné, že datové spoje nevedou jenom k registru instrukce, ale také k registru adresy instrukce. Tím je umožněno, aby se speciálními instrukcemi měnil obsah tohoto registru. To je nutné např. tehdy, chceme-li v programu pokračovat v jiném úseku paměti. Adresa nového úseku paměti se potom vloží jako nový obsah do registru adresy instrukce, který tímto obsahem automaticky adresuje následující část programu a z ní čte příští instrukci.

□ □ □ □ □ □ □ □ □ □

□

□

□

□

□

□

□

□

□

□

□

□

□

□

□

□

PRO VOLNOU

CHVILKU

5 FOR a=0 TO 155

10 PLOT a,a: DRAW 0,175-(2*a)

20 PLOT 255-a,a: DRAW 0,175-(2*a)

30 NEXT a

□ □ □ □ □ □ □ □ □ □

□

□

□

□

FUNKCE RND

Pro některé aplikace, např. pro počítačové hry, jsou zapotřebí náhodná čísla. Vytváření náhodných čísel v Basicu zajišťuje funkce RND.

Výsledkem volání funkce RND je náhodné číslo z intervalu $\langle 0, 1 \rangle$, parametr funkce většinou nemá význam, v některých implementacích možno použít ji bez parametrů.

Při každém vyvolání funkce RND se generuje podle jistého algoritmu další náhodné číslo, které ovšem není čistě náhodné,

ale tzv. pseudonáhodné, tzn. že pro většinu praktických aplikací má vlastnosti náhodného čísla, ale při opětovném spuštění programu se generuje stejná posloupnost pseudonáhodných čísel. V aplikacích, kde by to vadilo, lze v některých verzích Basicu použít příkaz RANDOMIZE, který způsobí, že se pokaždé použije jiná inicializace pro generování posloupnosti pseudonáhodných čísel, např. reálný čas a pod., čímž se dosáhne pokaždé jiné posloupnosti generovaných pseudonáhodných čísel.

Zpravidla potřebujeme náhodná čísla z jiného rozsahu než $\langle 0, 1 \rangle$. Musíme tedy číslo generované funkcí RND dále upravovat. Jak se to dělá, si ukážeme na několika příkladech.

příklad

X	$2 * \text{RND}(X)$	$\text{INT}(Y)$
0	0.0	0
0.1	0.2	
0.2	0.4	
0.3	0.6	
0.4	0.8	
0.5	1.0	1
0.6	1.2	
0.7	1.4	
0.7	1.4	
0.7	1.4	
0.7	1.4	1
0.8	1.6	
0.9	1.8	

Generování náhodné posloupnosti nul a jedniček, tj. simulace házení mincí (1, 0, rub).

Převod intervalu $\langle 0, 1 \rangle$ na dvouprvkovou množinu můžeme provést tak, že z intervalu $\langle 0, 0.5 \rangle$ budeme generovat nulu a z intervalu $\langle 0.5, 1 \rangle$ generovat jedničku. Např. pomocí funkce $\text{INT}(2 * \text{RND}(X))$ dostaneme to, co vidíte na obrázku vlevo. Chceme-li generovat náhodná čísla nap. z intervalu $\langle 1, 6 \rangle$, tj. házení kostkou, můžeme použít vztahu

$$\text{INT}(6 * \text{RND}(X)) + 1$$

Obecně pro interval celých čísel $\langle N_{\text{MIN}}, N_{\text{MAX}} \rangle$ lze použít $\text{INT}(A * \text{RND}(X)) + B$,

kde $A = (N_{\text{MAX}} - N_{\text{MIN}}) + 1$

$B = N_{\text{MIN}}$

Je-li pro účely cíle na obrazovce při počítačové hře třeba generovat náhodně čísla rozlišená na několik desetinných míst, je transformační vztah složitější;

$\text{INT}((A * \text{RND}(X)) * 10^D) / 10^D + B$ (*)

kde $A = N_{\text{MAX}} - N_{\text{MIN}} + 10^D$,

$B = N_{\text{MIN}}$, tj. dolní hranice generovaného rozsahu a

D je žádaný počet desetinných nebo celých míst.

Výraz 10^D udává rozlišovací schopnost, tzn. minimální možný rozdíl dvou hodnot ve vytvořené posloupnosti.

Příklad: chceme generovat náhodná čísla, která leží v intervalu $\langle 9,2, 19,3 \rangle$ s rozlišovací schopností 0,01. Použijeme uvedený výraz (*) ve tvaru

$\text{INT}(10,11 * \text{RND}(X)) * 10^2 / 10^2 + 9,2 =$

$\text{INT}(1011 * \text{RND}(X)) / 100 + 9,2$

neboť $D=2$, $B=9,2$ a $A=19,3 - 9,2 + 10 - D = 10,1 + 0,01 = 10,11$.

Rozlišovací schopnost může být celé, jak uvidíme na následujících příkladech.

Příklad: chceme generovat náhodná čísla, která leží v intervalu $\langle -100, 500 \rangle$ s rozlišovací schopností 10. Tentokrát získáme po dosažení tvar:

$\text{INT}(610 * \text{RND}(X)) * 0,1 / 0,1 - 100 =$

$\text{INT}(61 * \text{RND}(X)) * 10 - 100,$

neboť $D=-1$, $B=-100$ a $A=500 - (-100) + 10 - (-1) = 610$.

Příklad: chceme generovat náhodná sudá čísla z intervalu $\langle 0, 100 \rangle$. Nejprve vytvoříme výrazem $\text{INT}(51 * \text{RND}(X))$ náhodná čísla z intervalu $\langle 0, 50 \rangle$ a vynásobením 2 dostaneme sudé čísla z požadovaného intervalu.

Příště si již budeme povídat o základním použití jazyka Basic. budeme se postupně věnovat jednotlivým příkazům, které jsou na klávesnici. Proto si nezapomeňte zajistit odběr »AP« až do konce roku 1992.

PRODÁM logickou hru vytvořenou podle japonského hlavolamu SOKO (25,- Kčs)

Jde o přesouvání balíků v labyrintě na určené místo. Obsahuje 50 samostatných úloh. → Pouze na kazety ←

ADRESA: Mřenica František

□ □ □ □ □ PRAVENEČ 199 □ □

□ □ □ □ □ 972 16 □ □



JEDNODUCHÉ ÚČETNICTVÍ NA DIDAKTIKU M na úplně nové verzi programu »deník«



kteřá Vám již vypočítá »volnou úvahu« pro přiznání daně

Program »Deník« pro vedení jednoduchého účetnictví prodělal od svého vzniku již mnoho změn. Především díky jeho prvním uživatelům, kteří nám poskytli své poznatky a náměty na vylepšení a doplnění tohoto užitečného programu.

Současná verze již má odstraněny všechny závažné chyby, které se v programu vyskytly.

Protože programy jsou prodávány se smlouvou, které pokud ji uživatel podepíše a odešle zpět, zaručuje bezplatný servis na

chod programů, nepřišel nikdo zkrátka. Všem evidovaným uživatelům byla zaslána výzva k výměně programu.

»DENÍK« je rozšířen o evidenci vydání a příjmů, které neovlivňují daňový základ. Novou funkcí zde má klávesa EDIT která umožní "dostat se" do již vyplněné obrazovky a přidán byl mód EXTEND, kterým lze kdykoliv během psaní vyvolat tabulku s přehledem. Dále je program již vybaven pro důležité momenty chybovým hlášením, které uživatele upozorní na způsobenou chybu. Program si již také porovnává uvedené znaky a účtované částky. Pokud částka není v souladu se znakem, vydá chybové hlášení a umožní uživateli opravu. Po vyplnění celé obrazovky dá ještě uživateli šanci na opravu, dotazem zda je vše O. K.

»VOLNÁ ÚVAHA: VÝPOČET DANĚ

10.10. MĚSÍČ:	008400.00
10.10. POČET DĚTÍ:	005000.00
10.11. MANŽELST:	A 005000.00
10.1E. UŽIVNĚ:	005000.00
10.1G. INVALIDITA:	000000.00
10.1H. ČÁST. INVALID.:	003000.00
28.5. ZTRÁTA (HČS):	0000

»VOLNÁ ÚVAHA« je program umožňující rozřídění účtovných položek podle znaků, čímž uživatel získá informace o svém celoročním hospodaření.

Nejdůležitější částí programu je však bezesporu výpočet tzv. volné úvahy o tom, kolik bude činit jeho daňové příznání. Tento přehled si může vytvořit kdykoliv během roku a včas tím ovlivňovat své hospodaření.

Pro výpočet volné úvahy bylo použito pravděpodobně nejvíce využívaných paragrafů. Proto je program skutečně pouze »volná úvaha« a nikoliv »rozvaha« jak bylo původně plánováno. Pokud tedy podnikatel využívá ještě nějaké méně časté paragrafy, musí si je již doplnit sám. Výpočet z tohoto programu tedy slouží jako podklad pro vyplnění tiskopisu k daňovému příznání.

Samozřejmostí je, že provedený výpočet či třídění lze uložit na disk stejně jako i vytisknout na tiskárně. Pro tuto potřebu je součástí

PŘÍJMY:	047000,00
UVYDÁNÍ:	001450,60
01-ZÁHLAD DANĚ:	045549,40
02-ZAOKROULENĚ:	045500,00
03-510/1A (HL. ZAM.):	000400,00
06-510/1D (DĚTI):	005000,00
08-510/1F (MANŽEL -KA):	000000,00
07-510,1E (PL. VÝŽIV.):	000000,00
09-510,1G (INVALIDITA):	000000,00
10-510,1H (ČAS. INVALID):	003000,00
12-520,5 (ZTRÁTA PODM):	000000,00
14-SNÍŽ.O ODC.POL.:	014100,00
15-DANĚ (15%):	002115,00

>1<	ZÁHLADNÍ PROSTĚ.	0000000000
>2<	NÁKUP MATERIÁL	0000000000
>3<	NÁKUP DPH	0000000000
>4<	NÁKUP ZBOŽÍ	001230,00
>5<	MZDV/DANĚ ZE MZDV	0000000000
>6<	NEH./OÚCH.POJ.	0000000000
>7<	PROVOZNÍ REŽIE	000220,60
>8<	ODBYTOVÁ REŽIE	0000000000
>9<	PŘÍJEM/TRŽBY	047000,00
>U<	OSTATNÍ UYDÁNÍ	001000,00
>P<	OSTATNÍ PŘÍJEM	015000,00

diskety program

»TISKENZ« do kterého lze nahrát soubor z Deníku, z úvahy i výpočet příznání daně.

Tisknout je možné jednak různé samostatné části - např. pouze Přehled, Výpočet, Třídění a nebo celý Deník.

Před tiskem Deníku uvede uživatel nejprve datum ke kterému se tisk provede. Stránky jsou nastaveny na velikost papíru A4 a program si je postupně čísluje.

→ dokončení naleznete na staně NN ←



ČTENÁŘI
PRO
APÍČKO

FORMAT NA 433152

celkem jednoduše a snadno

Ve dvanáctém čísle loňského roku jsem otiskl informaci o programu »KOMANDER 1« pro disketovou jednotku D-40, kterou údajně nabízí a. s. Didactic ve Skalici. Několik čtenářů si program skutečně objednalo, ale bez úspěchu. Žádný takový program prý Skalica nemá. To zřejmě donutilo některé z vás znovu prostudovat manuál k disketové jednotce a výsledek se skutečně dostavil!

```
[46] 10 INPUT "Jmeno diskety " ;a$
[58] 20 POKE #6,43
[30] 30 POKE #7,10
[78] 40 LET a$="a:"+a$
[8F] 50 FORMAT a$
```

nelze použít na kdekou disketu. Tak například při formátování floppy disku typu »Double-Sided Double Density« ohlásí po skončení 2 až 5 bloků vadných. Po několikaerém zopakování formátu na 433 se vadné bloky postupně snižují. Naopak u floppy disku typu »Double-sided/Quad Density« se format povedl hned na poprvé. To podotýkám proto, aby jste nám zbytečně nepsali, jak špatné diskety se v poslední době prodávají.

```
Format complete.
Formatted 845 good blocks
and 0 bad blocks.
Total capacity is 433152 Bytes.
```

To je tedy jedna z funkcí, kterou nabízel zmíněný program KOMANDER 1. Jeho další funkcí je přejmenování názvu floppy disku. I to se dá provést bez speciálního programu (viz. další strana AP)

Na důkaz toho, že to skutečně funguje, otiskujeme kopii z obrazovky výpisu po skončení formátování.

```

[36] 10 INK 7: PAPER 0: BORDER
0: CLS
[95] 20 PRINT AT 10,0; FLASH 1
;"Stlač Libovolnou klavesu": PAU
SE 0
[19] 30 LET t=5: LET adresa=50
000
[F8] 40 PRINT INVERSE 1;AT 10,
0;"Vlož disketu do D-40, po stis
ku klavesy ji začnu přejmenováva
t!"
[3E] 50 READ z"a:",0,adresa
[BA] 60 PRINT AT 1,0; INVERSE
1;"Stare jmeno: "; INVERSE 0;
[8A] 70 FOR a=192 TO 201
[47] 72 IF PEEK (50000+a)<32 T
HEN PRINT " ";: GO TO 90
[0B] 80 PRINT CHR$ PEEK (50000
+a);
[85] 90 NEXT a
[04] 100 INPUT AT 21,0;AT 2,0;
INVERSE 1;" Nove jmeno: "; INVER
SE 0; LINE d$
[79] 105 IF LEN d$>10 THEN BEEP
1,-27: GO TO 100
[86] 108 PRINT AT 4,0; INVERSE
1;"Přejmenuji na: ";d$; INVERSE
0;
[E7] 110 PRINT @0;AT -1,4;"JE T
O O.K. ? a / n"
[00] 115 PAUSE 0
[F7] 120 IF CODE INKEY$=65 OR C
ODE INKEY$=97 THEN GO TO 130
[27] 125 GO TO 60
[06] 130 FOR a=192 TO 201
[F7] 135 POKE adresa+a,0
[E7] 140 NEXT a
[9B] 145 LET a=192
[7C] 150 FOR f=1 TO LEN d$
[BA] 155 POKE adresa+a,CODE d$(
f)
[0C] 160 LET a=a+1
[05] 165 NEXT f
[63] 170 RESTORE z"a:",0,50000
[80] 175 CLS
[39] 180 CAT

```

RENAME DISK

Pavel Jisl z Černošic, žák 7. třídy, se zamýšlel nad tím, jak přejmenovat disketu pomocí Basicového programu.

A protože dobře četl manuál k jednotce, přišel na to, jak to provést.

Program si nejprve načte sektor 0 (BOOT SEKTOR) od adresy 50000 a pak již příkazy PEEK a POKE přejmenuje disketu novým jménem.

Snad se na mě Pavle nebudeš zlobit, že jsem ti program trochu upravil. Zadávání nového jména po jednom znaku není vhodné. Chybně napsané jméno by se nedalo opravit. Méně zkušený uživatel by byl možná i trochu zmaten z otazníků za názvem, který je kratší než 10 znaků. Nyní input přijme celé jméno, zkontroluje jeho délku a zeptá se zda je to O.K., když ne, může se jméno napsat znova. Protože cyklus

FOR vypisuje celou délku pro název, tedy 10 znaků, je doplněn podmínkou, že pokud je znak menší než 32 (mezera), aby vypisoval mezeru (původní CHR\$ PEEK reagoval vypisováním otazníků). Po odsouhlasení

nového jména diskety je nejprve adresa pro uložení jména nahrazena nulami v délce deseti znaků. Teprve potom je na tuto adresu uloženo nové jméno. Program se takto zkrátil a je bezpečnější. K ostatním programům, které si poslal, se vrátím zase někdy příště.



RUN FOR D 40 verze 2

Miroslav Mošať z Hrušovan reagoval na zveřejnění popisu svého programu RUN FOR D40 tím, že nám pohotově zaslal druhou verzi tohoto velice užitečného programu pro všechny uživatele disketové jednotky D-40. Verze je rozšířena o funkci, která umožňuje výběr mechaniky A nebo B. Má pravdu v tom, že RUN FOR D40 nevyniká pompézní grafickou úpravou, jako některé od profesionálních firem, je však univerzální a dá se tedy použít pro zavádění libovolných programů. Jeho zpuštění je prakticky okamžité, což je zřejmě také proto, že údaje čte přímo z obrazovky. Odpadá zde to dlouhé čekání, než jsou názvy souborů uloženy na nějakou adresu a pak z ní zase zpětně vypisovány na obrazovku. Již dávno tvrdím, že v jednoduchosti je síla.

Program se ovládá klávesami 5,6,7 a 8 nebo Q,A,O a P, ale i šipkami. Výběr mechaniky A nebo B se provádí klávesou »M«, což je signalizované v pravém dolním rohu obrazovky. Klávesa »T« zapíná a vypíná test mezer (SPACE) a uživatel je o tom informován v pravém spodním rohu obrazovku pod signalizací mechaniky. U verze 2 se BORDER a CAT spouští jen jednou klávesou, tedy B nebo C. Ostatní funkce i činnost programu zůstává stejná jako u verze 1.

Pokud nám to pan Mošať dovolí, budeme RUN FOR D40 umísťovat na všechny diskety zasláné do SPD.

PROGRAM POKE poprvé v úpíčku

```

[9F] 10 LET a=1: LET b=40000
[78] 12 INPUT "Pocet vkladanych
h hodnot ? ";p
[38] 13 IF p<>INT p OR p<=0 OR
p>30 THEN BEEP .5,10: GO TO 12
[86] 15 DIM M(p,2)
[87] 20 INPUT "Zadaj meno prog
ramu ? "; LINE a$
[88] 30 INPUT "adresa ? ";c
[89] 35 IF c<>INT c OR c<16384
OR c>65535 THEN BEEP .5,1: GO T
O 30
[35] 37 LET M(a,1)=c
[79] 40 INPUT "hodnota ? ";d
[89] 42 IF d<>INT d OR d>255 O
R d<0 THEN BEEP .5,20: GO TO 40
[3E] 43 LET M(a,2)=d
[38] 50 PRINT a$;" ";c;" ";d
[61] 51 LET a=a+1: LET p=p-1
[84] 52 IF p<>0 THEN GO TO 30
[66] 53 LET p=a-1: LET a=1
[65] 54 GO SUB 150: GO SUB 55:
GO SUB 70: GO SUB 105: GO SUB 1
60: GO SUB 120: GO SUB 220: GO T
O 54
[3E] 55 LET c=c-16256
[9A] 60 LET c1=INT (c/512): LE
T c2=c-c1*512: RETURN
[18] 70 PRINT "sektor ";c1
[DD] 80 PRINT "adresa ";(c2+b)
[8C] 100 READ *a$,c1,b: RETURN
[52] 105 PRINT "povodna hodnota
";PEEK (c2+b)
[AD] 110 POKE (c2+b),d: RETURN
[82] 120 RESTORE *a$,c1,b: RETU
RN
[6A] 130 STOP
[61] 150 LET c=M(a,1): LET d=M(
a,2): LET M(a,1)=0: LET M(a,2)=0
: LET a=a+1
[78] 152 IF c=0 THEN GO TO 150
[9F] 155 RETURN
[CD] 160 FOR x=1 TO p
[5C] 170 LET q=M(x,1): LET c3=c
1*512+16256
[83] 180 IF q=0 THEN GO TO 200
[AF] 190 IF q=c3 AND q<=(c3+51
2) THEN GO SUB 210: GO SUB 55: G
O SUB 105
[3A] 200 NEXT x

```

Protože »AP« se již věnuje pouze oblasti programování, nebyla dosud zveřejněna ani jedna rada jak poukat hry.

Přesto si jistě občas sem tam nějakou hru zahrájete. V tom případě se vám možná bude docela dobře hodit program, který nám poslal také Miroslav Mošal z Hrušovan.

Program umožňuje zadat do libovolné hry nebo programu jakoukoliv hodnotu (POKE), ale musí být splněna jedna podmínka: »hra nebo program musí být uložen na disk tlačítkem SNAP.

Je to dobré hlavně pro začátečníky v programování, neboť program jim umožní vložit jakýkoliv POKE do libovolné hry.

Princip činnosti programu je velmi jednoduchý. Nejprve si vyžádá jméno programu a počet vkládaných hodnot POKE. Potom si vyžádá hodnoty, které přepočítá na sektory,

```

[101] 205 RETURN
[111] 210 LET C=M(x,1): LET d=M(
x,2): LET M(x,1)=0: LET M(x,2)=0
: RETURN
[109] 220 FOR x=1 TO P
[118] 230 LET q=M(x,1)
[130] 240 IF q(<)0 THEN RETURN
[160] 250 NEXT x
[191] 260 PRINT "Chcete nahrat pr
ogram a/n ? "
[130] 270 IF INKEY$="a" OR INKEY
$="A" THEN LOAD #a$
[121] 280 IF INKEY$="n" OR INKEY
$="N" THEN STOP
[141] 290 GO TO 270

```

někaké hodnoty, nelze je již vrátit na původní stav. Proto je lépe si zhotovit »záložník« SNAP programu, ve kterém chceme měnit hodnoty. Věřím však tomu, že další hloubavý čtenář program POKE o tuto funkci jistě doplní. Pokud na to přijde, ať neváhá a pošle to také do »ápíčka« pro zveřejnění v rubrice »ČTENÁŘI PRO ÁPÍČKO«.

a požadovaný sektor načte. Pak přepíše hodnotu a sektor nahraje zpět už z upravenými hodnotami.

Ani tento program není nijak graficky upravený, ale jde tu hlavně o nápad, jak si jednoduše poradit se zadáváním POKE.

Program nemá žádnou funkci typu UNDO, resp. ZPĚT, takže když změním v programu

O REM zase jinak

```

[170] 10 REM          0 REM
[171] 20 LET s=0: LET adr=23300
[187] 30 FOR i=0 TO 63
[193] 40 READ a: LET s=s+a: POK
E adr+i,a
[181] 50 NEXT i: IF s(<)5495 THE
N PRINT "CHYBA!": STOP
[191] 60 SAVE # "0REM" CODE adr,6
4
[131] 1000 DATA 231,205,130,28,20
5,153,30,96,185,229
[161] 1010 DATA 1,6,0,9,66,77,42,
83,92,84
[141] 1020 DATA 93,43,213,205,85,
22,225,34,83,92
[171] 1030 DATA 193,54,0,35,54,0,
35,3,3,113
[191] 1040 DATA 35,112,35,54,234,
35,11,11,84,93
[151] 1050 DATA 19,54,48,237,176,
54,13,42,61,92
[101] 1060 DATA 43,43,249,201

```

Nejvíce musím pochválit čtenáře Vladimíra Vojtu z Aše. Ten totiž dokázal během dvou týdnů zaslat do redakce celkem třikrát vlastní verzi programu pro umístění strojních rutin do řádku 0 REM.

Původně ho k tomu vyprovokoval chybně opsaný program 0 REM z 11. čísla AP (bez verifikátoru). Na chybu přišel až později, ale to už měl hotový

→ SINCLAIR CLUB ←

P.O. BOX 132 - OSTRAVY 363 01

nabídka pro uživatele

DIDAKTIK 40 - Máte disketovou jednotku D-40?
Nevíte jak ji nejlépe využít?

a K 6304 Máte problémy se sháněním software?

Pokud jsou vaše odpovědi: ANO, NE, ANO, pak vám mohu doporučit nabídku, která nám přišla ze Sinclair Clubu v Ostrově.

Členem klubu uživatelů D-40 se může stát každý uživatel této jednotky, který řádně vyplní přihlášku a zašle roční klubový příspěvek na adresu Klubu D-40. Ten činí 50,-Kčs + 5 korunových známek.



co pro vás klub D 40 nabízí

- jednou ročně obdržíte poštovní disketu v ceně 20,-Kčs se seznamem členů Klubu D-40 a Sinclair Clubu Ostrav
 - 5x ročně písemnou informaci o činnosti a aktivitě klubu a nejen o ní
 - informace o nových programech pro D-40
 - programy public domain vytvářené pro klub
 - poradenskou službu k hardware a software řadiče D-40
 - zvýhodněný prodej některých zajímavých původních programů pro ZX5 s licencí pro členy klubu
 - možnost zúčastňovat se příležitostných setkání Klubu D-40 (organizace Sinclair Club Ostrav)
- ☐ Přihlášku a další informace získáte

proti známce na této adrese:

→ / Jaroslav Kařínek ←
 KLUB UŽIVATELŮ D-40
 M.Gorkého 3
 431 91 VEJPRTY

telefon: (0398) 95 62 26

P.S.: Tyto informace byly původně určeny jen pro »mou« lepší orientaci. Snad se tedy nebude na mne J.Kařínek zlobit, že jsem je zveřejnil. Činnost tohoto klubu je totiž zajímavá, že by asi bylo sobecké si tyto informace ponechat jen pro sebe.

sinclair club ostrov

založila v červnu minulého roku programátorská skupina »Studio 22«. Za duchovního otce klubu je považován J.Krejčí, který tohle všechno vymyslel a taky zatím hlavně organizuje.

Za krátkou chvíli existence má klub přes 50 členů a několik tzv. poštovních. Stáli také u zrodu Klubu D-40 ve Vejprtech. Od začátku udržují dobré styky s časopisem »ZX magazin« a několika předními čs. programátory.

Společně s Post Spectrum Clubem z Bánské Bystrice zakládají společnou databázi uživatelů ZX Spectrum a jeho pokračovatelů. S tímto klubem též připravují vydání několika zajímavých příruček pro ZXs a o nich pochopitelně budeme čtenáře AP také informovat.

a ještě něco pro robotrona

Klub má k dispozici velké množství informací a firemních materiálů k této tiskárně a také zkušenosti s připojováním této tiskárny přes kdejaký interface. Mimochodem, standartní zapojení po sériové lince pochází z jejich dílen.

V klubu také vznikla nová verze (kompatibilní s předcházími) zapojení desky »cartridge« typu Centronics pro tuhle tiskárnu. Protože žádosti o pomoc a radu s připojováním tiskárny přibývá, rozhodli se vydat »PŘÍRUČKU ČESKÉHO UŽIVATELE K5384«, ve které chtějí popsat své zkušenosti s touto tiskárnou.

Podobně se také věnují plotteru XY41NN, pro který chystají něco podobného. Již nyní však mohou poposkytnout kontakt na zajímavé programy pro tuto grafickou jednotku.

chcete

rozšířit možnosti svého počítače ?

SH2

J. BROŽÍK - Lužická 296/111
332 01 - ROKYCANY

VÁM NABÍZÍ
→ PRO POČÍTAČE... ↓↓

SPECTRUM - DIDAKTIK - DELTA - SAM COUPE

INTERFACE 8255

pro připojení tiskáren, plotterů a jiných periférií → NOVINKA+
IF je průchozí, možnost připojení dalších periférií za 1F □ 365,-Kčs
s KEMPSTON s autofire □ 445,-Kčs

INTERFACE II

80 KB

připojení dvou joysticků □ 295,-Kčs
vylepšené zapojení z ST11/87 + kazeta se software
pro ZX SPECTRUM □ 695,-Kčs
pro DIDAKTIK GAMA □ 395,-Kčs

DIDI SUPER

SNAP, opětavné

pouze pro Didaktik Gama
možnost zastavení libovolného programu POKE,
spuštění, využití libovolné ROM + software □ 245,-Kčs

ZPOMALOVAČ HER

0-100%,
zabudování
do libovolného
interface nebo
počítače lze i do
SAM COUPE

umožňuje plynulé zpomalení jakéhokoliv programu

□ 145,-Kčs

PŘÍDAVNÁ ROM

16 kB ROM del výběru; TURBO, ISO, BETA a další □ 295,-Kčs

SOUNDFACE S AY3810

3 stereo zvukové kanály, zvukové možnosti Spectra 128,
2 I/O porty □ 195,-Kčs

KZD 1

pouze do vyčerpání zásob !
8x rychlejší než standardní mgf, plně ovládané
počítačem

□ 595,-Kčs

dále nabízíme

THERMO TISKÁRNU ROBOTRON K 6309

s připojením k Vašemu paralelnímu interface (příp. k našemu 1F8255)

+ software

→ 1990.-KČS ←

+ 3 role thermopapíru

→ 1 ROK ZÁRUKA! ←



připravujeme

- ❑ RS 232
- ❑ 80kB dle ST11/87 pro 128 Spectrum
- ❑ světelné pero
- ❑ 512 kB pro SAM COUPE
- ❑ rozdvajka pro JOY pro SAM COUPE
- ❑ interface 8255 pro SAM COUPE

VÝKUP A
DISTRIBUCE
SOFTWARE
PRO SPECTRUM
A SAM COUPE

SH2

Pro zájemce, kteří si již zakoupili kazetopáskovou jednotku KZD-1 upozorňujeme, že máme provedeno připojení ke Spectru! Můžete se na nás kdykoliv obrátit!

Lužická 296
ROKYCANY
337 01

píše na naši adresu

POZOR NA PŘEDPLATNÉ!

Toto číslo je posledním číslem 1.čtvrtletí a proto si nezapomeňte zajistit odběr »AP« na další období! Předplatit si můžete částkou 30,-Kčs na 2.čtvrtletí ❑ 60,-Kčs do září 1992 ❑ 90,-Kčs do konce roku V každém případě však musí být uhrazeno nejpozději do 10.března!



FLOPPY DISK

MAGNETOFON

SECOM

PUBLIC

DOMAIN

PROGRAM MAKER

Po úspěšném kompletu programů pro tvorbu a zpracování grafiky a textu »GRAPHICS« je pro vás připraven další komplet, tentokrát převážně strojních rutin. Mají usnadnit programování v Basicu.

Je to vlastně výběr strojních rutin, při psaní a odladování programu mohou v mnohém pomoci.

Některé již budete asi znát ze stránek »AP« a o některých možná i z jiných časopisů.

Protože právě probíhá seriál o kompilátorech Basicových programů, je na disketě zařazen kompilátor »COLT« a »MC2«, který je částečně upravený MCODER.

Pro bezpečnější psaní programu je určena rutina »AUTOLINE«, která automaticky čísluje příkazové řádky. Programátor si tedy může předem určit krokování a dál už jen psát a psát. . . . Pokud se v jeho programu často vyskytují stejné kombinace příkazů, jako například;

IF CODE INKEY#88 THEN GO TO a pod,
pak s radostí využije rutiny, která mu umožní předem si nadefinovat takovou kombinaci pod nějakou klávesu. Jejím stiskem se mu pak automaticky celá tato kombinace do programu sama umístí. Až bude svůj program tzv. doladovat, jistě ocení rutinu pro přečíslování celého programu včetně všech skoků GOTO a GOSUB, stejně jako rutinu »TRACE«, která mu při běhu programu bude neustále vypisovat, na kterém řádku se právě program vykonává. Takových »lahůdek« je tam ale více. Aby to majitelům magnetofonů nebyla líto, ta sděluji, že si mohou poslat kazetu.

